



The Weblate Manual

Реліз 4.16.4

Michal Čihař

бер. 16, 2023

1	Документація для користувачів	1
1.1	Основи Weblate	1
1.2	Реєстрація і профіль користувача	1
1.3	Переклад за допомогою Weblate	10
1.4	Отримання і вивантаження перекладів	20
1.5	Глосарій	23
1.6	Перевірки і виправлення	26
1.7	Пошук	59
1.8	Процеси перекладу	64
1.9	Поширені питання та відповіді на них	68
1.10	Підтримувані формати файлів	77
1.11	Налаштування інтеграції із керуванням версіями	100
1.12	Програмний інтерфейс REST Weblate	109
1.13	Клієнт Weblate	157
1.14	Програмний інтерфейс Python Weblate	163
2	Документація для адміністраторів	165
2.1	Настанови з налаштування	165
2.2	Розгортання Weblate	234
2.3	Оновлення Weblate	235
2.4	Резервне копіювання і пересування Weblate	245
2.5	Розпізнавання	252
2.6	Керування доступом	262
2.7	Проекти перекладу	272
2.8	Визначення мов	291
2.9	Безперервна локалізація	294
2.10	Ліцензування перекладів	303
2.11	Процес перекладу	305
2.12	Перевірки і виправлення	311
2.13	Налаштування автоматичних пропозицій	322
2.14	Додатки	333
2.15	Пам'ять перекладів	351
2.16	Налаштування	353
2.17	Зразок налаштувань	381
2.18	Команди керування	396
2.19	Оголошення	408
2.20	Списки складників	411
2.21	Додаткові модулі Weblate	412
2.22	Налаштування Weblate	417
2.23	Інтерфейс керування	420
2.24	Отримання підтримки щодо Weblate	428

2.25	Правові документи	431
3	Документація для учасників	433
3.1	Участь у розробці Weblate	433
3.2	Перші кроки у програмуванні коду Weblate	435
3.3	Початковий код Weblate	440
3.4	Діагностика у Weblate	441
3.5	Нутроші Weblate	443
3.6	Розробка додатків	444
3.7	Оболонка Weblate	446
3.8	Надсилання звітів щодо вад у Weblate	447
3.9	Комплекс тестування Weblate і неперервна інтеграція	448
3.10	Схеми даних	450
3.11	Створення випусків Weblate	453
3.12	Безпека і конфіденційність	454
3.13	Участь у розробці модулів Weblate	455
3.14	Про Weblate	456
3.15	Ліцензія	457
4	Журнал змін	458
4.1	Weblate 4.16.4	458
4.2	Weblate 4.16.3	458
4.3	Weblate 4.16.2	458
4.4	Weblate 4.16.1	459
4.5	Weblate 4.16	459
4.6	Weblate 4.15.2	459
4.7	Weblate 4.15.1	459
4.8	Weblate 4.15	460
4.9	Weblate 4.14.2	460
4.10	Weblate 4.14.1	461
4.11	Weblate 4.14	461
4.12	Weblate 4.13.1	462
4.13	Weblate 4.13	462
4.14	Weblate 4.12.2	463
4.15	Weblate 4.12.1	463
4.16	Weblate 4.12	463
4.17	Weblate 4.11.2	464
4.18	Weblate 4.11.1	464
4.19	Weblate 4.11	464
4.20	Weblate 4.10.1	465
4.21	Weblate 4.10	465
4.22	Weblate 4.9.1	466
4.23	Weblate 4.9	466
4.24	Weblate 4.8.1	467
4.25	Weblate 4.8	467
4.26	Weblate 4.7.2	468
4.27	Weblate 4.7.1	468
4.28	Weblate 4.7	468
4.29	Weblate 4.6.2	469
4.30	Weblate 4.6.1	469
4.31	Weblate 4.6	469
4.32	Weblate 4.5.3	470
4.33	Weblate 4.5.2	470
4.34	Weblate 4.5.1	471
4.35	Weblate 4.5	471
4.36	Weblate 4.4.2	472
4.37	Weblate 4.4.1	472
4.38	Weblate 4.4	473

4.39	Weblate 4.3.2	473
4.40	Weblate 4.3.1	474
4.41	Weblate 4.3	474
4.42	Weblate 4.2.2	475
4.43	Weblate 4.2.1	475
4.44	Weblate 4.2	475
4.45	Weblate 4.1.1	476
4.46	Weblate 4.1	477
4.47	Weblate 4.0.4	478
4.48	Weblate 4.0.3	478
4.49	Weblate 4.0.2	478
4.50	Weblate 4.0.1	479
4.51	Weblate 4.0	479
4.52	Випуски Weblate 3.x	480
4.53	Випуски Weblate 2.x	492
4.54	Випуски Weblate 1.x	504
4.55	Випуски Weblate 0.x	508
Python Module Index		512
HTTP Routing Table		513
Индекс		516

1.1 Основи Weblate

1.1.1 Структура проєктів і складників

У Weblate переклади упорядковано за проєктами та складниками. Кожен проєкт може містити декілька складників, які містять переклади окремими мовами. Складник відповідає одному файлу перекладу (наприклад, *GNU Gettext* або *Рядкові ресурси Android*). Проєкти призначено для полегшення упорядкування складників у логічні набори (наприклад, для групування усіх перекладів, які використовуються у одному застосунку).

На внутрішньому рівні, у кожному проєкті містяться переклади типових рядків, які типово поширюються на усі інші складники перекладу. Це полегшує роботу, усуваючи потребу у нудному повторному перекладі та ручному перенесенні перекладів між версіями. Поширення перекладів можна вимкнути на рівні *складника* за допомогою *налаштування поширення перекладів*, якщо переклади мають відрізнятися.

Дивись також:

`../devel/integration`

1.2 Реєстрація і профіль користувача

1.2.1 Реєстрація

Здійснювати навігацію проєктами, переглядати переклади або пропонувати переклади за типових налаштувань може будь-хто. Зберігати зміни і отримувати належні авторські права на переклади можуть лише зареєстровані користувачі.

Ви можете зареєструватися, виконавши декілька простих кроків:

1. Заповніть реєстраційну форму вашими реєстраційними даними.
2. Активуйте реєстрацію переходом за посиланням у отриманому вами повідомленні електронної пошти.
3. Якщо хочете, скоригуйте ваш профіль, вибравши відомі вам мови.

1.2.2 Домівка

Після входу до системи ви зможете побачити огляд проєктів і складників, а також дані щодо поступу відповідних перекладів.

Нове в версії 2.5.

Типово, буде показано складники проєктів, за якими ви спостерігаєте, для бажаних для вас мов.

Підказка: Ви можете перемкнутися на інші панелі перегляду за допомогою навігаційних вкладок.

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate' logo and links to 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a user profile section with a 'Your profile' link and a menu of settings: 'Languages', 'Preferences' (highlighted), 'Notifications', 'Account', 'Profile', 'Teams', 'Licenses', 'Audit log', and 'API access'. The 'Preferences' panel is open, showing several settings:

- ☐ Hide completed translations on the dashboard
- Translation editor mode**: Full editor (dropdown)
- Zen editor mode**: Top to bottom (dropdown)
- Number of nearby strings**: 15 (input field)
- Number of nearby strings to show in each direction in the full editor.
- ☒ Show secondary translations in the Zen mode
- ☐ Hide source if a secondary translation exists
- Editor link**: (input field)
- Enter a custom URL to be used as link to the source code. You can use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders.
- Special characters**: (input field)
- You can specify additional special visual keyboard characters to be shown while translating. It can be useful for characters you use frequently, but are hard to type on your keyboard.
- Default dashboard view**:
 - ☒ Watched translations
 - ☐ Suggested translations

A 'Save' button is located at the bottom of the preferences panel.

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

У меню ви побачите такі пункти:

- За допомогою пункту головного меню *Проекти > Перегляд усіх проєктів* можна переглянути стан перекладу для всіх проєктів у екземплярі Weblate.
- Якщо вибрати мову у головному меню *Мови* Weblate покаже стан перекладу усіх проєктів, фільтрований за однією з ваших основних мов.
- За допомогою сторінки *Відстежувані переклади* домашньої сторінки ви зможете переглянути стану тих перекладів, за якими ви стежите, з фільтруванням за вашими основними мовами.

Крім того, за допомогою спадного списку ви можете переглянути будь-яку кількість *списків складників*, наборів складників проєкту, які попередньо налаштовано адміністратором Weblate, див. [Списки складників](#).

Налаштувати вашу типову особисту панель приладів за допомогою розділу *Preferences* параметрів профілю вашого запису користувача.

Примітка: Якщо Weblate налаштовано для одного проєкту за допомогою змінної `SINGLE_PROJECT` у файлі `settings.py` (див. [Налаштування](#)), панель приладів не буде показано, оскільки користувача буде переспрямовано до сторінки єдиного проєкту або складника.

1.2.3 Профіль користувача

Доступ до профілю користувача можна отримати клацанням на піктограмі вашого користувача у верхньому правому куті верхнього меню. Вам слід вибрати пункт *Налаштування*.

Профіль користувача містить параметри вашого облікового запису. Ім'я і адреса електронної пошти використовуються у внесках до системи керування версіями, тому слід точно вказати ці дані.

Примітка: Для всіх наборів мов буде запропоновано лише поточні мови перекладу.

Підказка: Надіслати запит щодо додавання інших мов або додати інші мови, якими ви хочете перекласти проєкт, можна натисканням відповідної кнопки.

Мови

1.2.4 Мова інтерфейсу

Виберіть мову, якою має бути показано інтерфейс користувача.

Перекладено мовами

Виберіть мови, якими ви хочете перекладати, і їх буде запропоновано на головній сторінці списку відстежуваних проєктів, отже вам буде простіше отримати доступ до перекладів кожною з цих мов.

Weblate Dashboard Projects Languages Checks							
Dashboard							
Watched translations 13		Suggested translations 5		Insights		Search	
Translation	Translated	Unfinished	Unfinished words	Unfinished characters	Checks	Suggestions	Comments
WeblateOrg/Android — Czech MIT	76%	3	3	10			
WeblateOrg/Django — Czech GPL-3.0	96%	1	12	69	4		
WeblateOrg/Django — Hebrew GPL-3.0	92%	2	15	80			
WeblateOrg/Django — Hungarian GPL-3.0	69%	8	109	671	1		
WeblateOrg/Djangojs — Hungarian GPL-3.0	96%	2	6	28			
WeblateOrg/Djangojs — Hebrew GPL-3.0	✓						
WeblateOrg/Djangojs — Czech GPL-3.0	✓						
WeblateOrg/Language names — Czech GPL-3.0	✓						
WeblateOrg/Language names — Hungarian GPL-3.0	81%	4	5	32			
WeblateOrg/Language names — Hebrew GPL-3.0	✓						
WeblateOrg/WebblateOrg — Hungarian GPL-3.0	✓						
WeblateOrg/WebblateOrg — Czech GPL-3.0	✓						
WeblateOrg/WebblateOrg — Hebrew GPL-3.0	✓						

Допоміжні мови

Ви можете визначити, переклади якими допоміжними мовами буде додатково показано вам під час перекладу. Приклад наведено на наступному зображенні, де допоміжною мовою вибрано іврит:

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, the breadcrumb 'WeblateOrg / Django / Czech / Translate' is visible. The main area is divided into a left sidebar with navigation links like 'Nearby strings', 'Comments', 'Automatic suggestions', 'Other languages', and 'History'. The central part is the 'Translation' editor, which has input fields for 'Hebrew', 'English', and 'Czech'. The 'Czech' field contains the text 'Soubory'. Below the input fields are buttons: 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. The right sidebar contains a 'Glossary' section and a 'String information' section with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.

1.2.5 Налаштування

Типовий вигляд домівки

За допомогою вкладки *Налаштування* ви можете вибрати, які із доступних сторінок панелі приладів буде типово показано. Якщо ви виберете *Список складників*, вам слід буде вибрати, який саме список складників буде показано, за допомогою спадного списку *Типовий список складників*.

Дивись також:

Списки складників

Посилання на редактор

Типово, буде показано посилання на початковий код у браузері, який налаштовано у *складнику*.

Підказка: Установленням *Посилання на редактор* ви можете використати ваш локальний редактор для відкриття файлу початкового коду з рядками перекладу у системі керування версіями. Ви можете скористатися *розміткою шаблонів*.

Зазвичай, це щось подібне до `editor://open/?file={{filename}}&line={{line}}`.

Дивись також:

Докладні відомості щодо реєстрації нетипових протоколів адрес для редактора можна знайти у [документації до Nette](#).

Спеціальні символи

Додаткові спеціальні символи, які слід включити на *візуальну клавіатуру*.

1.2.6 Сповіщення

Підписатися на різноманітні сповіщення можна за допомогою вкладки *Сповіщення*. Сповіщення для вибраних подій у відстежуваних або адміністрованих проєктах буде надіслано на вашу адресу електронної пошти.

Деякі зі сповіщень надсилатимуться лише для подій у ваших мовах (наприклад, щодо нових рядків для перекладу), інші ж надсилатимуться на рівні складників (наприклад, повідомлення про помилки об'єднання). Ці дві групи сповіщень у параметрах візуально відокремлено.

Ви можете увімкнути або вимкнути сповіщення для відстежуваних та адміністрованих проєктів. Подальше коригування (або вимикання) сповіщень можна виконати на рівні проєктів та складників. Відкрийте сторінку огляду складника і виберіть відповідний варіант з меню *Відстежується*.

Якщо позначено пункт *Автоматичне спостереження за проєктами із внеском*, ви автоматично починатимете стеження за проєктами, у яких переклали хоч рядок. Типове значення залежить від `DEFAULT_AUTO_WATCH`.

Примітка: Ви не отримуватиме сповіщень щодо ваших власних дій.

Підказка: Надсилання сповіщень обмежено. Ви не отримуватимете більше, ніж 1000 повідомлень електронної пошти щодня. Усі сповіщення поза цією кількістю буде відкинуто.

[Powered by Weblate 4.16](#)
[About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

1.2.7 Обліковий запис

За допомогою вкладки *Обліковий запис* ви можете налаштувати базові параметри облікового запису, з'єднати з ним різноманітні служби, якими ви можете скористатися для реєстрації на Weblate, повністю вилучити ваш обліковий запис або отримати дані вашого користувача (див. [Експорт даних користувача Weblate](#)).

Примітка: Список служб залежить від ваших налаштувань Weblate, але він може включати популярні сайти, зокрема GitLab, GitHub, Google, Facebook та Bitbucket, а також інші сайти, які надають доступ до OAuth 2.0.

Webplate

Dashboard

Projects ▾

Languages ▾

Checks ▾

+

▾

▾

...

Your profile

Languages

Preferences

Notifications

Account

Profile

Teams

Licenses

Audit log

API access

Account

Username

testuser

Username may only contain letters, numbers or the following characters: @ . + - _

Full name

Weblate Test

E-mail

weblate@example.org

You can add another e-mail address below.

Commit e-mail

Use account e-mail address

Save

Current user identities

Identity	User ID	Action
 Password	testuser	Change password
 E-mail	weblate@example.org	Disconnect
 Google	weblate@example.org	Disconnect
 GitHub	123456	Disconnect
 Bitbucket	weblate	Disconnect

Add new association

 E-mail

Removal

Account removal deletes all your private data.

Remove my account

User data

You can download all your private data.

Download user data

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.2.8 Профіль

Усі поля на цій сторінці є необов'язковими та можуть бути будь-коли вилучені. Заповнюючи їх, ви надаєте нам згоду на поширення цих даних усюди, де ми показуватимемо дані вашого профілю користувача.

The private commit e-mail will be used instead of your account e-mail in version control commits. Use this to avoid leaking your real e-mail there. Be aware that using different e-mail can disconnect your contributions on other servers (for example your contributions will no longer link to your profile on GitHub). The private e-mail can be turned on site-wide using `PRIVATE_COMMIT_EMAIL_OPT_IN`.

Для кожного користувача може бути показано аватар (залежно від значення параметра `ENABLE_AVATARS`). Зображення буде отримано за допомогою <https://gravatar.com/>.

1.2.9 Ліцензії

1.2.10 Доступ до API

Тут ви можете отримати або скинути ваш жетон доступу до програмного інтерфейсу.

1.2.11 Часопис перевірок

Часопис перевірок зберігає записи дій, які виконувалися із вашим обліковим записом. У ньому можна знайти IP-адреси та записи браузера для кожної важливої дії з вашим обліковим записом. Критичні дії також призводять до надсилання сповіщення на основну адресу електронної пошти.

Дивись також:

Робота за реверсивним проксі-сервером

1.3 Переклад за допомогою Weblate

Дякуємо за вашу зацікавленість у перекладі за допомогою Weblate. Для проєктів можна налаштувати або безпосередній переклад, або переклад прийняттям пропозицій користувачів без облікових записів.

Загалом, передбачено два режими перекладу:

- Проєкт приймає безпосередні переклади
- Проєкт приймає лише пропозиції, які автоматично підтверджуються, якщо за них проголосує визначена кількість користувачів

Щоб дізнатися більше про типові процедури перекладу, будь ласка, зверніться до розділу *Процеси перекладу*.

Варіанти видимості проєкту перекладу:

- Загальнодоступний
- Видимий лише певній групі перекладачів

Дивись також:

Керування доступом, Процеси перекладу

1.3.1 Проекти перекладу

У проєктах перекладу містяться пов'язані складники — ресурси, які пов'язано із однією частиною програмного забезпечення, книгою або проєктом.

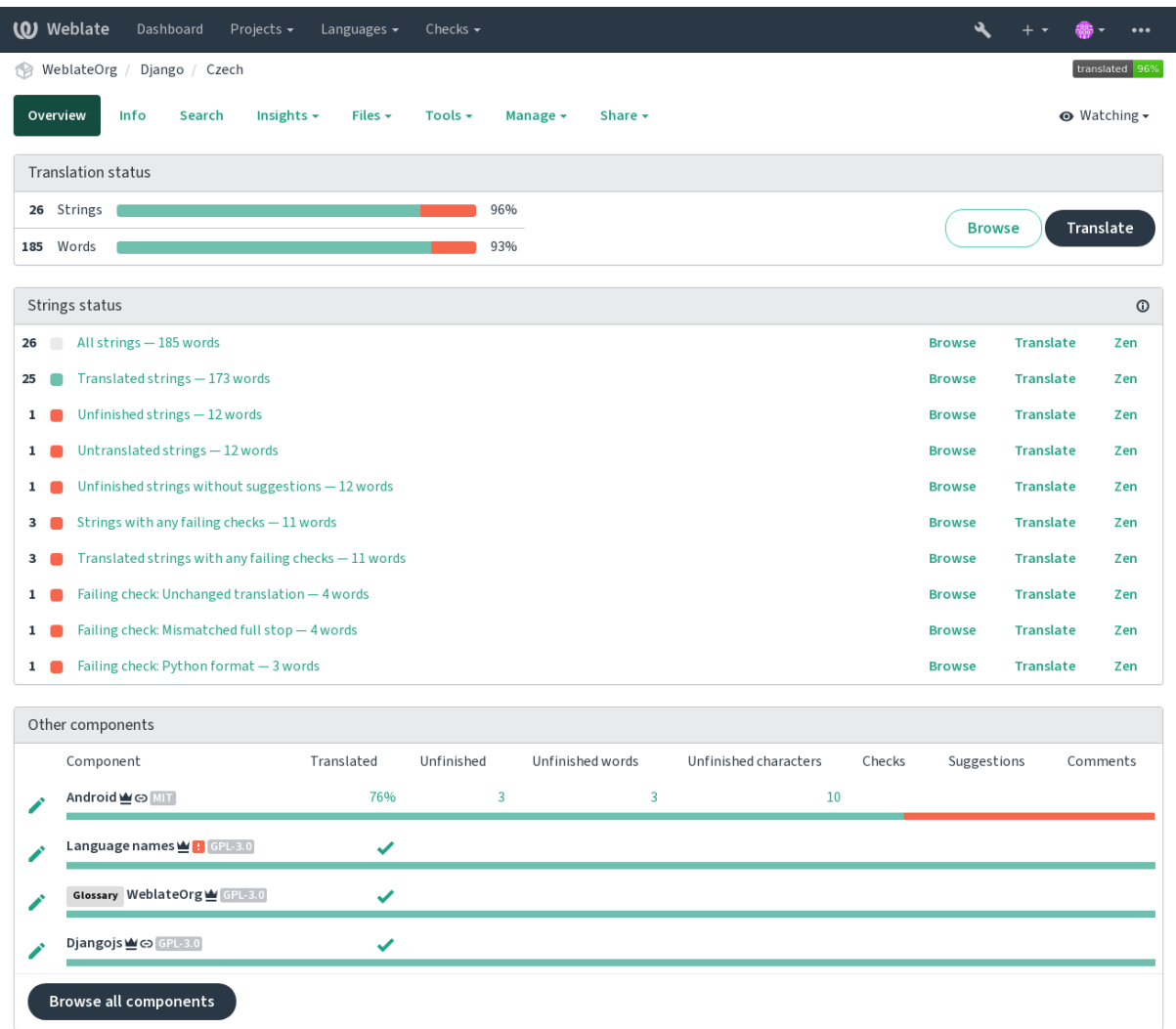
The screenshot shows the Weblate web interface. At the top is a navigation bar with links: Weblate, Dashboard, Projects, Languages, and Checks. Below this is a header for the 'WeblateOrg' project, showing a 'translated 85%' status. A secondary navigation bar includes links: Components (active), Languages, Info, Search, Insights, Files, Tools, Manage, and Share. There is also a 'Not watching' toggle. The main content area displays a table of translation components:

Component	Translated	Unfinished	Unfinished words	Checks	Suggestions	Comments
Android	79%	30	30	3		
Language names	95%	4	5			
Glossary WeblateOrg	✓					

Below the table is a button labeled 'Add new translation component'. At the bottom of the interface, a footer line reads: 'Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate'.

1.3.2 Посилання на переклади

На сторінці складника ви побачите набір посилань на сам переклад. Переклад поділено на окремі частини, зокрема *Неперекладені рядки* або *Незавершені рядки*. Якщо перекладено увесь проєкт і немає помилок, наступним буде лише пункт *Усі рядки*. Крім того, ви можете скористатися полем для пошуку для того, щоб знайти певний рядок або термін.



1.3.3 Пропозиції

Примітка: Справжні права доступу можуть бути різними і залежать від налаштувань Weblate.

Анонімні користувачі можуть лише (типово) надавати пропозиції. Пропозиції можуть надавати і зареєстровані користувачі, якщо вони не певні щодо перекладу, і їм потрібна думка іншого перекладача, який виконає рецензування перекладу.

Сканування пропозицій відбувається щоденно для видалення дублікатів або пропозицій, які відповідають поточному перекладу.

1.3.4 Коментарі

Можна створювати коментарі трьох типів: до перекладу, до початкових рядків або повідомляти про вади у початкових рядках, якщо увімкнено, за допомогою [Увімкнути рецензування початкових даних](#). Виберіть ту область, яка відповідає темі, яку ви хочете обговорити. Коментарі до початкового рядка є чудовим способом надання відгуків щодо початкового рядка, наприклад, якщо ви вважаєте, що його слід змінити, або хочете щось спитати щодо нього.

Ви можете скористатися синтаксисом Markdown в усіх коментарях і згадувати інших користувачів за допомогою записів — @згадка.

Дивись також:

report-source, [Рецензування початкових рядків](#), [Увімкнути рецензування початкових даних](#)

1.3.5 Варіанти

Варіанти використовуються для групування варіантів рядка різної довжини. Бібліотека вашого проєкту може використовувати різні рядки, залежно від розміну екрана або вікна.

Дивись також:

variants, [Варіанти](#)

1.3.6 Мітки

Мітки використовуються для категоризації рядків у межах проєкту. Мітками можна скористатися для подальшого коригування процесу локалізації, наприклад для визначення категорій рядків.

У Weblate використано такі мітки:

Перекладено автоматично

Рядок було перекладено за допомогою можливості [Автоматичний переклад](#).

Початкові рядки потребують рецензування

Рядок позначено для рецензування за допомогою можливості [Рецензування початкових рядків](#).

Дивись також:

labels

1.3.7 Переклад

На сторінці перекладу буде показано початковий рядок і поле для редагування перекладу. Якщо переклад має форми множини, буде показано кілька початкових рядків і поля для редагування перекладу з описом та мітками форм множини, які передбачено для мови перекладу.

Усі спеціальні символи пробілів буде підкреслено червоним і позначено сірими символами. Декілька послідовних пробілів також буде підкреслено червоним, щоб попередити перекладача про потенційну проблему з форматуванням.

На цій сторінці може бути показано різноманітні додаткові дані, більшість з яких походить з початкового коду проєкту (зокрема контекст, коментарі або дані про те, де використовується повідомлення). Над початковим рядком буде показано поля перекладів усіма вторинними мовами, які перекладачі вибрали у налаштуваннях (див. [Допоміжні мови](#)).

Під перекладом буде показано усі пропозиції інших перекладачів. Ви можете прийняти їх (✓), прийняти зі змінами (⇒) або вилучити пропозиції (✗).

Множина

Слова, які змінюють форму відповідно до кількості об'єктів, є словами із формами множини. У кожній мові є власне визначення форм множини. Наприклад, у англійській мові одна форма множини. В однині, наприклад, маємо «car», що неявно вказує на один автомобіль, а у множині — «cars» — два або більше автомобілів або взагалі поняття багатьох автомобілів як іменника. В інших мовах, наприклад, чеській або арабській кількість форм множини є більшою, а правила для визначення форми множини для різних мов є різними.

У Weblate передбачено повну підтримку усіх цих форм у кожній відповідній мові — кожна форма множини має власний переклад. Кількість полів і спосіб їхнього використання у перекладеному застосунку або проєкті залежать від налаштованої формули форм множини. Weblate показує лише основні відомості — докладніший опис можна знайти у [Language Plural Rules](#) — довіднику, який створено консорціумом Unicode.

Дивись також:

Форма множини

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main section is titled 'Translation' and shows the English source string '%(count)s word' and its plural forms. The Czech translation is shown as '%(count)s slovo'. The interface includes a 'Glossary' panel on the right, a 'String information' panel with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', and 'Translation file'. At the bottom, there is a 'Comments' section for users to discuss the translation.

Альтернативні переклади

Нове в версії 4.13.

Примітка: У поточній версії підтримку реалізовано лише для *Багатозначний файл CSV*.

У деяких форматах передбачено можливість декількох перекладів одного рядка. Ви можете додавати альтернативні переклади за допомогою меню *Засоби*. Будь-які порожні альтернативні переклади буде автоматично вилучено під час збереження.

Клавіатурні скорочення

Змінено в версії 2.18: У версії 2.18 клавіатурні скорочення було змінено так, щоб зменшити ймовірність конфлікту із клавіатурними скороченнями програм для перегляду інтернету та типовими клавіатурними скороченнями операційної системи.

Під час перекладу можна користуватися такими клавіатурними скороченнями:

Клавіатурне скорочення	Опис
Alt+Home	Перейти до першого перекладу у поточних результатах пошуку.
Alt+End	Перейти до останнього перекладу у поточних результатах пошуку.
Alt+PageUp або Ctrl+↑ або Alt+↑ або Cmd+↑	Перейти до попереднього перекладу у поточних результатах пошуку.
Alt+PageDown або Ctrl+↓ або Alt+↓ або Cmd+↓	Перейти до наступного перекладу у поточних результатах пошуку.
Ctrl+Enter або Cmd+Enter	Надіслати поточну форму. Під час редагування перекладу це те саме, що натискання кнопки <i>Зберегти та продовжити</i> .
Ctrl+Shift+Enter або Cmd+Shift+Enter	Скасувати позначення перекладу як такого, що потребує редагування, і надіслати його до репозиторію.
Alt+Enter або Option+Enter	Надіслати рядок як пропозицію. Під час редагування перекладу це те саме, що натискання кнопки <i>Запропонувати</i> .
:kbd:`Ctrl+E` або Cmd+E	Фокусує поле редактора перекладу.
Ctrl+U або Cmd+U	Фокусує поле редагування коментаря.
Ctrl+M або Cmd+M	Показує вкладку <i>Автоматичні пропозиції</i> , див. <i>Автоматичні пропозиції</i> .
Від Ctrl+1 до Ctrl+9 або Від Cmd+1 до Cmd+9	Копіює заміник із вказаним номером з початкового рядка.
Ctrl+M` + від :kbd:`1 до 9 або Cmd+M` + від :kbd:`1 до 9	Копіює комп'ютерний переклад із вказаним номером до поточного перекладу.
Ctrl+I` + від :kbd:`1 до 9 або Cmd+I` + від :kbd:`1 до 9	Ігнорувати один запис у списку непройдених перевірок.
Ctrl+J або Cmd+J	Показує вкладку <i>Сусідні рядки</i> .
Ctrl+S або Cmd+S	Передати фокусування до поля пошуку.
Ctrl+O або Cmd+O	Копіювати початковий рядок.
:kbd:`Ctrl+Y` або Cmd+Y	Перемкнути прапорець <i>Потребує редагування</i> .

Візуальна клавіатура

Одразу над полем перекладу буде показано невеличку візуальну клавіатуру. Ця клавіатура може бути корисною для введення пунктуації (клавіатура є локальною для кожної з мов) або символів, яких немає на клавіатурі або які важко ввести.

Показані символи розподілено на три категорії:

- Налаштовані користувачем *спеціальні символи*, які визначено у розділі *Профіль користувача*
- Символи для певної мови, які вбудовано до Weblate (наприклад лапки або специфічні для мов із записом справа ліворуч символи)
- Символи, які налаштовано за допомогою можливості *SPECIAL_CHARS*

The screenshot shows the Weblate web interface for translating a string. The main area displays the source string "Files" and the target string "קבצים" (Kvitsim) in Hebrew. The interface includes a navigation bar at the top with "Weblate", "Dashboard", "Projects", "Languages", and "Checks". Below the navigation bar, the project path is "WeblateOrg / Django / Hebrew / Translate". The main translation area has a "Translation" header, a "Files" input field, and a "Hebrew" output field. A visual keyboard is shown below the output field. At the bottom of the main area are buttons: "Save and continue", "Save and stay", "Suggest", and "Skip". To the right of the main area is a sidebar with "Glossary", "String information", and "Translation file". The "String information" section shows details like "Screenshot context", "Explanation", "Labels", "Flags", "Source string location", "String age", "Source string age", and "Translation file". The footer of the page reads "Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate".

Контекст перекладу

Цей контекстний опис надає пов'язані дані щодо поточного рядка.

Атрибути рядка

Дані, подібні до ідентифікатора повідомлення, контексту (`msgctxt`) або розташування рядка у коді.

Знімки екрана

У Weblate передбачено можливість вивантаження знімків вікон, які надають перекладачам подробиці щодо того, де та як використано рядок, див. [Візуальний контекст для рядків](#).

Сусідні рядки

Показує сусідні повідомлення з файла перекладу. Зазвичай, такі повідомлення використовуються у подібному контексті і можуть бути корисними для підтримання однорідності перекладу.

Інші входження

Якщо повідомлення використовується у декількох частинах перекладу (наприклад, у декількох його складниках), на цій вкладці буде показано усі ці частини, якщо вони мають різні переклади (див. [Неузгодженість](#)). Ви можете вибрати той варіант перекладу, яким хочете скористатися.

Пам'ять перекладів

Погляд на подібні рядки, які вже було перекладено, див. [Пам'ять перекладів](#).

Глосарій

Показує терміни із глосарія проекту, які використано у поточному повідомленні.

Останні зміни

Список осіб, які нещодавно вносили зміни до цього повідомлення за допомогою Weblate.

Проект

Відомості щодо проекту, зокрема настанови для перекладачів або каталог чи посилання на рядок у репозиторії системи керування версіями проекту.

Якщо вам потрібні безпосередні посилання, у форматі перекладу має бути передбачено їхню підтримку.

Журнал перекладів

Типово, усі зміни (якщо це не вимкнено у параметрах складника) зберігаються у базі даних і можуть бути скасовані. Якщо хочете, ви також можете скасувати зміни у базовій системі керування версіями.

Довжина перекладеного рядка

Weblate може обмежувати довжину перекладу у декілька способів, забезпечуючи належні обмеження для рядка перекладу:

- Типовим обмеженням для перекладу є довжина, яка вдесятеро перевищує довжину початкового рядка. Це обмеження може бути вимкнено за допомогою параметра `LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH`. Якщо у вас виникнуть з цим проблеми, ймовірною причиною може бути те, що одномовний переклад помилково налаштовано як двомовний, що спричиняє те, що Weblate вважає ключ перекладу початковим рядком, замість справжнього початкового рядка. Див. [Двомовні і одномовні формати](#), щоб дізнатися більше.
- Максимальна довжина у символах визначається файлом перекладу або прапорцем, див. [Максимальна довжина перекладу](#).
- Максимальний оброблений розмір у пікселях визначається прапорцями, див. [Найбільший розмір перекладу](#).

1.3.8 Автоматичні пропозиції

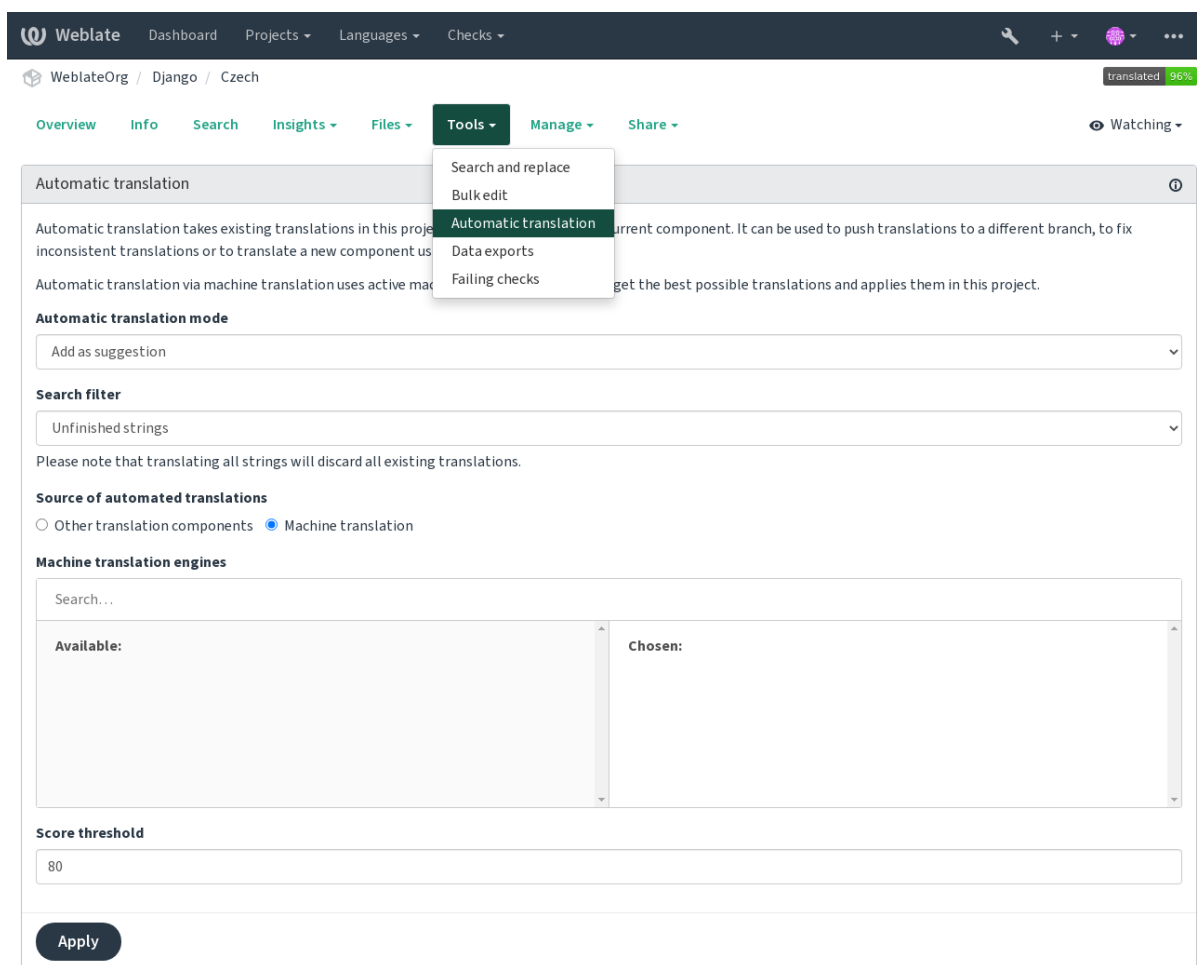
На основі налаштувань і мови перекладу Weblate запропонує вам вибрати один з кількох засобів комп'ютерного перекладу і *Пам'ять перекладів*. Усі комп'ютерні переклади доступні на окремій вкладці кожної сторінки перекладів.

Дивись також:

Список підтримуваних інструментів наведено у розділі *Налаштовування автоматичних пропозицій*.

1.3.9 Автоматичний переклад

Ви можете скористатися автоматичним перекладом, щоб започаткувати переклад на основі зовнішніх джерел. Цей інструмент називається *Автоматичний переклад*. Доступ до нього можна отримати з меню *Знаряддя* після вибору складника і мови:



Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Можна користуватися двома способами дій:

- Скористатися іншими складниками Weblate як джерелом перекладів.
- Скористатися вибраними службами комп'ютерного перекладу із перекладами, які перебувають над пороговим значенням якості.

Ви також можете вибрати, які рядки буде перекладено автоматично.

Попередження: Зверніть увагу на те, що у результаті буде перезаписано наявні переклади, якщо інструмент застосовано із широкими фільтрами, зокрема *Усі рядки*.

Корисно у декількох випадках, зокрема консолідації перекладів між різними складниками (наприклад, застосунком та його вебсайтом) або при започаткуванні перекладу для нового складника на основі наявних перекладів (пам'яті перекладів).

Для автоматично перекладених рядків програма встановлює мітку *Перекладено автоматично*.

Дивись також:

Підтримання синхронізації перекладів у різних складниках

1.3.10 Обмеження частоти

Щоб запобігти зловживанню інтерфейсу, до декількох дій, зокрема пошуку, надсилання контактної форми та перекладу, застосовується обмеження за частотою. Якщо обмеження буде перевищено, користувача буде заблоковано на певний період, протягом якого він не зможе виконувати ці дії.

Типові обмеження і їхнє коригування описано у підручнику з адміністрування, див. *Обмеження частоти*.

1.3.11 Пошук та заміна

Якщо ви хочете змінити термінологію або виконати якесь пакетне виправлення рядків, можете скористатися можливістю *Пошук та заміна*. Відповідний пункт можна знайти у меню *Знаряддя*.

Підказка: Не переймайтеся тим, що можете щось наплутати із рядками. Це двокрокова процедура, протягом якої вам буде показано редаговані рядки, перш ніж буде внесено справжні зміни.

1.3.12 Масове редагування

За допомогою пакетного редагування ви можете виконувати одну дію для декількох рядків одночасно. Ви визначаєте критерій пошуку рядків і дію для виконання, а програма оновлює усі відповідні рядки. Передбачено підтримку таких дій:

- Зміна стану рядка (наприклад, можна підтвердити усі рядки, які очікують на рецензування).
- Коригування прапорців перекладу (див. *Налаштування поведінки за допомогою прапорців*)
- Коригування міток рядків (див. labels)

Підказка: Цей інструмент називається *Пакетне редагування*. Доступ до нього можна отримати за допомогою меню *Знаряддя* проєкту, складника або перекладу.

Дивись також:

Додаток пакетного редагування

1.3.13 Матричний перегляд

Для ефективного порівняння різних мов ви можете скористатися матричним переглядом. Він доступний на сторінці кожного складника у меню *Знаряддя*. Спочатку виберіть усі мови, які ви хочете порівняти, і підтвердьте ваш вибір. Після цього, ви можете клацати на будь-якому перекладі для пришвидшення відкриття і редагування перекладу.

Матричний режим перегляду також є чудовим початковим пунктом для пошуку пропущених перекладів різними мовами і швидкого додавання їх на одну панель перегляду.

1.3.14 Режим дзен

Дзен-редактор можна увімкнути натисканням кнопки *Дзен* у верхній правій частині сторінки під час перекладу складника. Редактор має спрощене компонування без додаткових елементів інтерфейсу, зокрема *Сусідні рядки* та *Глосарій*.

Ви можете вибрати дзен-редактор як типовий за допомогою вкладки *Налаштування* на вашій сторінці *Профіль користувача*. На цій сторінці ви також можете вибрати між режимами списків перекладів *Згори вниз* або *Поруч*, залежно від ваших уподобань.

1.4 Отримання і вивантаження перекладів

Ви можете експортувати файли з перекладу, внести зміни і імпортувати змінені файли до системи. Таким чином, можна працювати поза мережею, а потім надіслати зміну до наявного перекладу. Це працює, навіть якщо початкові рядки було змінено під час перекладу.

Примітка: Доступні варіанти може бути обмежено параметрами *керування доступом*.

1.4.1 Отримання перекладів

З панелі керування проектом або складником, доступні для перекладу файли можна завантажити з меню *Файли*.

Перший варіант — завантажити файл у початковому форматі, оскільки він зберігається у репозиторії. У цьому випадку будь-які очікувані зміни в перекладі виконуються й оновлений файл буде надано без будь-яких перетворень.

Ви також можете отримати початкові дані перекладу в одному із поширених форматів локалізації. Перетворені файли буде скомпоновано із даними, які надаються Weblate, зокрема додатковим контекстом, коментарями або прапорцями. Кілька форматів файлів можна отримати за допомогою меню *Файли* ↓ *Налаштувати завантаження*:

- gettext PO
- XLIFF із розширеннями gettext
- XLIFF 1.1
- TermBase eXchange
- Пам'ять перекладів eXchange
- gettext MO (доступне, лише якщо у перекладі використано gettext PO)
- CSV
- Excel Open XML
- JSON (доступне лише для одномовних перекладів)

- Рядковий ресурс Android (доступне лише для одномовних перекладів)
- Рядки iOS (доступне лише для одномовних перекладів)

Підказка: Дані у перетворених файлах будуть різними через різні можливості форматів файлів. Огляд можливостей наведено у розділі *Можливості типів перекладу*.

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The 'Files' menu is open, showing options: 'Download translation', 'Customize download' (highlighted), and 'Upload translation'. Below the menu, there is a table of download options:

Quick downloads									
26	File in original format as translated in the repository								gettext PO file
26	All strings, converted files enriched with comments; suitable for offline translation	CSV	gettext MO	gettext PO	TBX	TMX	XLIFF 1.1 with gettext extensions	XLIFF 1.1	XLSX
1	Unfinished strings, converted files enriched with comments; suitable for offline translation	CSV	gettext MO	gettext PO	TBX	TMX	XLIFF 1.1 with gettext extensions	XLIFF 1.1	XLSX

Below the table is the 'Customize download' section with a dropdown menu set to 'All strings', a 'File format' section with radio buttons for 'gettext PO' (selected), 'XLIFF 1.1 with gettext extensions', 'XLIFF 1.1', 'TBX', 'TMX', 'gettext MO', 'CSV', and 'XLSX', and a 'Download' button.

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Дивись також:

`GET /api/translations/(string:project)/(string:component)/(string:language)/file/`

1.4.2 Вивантаження перекладів

Після внесення бажаних змін поза мережею скористайтеся пунктом *Вивантажити переклад* з меню *Файли*.

Підтримувані формати файлів

Вивантажувати можна файл у будь-якому підтримуваному форматі. Втім, рекомендуємо скористатися тим самим форматом, який використовується для перекладу, інакше не можна гарантувати належний переклад усіх рядків.

Дивись також:

Підтримувані формати файлів, Отримання і вивантаження перекладів

Способи імпортування

Під час вивантаження файлів перекладу ви можете вибрати значення таких параметрів:

Додати як переклад (**translate**)

Імпортовані рядки буде додано як переклади до наявних рядків. Це найпоширеніший варіант, який є типовим.

Буде використано лише переклади із вивантаженого файла, жодних додаткових даних.

Додати як пропозицію (**suggest**)

Імпортовані рядки буде додано як пропозиції. Виберіть цей варіант, якщо ви хочете вивантажити переклади для рецензування.

Буде використано лише переклади із вивантаженого файла, жодних додаткових даних.

Додати як переклад, що потребує редагування (**fuzzy**)

Імпортовані рядки буде додано як переклади, які потребують редагування. Цей варіант може бути корисним, якщо ви хочете, щоб переклади було використано, але також і рецензовано.

Буде використано лише переклади із вивантаженого файла, жодних додаткових даних.

Замінити наявний файл перекладу (**replace**)

Наявний файл буде замінено новими даними. Це призведе до втрати наявних перекладів. Будьте обережним із використанням цього варіанта.

Оновити початкові рядки (**source**)

Оновлює початкові рядки у двомовному файлі перекладу. За результатами подібна до *Оновити файли po, порівнявши із pot (msgmerge)*.

Підтримку цього варіанта дій передбачено лише для деяких форматів файлів.

Додати нові рядки (**add**)

Додає нові рядки до перекладу. Пропускає ті, які вже існують.

Якщо ви одночасно хочете додати нові рядки і оновити наявні переклади, вивантажте файл ще раз із позначенням пункту *Додати як переклад*.

Цим пунктом можна скористатися, лише якщо увімкнено *Керування рядками*.

З вивантаженого файла буде використано лише початковий рядок, переклад і ключ (контекст).

Дивись також:

```
POST /api/translations/(string:project)/(string:component)/(string:language)/file/
```

Обробка конфліктів

Визначає спосіб обробки вивантажених рядків, які вже перекладено.

Рядки, які потребують редагування

Також передбачено пункт для визначення способу обробки рядків у імпортованому файлі, які потребують редагування. Такі рядки може бути оброблено у три способи: «Не імпортувати», «Імпортувати як рядок, який потребує редагування» або «Імпортувати як перекладений».

Перевизначення авторства

Якщо ви маєте права доступу адміністратора, ви можете вказати авторство вивантаженого файла. Це може бути корисним, якщо ви отримали файл у якийсь спосіб і хочете об'єднати його із наявними перекладами, одночасно зберігши авторські права справжнього перекладача.

1.5 Глосарій

До кожного проекту можна включити один або декілька глосаріїв для зберігання термінології. Глосарії спрощують підтримання однорідності перекладу.

Глосаріями для окремих мов можна керувати окремо, але усі глосарії зберігаються разом як єдиний складник, який допомагає адміністраторам проекту та перекладачам декількома мовами підтримувати міжмовну однорідність перекладу. Терміни з поточного перекладеного рядка буде показано на бічній панелі редактора перекладів.

1.5.1 Керування глосаріями

Змінено в версії 4.5: У нових версіях глосарії є звичайними складниками перекладу — ви можете скористатися усіма можливостями Weblate для роботи з ними — коментування, зберігання у віддаленому репозиторії або додавання пояснень.

Ви можете зробити будь-який складник глосарієм за допомогою пункту *Використовувати як глосарій*. У одному проєкті може бути декілька глосаріїв.

Під час створення проєкту автоматично створюється порожній глосарій. Глосарії спільні для всіх складників одного проєкту. Ви можете налаштувати систему так, щоб глосарій був спільним для кількох проєктів за допомогою *Спільне використання даних у проєктах* у відповідному складнику глосарія.

Складник глосарія виглядає як будь-який інший складник у Weblate, але має додаткову кольорову мітку:

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, the breadcrumb is 'WeblateOrg / Glossary WeblateOrg / Czech'. The main header has tabs: 'Overview' (selected), 'Info', 'Search', 'Insights', 'Files', 'Tools', and 'Share'. On the right, there's a 'Not watching' toggle.

The 'Overview' section shows the 'Translation status' with two progress bars: '2 Strings' at 100% and '3 Words' at 100%. There are buttons for 'Add new glossary term', 'Browse', and 'Translate'.

The 'Strings status' section shows two items: 'All strings — 3 words' and 'Translated strings — 3 words', each with 'Browse', 'Translate', and 'Zen' links.

The 'Other components' section has a table with columns: Component, Translated, Unfinished, Unfinished words, Checks, Suggestions, and Comments. It lists 'Django' (96% translated, 1 unfinished word, 12 unfinished words, 3 checks) and 'Language names' (100% translated). A 'Browse all components' button is at the bottom.

At the very bottom, there's a footer: 'Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate'.

Ви можете здійснювати навігацію усіма термінами глосарія:

The screenshot shows the 'Browse' view of the glossary. The breadcrumb is 'WeblateOrg / Glossary WeblateOrg / Czech / Browse'. There are navigation buttons (back, forward, first, last) and a search bar. A button 'Add new glossary term' is on the right.

The main content area shows a list of terms with two columns: 'English' and 'Czech'. The first term is 'machine translation' (English) and 'strojový překlad' (Czech). The second term is 'project' (English) and 'projekt' (Czech).

At the bottom, there's a footer: 'Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate'.

або редагувати їх як будь-які переклади.

1.5.2 Терміни глосарія

Терміни глосарія перекладають так само як виконують звичайні переклади. Ви можете увімкнути додаткові можливості за допомогою меню *Знаряддя* для всіх термінів.

The screenshot shows the Weblate web interface for managing a glossary. The main area is titled 'Glossary term' and contains input fields for 'English' (with the value 'project') and 'Czech' (with the value 'projekt'). Below these is a 'Needs editing' checkbox and an 'Explanation' field. At the bottom of the main editor are buttons: 'Save and continue', 'Save and stay', 'Suggest', 'Skip', and 'Tools'. The 'Tools' dropdown menu is open, displaying several actions: 'Delete string', 'Mark as untranslatable', 'Mark as forbidden translation', 'Mark as terminology', and 'Add variant of this string'. To the right of the main editor, there are two side panels: 'Glossary' showing a list of terms and 'String information' showing details about the current string, including its age and the translation file status.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Терміни, які не слід перекладати

Нове в версії 4.5.

Позначення певних перекладів термінів глосарія придатними лише для читання пакетним редагуванням, введенням прапорця або за допомогою пункту меню *Знаряддя* ↓ *Позначити не перекладним*, означає, що терміни не можна перекладати. Скористайтеся цим для зареєстрованих торговельних марок або інших термінів, які не слід змінювати під час перекладу. Відповідні терміни буде візуально підсвічено на бічній панелі глосарія.

Дивись також:

Налаштування поведінки за допомогою прапорців

Заборонені переклади

Нове в версії 4.5.

Позначення певних перекладів термінів глосарія як заборонених пакетним редагуванням, введенням прапорця або за допомогою пункту меню *Знаряддя* ↓ *Позначити як заборонений переклад*, означає, що терміни **не** слід використовувати. Скористайтеся цим для прояснення перекладу, коли певні слова є неоднозначними або можуть мати неочікувані значення.

Дивись також:

Налаштування поведінки за допомогою прапорців

Термінологія

Нове в версії 4.5.

Позначення певних перекладів термінів глосарія як термінології пакетним редагуванням, введенням прапорця або за допомогою пункту меню *Знаряддя* ↓ *Позначити як термінологію*, додає записи термінів до усіх мов у глосарії. Скористайтеся цим для важливих термінів, переклад яких слід обміркувати та зберігати значення в усіх мовах.

Дивись також:

Налаштування поведінки за допомогою прапорців

Варіанти

Варіанти є загальним способом об'єднання груп рядків. На бічній панелі глосарія під час перекладу буде показано список усіх варіантів терміна.

Підказка: Ви можете скористатися цим для додавання скорочених варіантів або коротших виразів терміна.

Дивись також:

variants

1.6 Перевірки і виправлення

Перевірки якості допомагають знайти типові помилки перекладачів, що забезпечує якість перекладу. Підсумками перевірок можна нехтувати, якщо ви впевнені у їхній хибності.

Одразу після надсилання перекладу, який не проходить перевірки, Weblate покаже користувачеві повідомлення про помилку:

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate
 translated 96%

The translation has been saved, however there are some newly failing checks: Missing plurals, Python format

1 / 1

Custom search

'%(count)s word'

Position

1

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

Czech, Few

několik slov

Czech, Many

%(count)s slov

Plural formula:

(n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2

Needs editing

Save and continue

Save and stay

Suggest

Skip

Nearby strings

Comments

Automatic suggestions

Other languages

History

New comment

Comment on this string for fellow translators and developers to read.

Scope

Translation comment, discussions with other translators

Is your comment specific to this translation, or generic for all of them?

New comment

You can use Markdown and mention users by @username.

Save

Things to check

Python format

Following format strings are missing:

%(count)s

Dismiss

For all languages

Missing plurals

Some plural forms are untranslated

Dismiss

For all languages

Glossary

English

Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

10 seconds ago

Source string age

10 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/django.po, string 5

pending

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

1.6. Перевірки і виправлення

27

1.6.1 Автоматичні виправлення

Окрім *перевірок*, Weblate може автоматично виправляти деякі типові помилки у рядках перекладу. Користуйтеся цим обережно, щоб не наробити помилок.

Дивись також:

AUTOFIX_LIST

1.6.2 Перевірки якості

Weblate застосовує до рядків широкий діапазон перевірок якості. У наступному розділі докладно описано усі ці перевірки. Також передбачено специфічні для мов перевірки. Якщо у звітах вади ви знайшли помилку, повідомте нам про неї.

Дивись також:

CHECK_LIST, *Налаштування поведінки за допомогою прапорців*

1.6.3 Перевірки перекладів

Виконуються після кожної зміни у перекладі, допомагаючи перекладачам підтримувати високу якість перекладів.

Розмітка BBCode

Підсумок

ВВ-код у перекладі не збігається з ВВ-кодом у джерелі

Область

перекладені рядки

Клас перевірки

`weblate.checks.markup.BBCodeCheck`

Перевірити ідентифікатор

`bbcode`

Прапорець для нехтування

`ignore-bbcode`

BBCode використовує просту розмітку, наприклад, надає змогу позначити важливі частини повідомлення напівжирним або курсивним шрифтом.

Ця перевірка забезпечує наявність розмітки у перекладі.

Примітка: Спосіб виявлення BBCode у поточній версії є доволі простим, тому ця перевірка може помилково повідомляти про хиби у перекладі.

Послідовне повторення слів

Нове в версії 4.1.

Підсумок

Текст містить те саме слово двічі поспіль:

Область

перекладені рядки

Клас перевірки

`weblate.checks.duplicate.DuplicateCheck`

Перевірити ідентифікатор

`duplicate`

Прапорець для нехтування

`ignore-duplicate`

Перевіряє переклад на відсутність послідовного дублювання слів. Дублювання, зазвичай, вказує на помилку у перекладі.

Підказка: До цієї перевірки включено специфічні для мови правила для уникнення хибного спрацьовування. Якщо у вашому випадку рядок без помилок не проходить перевірки, повідомте про це нас. Див. розділ *Надсилання звітів щодо вад у Weblate*.

Не слідує глосарію

Нове в версії 4.5.

Підсумок

Переклад не слідує термінам, які визначено у глосарії.

Область

перекладені рядки

Клас перевірки

`weblate.checks.glossary.GlossaryCheck`

Перевірити ідентифікатор

`check_glossary`

Прапорець для вмикання

`check-glossary`

Прапорець для нехтування

`ignore-check-glossary`

Цю перевірку має бути увімкнено за допомогою прапорця `check-glossary` (див. *Налаштування поведінки за допомогою прапорців*). Будь ласка, зважте на таке, перш ніж її вмикати:

- Виконує точну перевірку збігу рядків. У глосарії мають міститися терміни в усіх варіантах.
- Зіставлення всіх рядків з глосарієм ресурсовитратна операція. Вона сповільнює будь-які дії у Weblate, які стосуються виконання перевірок, подібних до імпортування рядків або перекладу.

Дивись також:

Глосарій, Налаштування поведінки за допомогою прапорців, Прапорці перекладу

Подвійний пробіл

Підсумок

Переклад містить подвійний пробіл

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.DoubleSpaceCheck`

Перевірити ідентифікатор

`double_space`

Прапорець для нехтування

`ignore-double-space`

Перевіряє, чи немає у перекладі подвійних пробілів, щоб уникнути хибних спрацювань інших пов'язаних із пробілами перевірок.

Перевірка вважається пройденою, якщо буде виявлено подвійний пробіл у початковому рядку, що означає, що подвійний пробіл було використано навмисно.

Форматовані рядки

Перевіряє, чи є однаковим форматування у початковому рядку і перекладі. Пропускання рядків форматування у перекладі, зазвичай, спричиняє серйозні проблеми, тому форматування у рядках перекладів має, зазвичай, збігатися із форматуванням у початковому рядку.

У Weblate передбачено підтримку рядків форматування декількох мов програмування. Перевірка не вмикається автоматично, а лише якщо рядок позначено відповідним прапорцем (наприклад *c-format* для форматування C). Gettext додає такі прапорці автоматично, але вам, ймовірно, доведеться додавати прапорці вручну для файлів в інших форматах, або якщо ваші файли PO не створено за допомогою **xgettext**.

Це можна зробити на рівні модуля перекладу (див. *Додаткові відомості щодо початкових рядків*) або у *складнику*. Визначити перевірку на рівні складника простіше, але це може призвести до помилкових спрацювань, якщо рядок не позначено як форматований, але у ньому все ж використано синтаксис рядків форматування.

Підказка: Якщо специфічна перевірка форматування є недоступною у Weblate, ви можете скористатися загальною *перевіркою на символи-замінники*.

Окрім перевірки, Weblate підсвітить рядки форматування для спрощення вставлення їх до рядків перекладу:

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The main panel displays the translation of the English string '%(count)s word'. The Czech translations are shown for different grammatical forms: Singular, Plural, Czech, One, Czech, Few, and Czech, Many. The right sidebar contains a 'Glossary' section, 'String information' (including screenshot context, explanation, labels, flags, source string location, string age, and translation file), and a 'Nearby strings' section. The bottom of the interface includes a footer with navigation links and a 'Browse all string changes' link.

Рядок інтерполяції AngularJS

Підсумок

Рядки інтерполяції AngularJS не відповідають джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.angularjs.AngularJSInterpolationCheck`

Перевірити ідентифікатор

`angularjs_format`

Прапорець для вмикання

`angularjs-format`

Прапорець для нехтування

`ignore-angularjs-format`

Приклад рядка іменованого форматування

`Your balance is {{amount}} {{ currency }}`

Дивись також:

Форматовані рядки, *Інтерполяція тексту AngularJS*

Формат C

Підсумок

Рядок, формату C, не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.CFormatCheck`

Перевірити ідентифікатор

`c_format`

Прапорець для вмикання

`c-format`

Прапорець для нехтування

`ignore-c-format`

Приклад рядка простого форматування

`There are %d apples`

Приклад рядка позиційного форматування

`Your balance is %1$d %2$s`

Дивись також:

Форматовані рядки,

Рядки форматування C, форматування printf C

Формат C#

Підсумок

Рядок формату C# не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.CSharpFormatCheck`

Перевірити ідентифікатор

`c_sharp_format`

Прапорець для вмикання

`c-sharp-format`

Прапорець для нехтування

`ignore-c-sharp-format`

Приклад рядка позиційного форматування

`There are {0} apples`

Дивись також:

[Форматовані рядки](#), [Форматування рядків C#](#)

Літерали шаблонів ECMAScript

Підсумок

Літерали шаблонів ECMAScript не відповідають джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.ESTemplateLiteralsCheck`

Перевірити ідентифікатор

`es_format`

Прапорець для вмикання

`es-format`

Прапорець для нехтування

`ignore-es-format`

Приклад інтерполяції

There are `${number}` apples

Дивись також:

[Форматовані рядки](#), [Літерали шаблонів](#)

i18next інтерполяція

Нове в версії 4.0.

Підсумок

Інтерполяція i18next не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.I18NextInterpolationCheck`

Перевірити ідентифікатор

`i18next_interpolation`

Прапорець для вмикання

`i18next-interpolation`

Прапорець для нехтування

`ignore-i18next-interpolation`

Приклад інтерполяції

There are `{{number}}` apples

Приклад вкладеності

There are `$t(number)` apples

Дивись також:

[Форматовані рядки](#), [Інтерполяція i18next](#)

MessageFormat ICU

Нове в версії 4.9.

Підсумок

Помилки синтаксису та/або невідповідності заповнювача в рядках ICU MessageFormat.

Область

перекладені рядки

Клас перевірки

`weblate.checks.icu.ICUMessageFormatCheck`

Перевірити ідентифікатор

`icu_message_format`

Прапорець для вмикання

`icu-message-format`

Прапорець для нехтування

`ignore-icu-message-format`

Приклад інтерполяції

`There {number, plural, one {is one apple} other {are # apples}}.`

У цій перевірці передбачено підтримку чистих повідомлень MessageFormat ICU, а також ICU із простими тегами XML. Ви можете налаштувати поведінку цієї перевірки за допомогою `icu-flags:*`, або доповнивши підтримку XML, або вимкнувши певні підлеглі перевірки. Наприклад, вказаний нижче прапорець вмикає підтримку XML і вимикає перевірку підповідомлень із формами множини:

```
icu-message-format, icu-flags:xml:-plural_selectors
```

<code>xml</code>	Увімкнути підтримку простих тегів XML. Типово, обробку тегів XML оптимізовано. Окремі символи < буде проігноровано, якщо вони не є природною частиною тегу.
<code>strict-xml</code>	Увімкнути підтримку строгої обробки тегів XML. Усі символи < має бути екрановано, якщо вони не є частиною тегу.
<code>-highlight</code>	Вимкнути підсвічування заміників у редакторі.
<code>-require_oth</code>	Вимкнути вимогу щодо того, що підповідомлення повинні належати до селектора <code>other</code> .
<code>-submessage_</code>	Пропустити перевірку того, чи селектори підповідомлень відповідають джерелу.
<code>-types</code>	Пропустити перевірку відповідності типів заміників джерелу.
<code>-extra</code>	Пропустити перевірку того, чи є замітники, яких немає у початковому рядку.
<code>-missing</code>	Пропустити перевірку того, чи не пропущено замітники, які були у початковому рядку.

Крім того, якщо не увімкнено `strict-xml`, але увімкнено `xml`, ви можете скористатися прапорцем `icu-tag-prefix:ПРЕФІКС`, щоб вказати, що усі теги XML мають починатися із вказаного рядка. Наприклад, вказаний нижче прапорець призводить до того, що відповідність тегами XML встановлюється, лише якщо вони починаються з `<x::`

```
icu-message-format, icu-flags:xml, icu-tag-prefix:"x:"
```

Тобто, відповідність буде встановлено для `<x:link>натисніть тут</x:link>`, але не для `це`.

Дивись також:

Синтаксис ICU MessageFormat, Форматовані рядки, ICU: форматування повідомлень, FormatJS: синтаксис повідомлень

Формат Java

Підсумок

Рядок формату Java не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.JavaFormatCheck`

Перевірити ідентифікатор

`java_printf_format`

Прапорець для вмикання

`java-printf-format`

Прапорець для нехтування

`ignore-java-printf-format`

Приклад рядка простого форматування

There are %d apples

Приклад рядка позиційного форматування

Your balance is %1\$d %2\$s

Змінено в версії 4.14: Зазвичай, перемикається прапорцем `java-format`, було змінено для сумісності із GNU gettext.

Дивись також:

Форматовані рядки, *Рядки форматування Java*

Формат Java Message

Підсумок

Рядок формату Java Message не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.JavaMessageFormatCheck`

Перевірити ідентифікатор

`java_format`

Прапорець для безумовного вмикання

`java-format`

Прапорець для вмикання автовиявлення

`auto-java-messageformat` вмикає перевірку, лише якщо у початковому рядку є рядок форматування

Прапорець для нехтування

`ignore-java-format`

Приклад рядка позиційного форматування

There are {0} apples

Змінено в версії 4.14: Зазвичай, перемикається прапорцем `java-messageformat`, було змінено для сумісності із GNU gettext.

Ця перевірка визначає, чи є рядок форматування коректним для класу `Java MessageFormat`. Окрім встановлення відповідності рядків форматування у фігурних дужках, також буде виконано перевірку відповідності

одинарних лапок специфічного призначення. Одинарні лапки має бути записано як ' '. Якщо у лапок немає пари, обробник вважатиме їх початком цитування і не показуватиме при обробці рядка.

Дивись також:

Форматовані рядки, [MessageFormat Java](#)

Формат JavaScript

Підсумок

Рядок, формату JavaScript, не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.JavaScriptFormatCheck`

Перевірити ідентифікатор

`javascript_format`

Прапорець для вмикання

`javascript-format`

Прапорець для нехтування

`ignore-javascript-format`

Приклад рядка простого форматування

There are %d apples

Дивись також:

Форматовані рядки, [Рядки форматування JavaScript](#)

Формат Lua

Підсумок

Рядок, формату Lua, не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.LuaFormatCheck`

Перевірити ідентифікатор

`lua_format`

Прапорець для вмикання

`lua-format`

Прапорець для нехтування

`ignore-lua-format`

Приклад рядка простого форматування

There are %d apples

Дивись також:

Форматовані рядки, [Рядки форматування Lua](#)

Форматування Object Pascal

Підсумок

Рядок форматування Object Pascal не збігається із джерелом

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.ObjectPascalFormatCheck`

Перевірити ідентифікатор

`object_pascal_format`

Прапорець для вмикання

`object-pascal-format`

Прапорець для нехтування

`ignore-object-pascal-format`

Приклад рядка простого форматування

There are %d apples

Дивись також:

Форматовані рядки, рядки форматування Object Pascal рядки форматування Free Pascal рядки форматування Delphi

Відсоток заповнювачів

Нове в версії 4.0.

Підсумок

Відсоток заповнювачів не відповідає джерелам

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.PercentPlaceholdersCheck`

Перевірити ідентифікатор

`percent_placeholders`

Прапорець для вмикання

`percent-placeholders`

Прапорець для нехтування

`ignore-percent-placeholders`

Приклад рядка простого форматування

There are %number% apples

Дивись також:

Форматовані рядки,

Формат Perl

Підсумок

Рядок формату Perl не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.PperlFormatCheck`

Перевірити ідентифікатор

`perl_format`

Прапорець для вмикання

`perl-format`

Прапорець для нехтування

`ignore-perl-format`

Приклад рядка простого форматування

There are %d apples

Приклад рядка позиційного форматування

Your balance is %1\$d %2\$s

Дивись також:

Форматовані рядки, [Perl sprintf](#), [Рядки форматування Perl](#)

Формат PHP

Підсумок

Рядок, формату PHP, не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.PHPFormatCheck`

Перевірити ідентифікатор

`php_format`

Прапорець для вмикання

`php-format`

Прапорець для нехтування

`ignore-php-format`

Приклад рядка простого форматування

There are %d apples

Приклад рядка позиційного форматування

Your balance is %1\$d %2\$s

Дивись також:

Форматовані рядки, [Документація PHP з sprintf](#), [Рядки форматування PHP](#)

Формат дужок Python

Підсумок

Рядок, формату Python brace, не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.PythonBraceFormatCheck`

Перевірити ідентифікатор

`python_brace_format`

Прапорець для вмикання

`python-brace-format`

Прапорець для нехтування

`ignore-python-brace-format`

Простий рядок форматування

There are {} apples

Приклад рядка іменованого форматування

Your balance is {amount} {currency}

Дивись також:

Форматовані рядки, *Форматування із фігурними дужками у Python*, *Рядки форматування у Python*

Формат Python

Підсумок

Рядок, формату Python, не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.PythonFormatCheck`

Перевірити ідентифікатор

`python_format`

Прапорець для вмикання

`python-format`

Прапорець для нехтування

`ignore-python-format`

Простий рядок форматування

There are %d apples

Приклад рядка іменованого форматування

Your balance is %(amount)d %(currency)s

Дивись також:

Форматовані рядки, *Форматування рядків у Python*, *Рядки форматування у Python*

Формат Qt

Підсумок

Рядок формату Qt не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.qt.QtFormatCheck`

Перевірити ідентифікатор

`qt_format`

Прапорець для вмикання

`qt-format`

Прапорець для нехтування

`ignore-qt-format`

Приклад рядка позиційного форматування

There are %1 apples

Дивись також:

Форматовані рядки, [Qt QString::arg\(\)](#)

Формат Qt у множині

Підсумок

Рядок формату Qt у множині не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.qt.QtPluralCheck`

Перевірити ідентифікатор

`qt_plural_format`

Прапорець для вмикання

`qt-plural-format`

Прапорець для нехтування

`ignore-qt-plural-format`

Приклад рядка форматування форм множини

There are %Ln apple(s)

Дивись також:

Форматовані рядки, [Підручник з i18n Qt](#)

Формат Ruby

Підсумок

Рядок формату Ruby не збігається з джерелом

Область

перекладені рядки

Клас перевірки

`weblate.checks.ruby.RubyFormatCheck`

Перевірити ідентифікатор

`ruby_format`

Прапорець для вмикання

`ruby-format`

Прапорець для нехтування

`ignore-ruby-format`

Приклад рядка простого форматування

There are %d apples

Приклад рядка позиційного форматування

Your balance is %1\$f %2\$s

Приклад рядка іменованого форматування

Your balance is %+2<amount>f %<currency>s

Рядок із іменованим шаблоном

Your balance is %{amount} %{currency}

Дивись також:

Форматовані рядки, `Ruby Kernel#sprintf`

Формат Scheme

Підсумок

Рядок форматування Scheme не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.SchemeFormatCheck`

Перевірити ідентифікатор

`scheme_format`

Прапорець для вмикання

`scheme-format`

Прапорець для нехтування

`ignore-scheme-format`

Приклад рядка простого форматування

There are ~d apples

Дивись також:

Форматовані рядки, `Srfi 28`, формат `Chicken Scheme`, форматове виведення у `Guile Scheme`

Форматування I18n Vue

Підсумок

Форматування I18n Vue не збігається із початковим рядком

Область

перекладені рядки

Клас перевірки

`weblate.checks.format.VueFormattingCheck`

Перевірити ідентифікатор

`vue_format`

Прапорець для вмикання

`vue-format`

Прапорець для нехтування

`ignore-vue-format`

Іменоване форматування

There are {count} apples

Форматування i18n Rails

There are %{count} apples

Пов'язані повідомлення локалі

`@:message.dio @:message.the_world!`

Дивись також:

Форматовані рядки, Форматування I18n Vue, пов'язані повідомлення локалі i18n Vue

Вже перекладено

Підсумок

Цей рядок перекладено раніше

Область

усі рядки

Клас перевірки

`weblate.checks.consistency.TranslatedCheck`

Перевірити ідентифікатор

`translated`

Прапорець для нехтування

`ignore-translated`

Означає, що рядок вже було перекладено. Таке може трапитися, якщо переклади було скасовано у системі керування версіями або втрачено у інший спосіб.

Неузгодженість

Підсумок

Цей рядок має більше одного перекладу в цьому проєкті або не перекладений у деяких складниках.

Область

усі рядки

Клас перевірки

`weblate.checks.consistency.ConsistencyCheck`

Перевірити ідентифікатор

inconsistent

Прапорець для нехтування

ignore-inconsistent

Weblate шукає різні переклади однакових рядків у усьому проєкті, щоб забезпечити однорідність перекладів.

Ця перевірка вважається не пройденою, якщо у межах проєкту буде знайдено різні переклади одного рядка. Неоднорідність буде показано серед інших даних щодо перекладу. Знайти інші переклади рядка можна на вкладці *Інші входження*.

Цю перевірку буде застосовано до всіх складників у проєкті, для яких увімкнено [Дозволити поширення перекладу](#).

Підказка: З міркувань швидкодії перевірка може шукати не усі неузгодженості — кількість відповідників обмежено.

Примітка: Перевірка також вважається не пройденою, якщо рядок перекладено в одному складнику проєкту і не перекладено в іншому. Нею можна скористатися як швидким способом обробки рядків, які не перекладено у деяких складниках: достатньо просто натиснути кнопку *Використати цей переклад*, яку показано для кожного з рядків на вкладці *Інші входження*.

Ви можете скористатися додатком [Автоматичний переклад](#) для автоматизації нових доданих рядків, які вже було перекладено в іншому складнику.

Дивись також:

[Підтримання синхронізації перекладів у різних складниках](#)

Використано літеру кашида

Нове в версії 3.5.

Підсумок

Декоративні букви кашиди не повинні використовуватися

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.KashidaCheck`

Перевірити ідентифікатор

kashida

Прапорець для нехтування

ignore-kashida

Декоративні літери кашиди не повинні використовуватися. Такі літери також називають татвілами.

Дивись також:

[Кашида у Вікіпедії](#)

Посилання Markdown

Нове в версії 3.5.

Підсумок

Посилання Markdown не відповідають джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.markup.MarkdownLinkCheck`

Перевірити ідентифікатор

`md-link`

Прапорець для вмикання

`md-text`

Прапорець для нехтування

`ignore-md-link`

Посилання Markdown не відповідають джерелу.

Дивись також:

[Посилання Markdown](#)

Markdown посилання

Нове в версії 3.5.

Підсумок

Посилання Markdown не збігаються з джерелом

Область

перекладені рядки

Клас перевірки

`weblate.checks.markup.MarkdownRefLinkCheck`

Перевірити ідентифікатор

`md-reflink`

Прапорець для вмикання

`md-text`

Прапорець для нехтування

`ignore-md-reflink`

Посилання Markdown не збігаються з джерелом.

Дивись також:

[Посилання у Markdown](#)

Синтаксис Markdown

Нове в версії 3.5.

Підсумок

Синтаксис Markdown не відповідає джерелу

Область

перекладені рядки

Клас перевірки

`weblate.checks.markup.MarkdownSyntaxCheck`

Перевірити ідентифікатор

`md-syntax`

Прапорець для вмикання

`md-text`

Прапорець для нехтування

`ignore-md-syntax`

Синтаксис Markdown не відповідає джерелу

Дивись також:

[Елементи span у Markdown](#)

Максимальна довжина перекладу

Підсумок

Переклад не перевищує заданої довжини

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.MaxLengthCheck`

Перевірити ідентифікатор

`max-length`

Прапорець для вмикання

`max-length`

Прапорець для нехтування

`ignore-max-length`

Перевіряє, чи прийнятною є довжина перекладів і чи вкладаються вони у доступне місце. У цій перевірці враховується лише довжина перекладу у символах.

На відміну від інших перевірок, прапорець слід встановити як пару `ключ:значення`, наприклад `max-length:100`.

Підказка: Ця перевірка використовує кількість символів — це значення може бути не найкращим варіантом, якщо для показу тексту використовується пропорційний шрифт. Справжню довжину тексту можна оцінити за допомогою перевірки *Найбільший розмір перекладу*.

Крім того, може стати у пригоді прапорець `replacements`: для розгортання заміників перед перевіркою рядка.

Якщо також використано прапорець `xml-text`, в обчисленнях довжини буде проігноровано теги XML.

Найбільший розмір перекладу

Підсумок

Перекладений текст не повинен перевищувати заданий розмір

Область

перекладені рядки

Клас перевірки

`weblate.checks.render.MaxSizeCheck`

Перевірити ідентифікатор

`max-size`

Прапорець для вмикання

`max-size`

Прапорець для нехтування

`ignore-max-size`

Нове в версії 3.7.

Перекладений текст не повинен перевищувати заданий розмір. Програма обробляє текст із перенесенням рядків і перевіряє, чи вкладається він у вказані обмеження.

Для перевірки потрібні два параметри — максимальна ширина і максимальна кількість рядків. Якщо кількість рядків не вказано, буде перевірено однорядковий текст.

Ви також можете налаштувати використаний шрифт за допомогою інструкції `font-*` (див. [Налаштування поведінки за допомогою прапорців](#)). Наприклад, вказані нижче прапорці перекладу встановлюють, що запис тексту шрифтом `ubuntu` розміру 22 має вкладатися у два рядки і ширину у 500 пікселів:

```
max-size:500:2, font-family:ubuntu, font-size:22
```

Підказка: Вам варто встановити інструкції `font-*` на рівні [складника](#), щоб той самий шрифт було використано для всіх рядків складника. Ви можете перевизначити значення на рівні рядка, якщо вам потрібно виконати специфічну перевірку для окремого рядка.

Крім того, може стати у пригоді прапорець `replacements`: для розгортання заміників перед перевіркою рядка.

Якщо також використано прапорець `xml-text`, в обчисленнях довжини буде проігноровано теги XML.

Дивись також:

[Керування шрифтами](#), [Налаштування поведінки за допомогою прапорців](#), [Максимальна довжина перекладу](#)

Не збіглися \n

Підсумок

Кількість літералів `\n` у перекладі не збігається з джерелом

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.EscapedNewlineCountingCheck`

Перевірити ідентифікатор

`escaped_newline`

Прапорець для нехтування

`ignore-escaped-newline`

Usually escaped newlines are important for formatting program output. Check fails if the number of `\n` literals in translation does not match the source.

Невідповідність двокрапки

Підсумок

Джерело та переклад не закінчуються двокрапкою

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.EndColonCheck`

Перевірити ідентифікатор

`end_colon`

Прапорець для нехтування

`ignore-end-colon`

Перевіряє, чи правильно відтворено двокрапки з початкового рядка у перекладі. Також виконується перевірка двокрапок у мовах, де вони не використовуються (китайській і японській).

Дивись також:

[Двокрапка у Вікіпедії](#)

Невідповідність багатокрапок

Підсумок

Джерело і переклад не завершуються трикрапкою

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.EndEllipsisCheck`

Перевірити ідентифікатор

`end_ellipsis`

Прапорець для нехтування

`ignore-end-ellipsis`

Перевіряє, чи відтворено кінцеві три крапки з початкового рядка у перекладі. Виконується перевірка лише для справжнього символу трьох крапок (...), а не трьох послідовних крапок (. . .).

Символ багатокрапки, зазвичай, краще виглядає у друкованому варіанті і краще обробляється системами озвучення тексту.

Дивись також:

[Багатокрапка у Вікіпедії](#)

Невідповідність знаків оклику

Підсумок

Джерело та переклад не закінчуються знаком оклику

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.EndExclamationCheck`

Перевірити ідентифікатор

`end_exclamation`

Прапорець для нехтування

`ignore-end-exclamation`

Перевіряє, чи правильно відтворено знаки оклику з початкового рядка у перекладі. Також виконується перевірка знаків оклику у мовах, де вони не використовуються (китайській, японській, корейській, вірменській, лімбу, м'янмській та нко).

Дивись також:

[Знак оклику у Вікіпедії](#)

Невідповідність кінцевої крапки

Підсумок

Джерело і переклад не завершуються кінцевою крапкою

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.EndStopCheck`

Перевірити ідентифікатор

`end_stop`

Прапорець для нехтування

`ignore-end-stop`

Перевіряє, чи правильно відтворено крапки з початкового рядка у перекладі. Також виконується перевірка крапок у мовах, де вони не використовуються (китайській, японській, деванагарі та урду).

Дивись також:

[Крапка у Вікіпедії](#)

Невідповідність знаків питання

Підсумок

Джерело та переклад не закінчуються знаком питання

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.EndQuestionCheck`

Перевірити ідентифікатор

`end_question`

Прапорець для нехтування

`ignore-end-question`

Перевіряє, чи правильно відтворено знаки питання з початкового рядка у перекладі. Також виконується перевірка знаків питання у мовах, де вони не використовуються (вірменській, арабській, китайській, корейській, японській, ефіопській, ваї або коптській).

Дивись також:

[Знак питання у Вікіпедії](#)

Невідповідність крапки з комою

Підсумок

Джерело і переклад не завершуються крапкою з комою

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.EndSemicolonCheck`

Перевірити ідентифікатор

`end_semicolon`

Прапорець для нехтування

`ignore-end-semicolon`

Перевіряє, чи відтворено кінцеві двокрапки з початкового рядка у перекладі.

Дивись також:

[Крапка з комою у Вікіпедії](#)

Розбіжність розривів рядків

Підсумок

Кількість знаків нових рядків у перекладі не збігається з джерелом

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.NewLineCountCheck`

Перевірити ідентифікатор

`newline-count`

Прапорець для нехтування

`ignore-newline-count`

Usually newlines are important for formatting program output. Check fails if the number of new lines in translation does not match the source.

Бракує множини

Підсумок

Деякі форми множини не перекладено

Область

перекладені рядки

Клас перевірки

`weblate.checks.consistency.PluralsCheck`

Перевірити ідентифікатор

`plurals`

Прапорець для нехтування

`ignore-plurals`

Перевіряє, чи було перекладено усіх форми множини початкового рядка. Специфіку використання кожної з форм множини, яку використано, можна знайти у визначенні рядка.

Якщо деякі з форм множини пропущено, у деяких випадках користувачу буде показано порожній рядок замість форми множини.

Заповнювачі

Нове в версії 3.9.

Підсумок

У перекладі не вистачає деяких заміників

Область

перекладені рядки

Клас перевірки

`weblate.checks.placeholders.PlaceholderCheck`

Перевірити ідентифікатор

`placeholders`

Прапорець для вмикання

`placeholders`

Прапорець для нехтування

`ignore-placeholders`

Змінено в версії 4.3: Ви можете використовувати формальний вираз як заміник.

Змінено в версії 4.13: Із прапорцем `case-insensitive` програма не зважатиме на регістр символів у символах-замінника.

У перекладі пропущено деякі з символів-замінників. Символи-замінники видобуваються з файла перекладу або визначаються вручну за допомогою прапорця `placeholders`. Декілька символів-замінників можна відокремити двокрапкою, рядки з пробілами слід вказувати у лапках:

```
placeholders:$URL$:$TARGET$:"some long text"
```

Якщо ви знаєте синтаксичну конструкцію для заміників, ви можете скористатися регулярним виразом:

```
placeholders:r"%[^\% ]%"
```

Також можна вказати, що не слід враховувати регістр символів у символах-замінниках:

```
placeholders:$URL$:$TARGET$,case-insensitive
```

Дивись також:

Налаштовування поведінки за допомогою прапорців

Щільність знаків

Нове в версії 3.9.

Підсумок

Пропущений нерозривний пробіл перед подвійним знаком пунктуації

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.PunctuationSpacingCheck`

Перевірити ідентифікатор

`punctuation_spacing`

Прапорець для нехтування

`ignore-punctuation-spacing`

Перевіряє, чи є нерозривний пробіл перед символом пунктуації, що складається з двох частин (знаком оклику, знаком питання, крапкою з комою та двокрапкою). Правило щодо додавання пробілу використовується лише у деяких мовах, зокрема французькій та бретонській, де це правило є частиною типографії.

Дивись також:

[Французькі і англійські інтервали у Вікіпедії](#)

Регулярний вираз

Нове в версії 3.9.

Підсумок

Переклад не відповідає регулярному виразу

Область

перекладені рядки

Клас перевірки

`weblate.checks.placeholders.RegexCheck`

Перевірити ідентифікатор

`regex`

Прапорець для вмикання

`regex`

Прапорець для нехтування

`ignore-regex`

Переклад не відповідає формальному виразу. Вираз або видобувається з файла перекладу, або визначається вручну за допомогою прапорця `regex`:

```
regex: ^foo|bar$
```

Однакова множина

Підсумок

Деякі форми множини перекладені однаково

Область

перекладені рядки

Клас перевірки

`weblate.checks.consistency.SamePluralsCheck`

Перевірити ідентифікатор

`same-plurals`

Прапорець для нехтування

`ignore-same-plurals`

Перевірка вважається не пройденою, якщо у перекладі усі форми множини є однаковими. У більшості мов форми множини мають бути різними.

Перехід у новий рядок

Підсумок

Джерело і переклад не починаються з переходу у новий рядок

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.BeginNewlineCheck`

Перевірити ідентифікатор

`begin_newline`

Прапорець для нехтування

`ignore-begin-newline`

Символи розриву рядка, зазвичай, з'являються у початкових рядках з певних причин. Пропускання або додавання цих символів може призвести до проблем із форматуванням при використанні перекладеного тексту.

Дивись також:

Кінцевий знак нового рядка

Початкові пропуски

Підсумок

Джерело і переклад не починаються з однакового числа пропусків

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.BeginSpaceCheck`

Перевірити ідентифікатор

`begin_space`

Прапорець для нехтування

`ignore-begin-space`

Пробіл на початку рядка зазвичай використовується для відступів у інтерфейсі, тому його важливо зберігати.

Кінцевий знак нового рядка

Підсумок

Джерело і переклад не завершуються переходом у новий рядок

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.EndNewlineCheck`

Перевірити ідентифікатор

`end_newline`

Прапорець для нехтування

`ignore-end-newline`

Символи розриву рядка, зазвичай, з'являються у початкових рядках з певних причин. Пропускання або додавання цих символів може призвести до проблем із форматуванням при використанні перекладеного тексту.

Дивись також:

[Перехід у новий рядок](#)

Кінцеві пропуски

Підсумок

Джерело і переклад не завершуються пропуском

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.EndSpaceCheck`

Перевірити ідентифікатор

`end_space`

Прапорець для нехтування

`ignore-end-space`

Перевіряє, чи відтворено кінцеві крапки з початкового рядка у перекладі.

Кінцевий пробіл зазвичай використовується для відокремлення сусідніх елементів, тому його вилучення може призвести до помилок при показі тексту.

Переклад не змінено

Підсумок

Джерело і переклад збігаються

Область

перекладені рядки

Клас перевірки

`weblate.checks.same.SameCheck`

Перевірити ідентифікатор

`same`

Прапорець для нехтування

`ignore-same`

Вважається не пройденою, якщо початковий рядок збігається з перекладом, аж до принаймні одної з форм множини. Деякі рядки, які типові для всіх мов, буде проігноровано, а розмітку вилучено під час перевірки. Так Weblate уникає помилок під час перевірки.

Ця перевірка може допомогти рядки, які помилково не було перекладено.

Типовою поведінкою для цієї перевірки є виключення слів з вбудованого «чорного» списку з перевірок. Слова зі списку часто не перекладають. Перевірка є важливою для усування помилкових спрацювань на коротких рядках, які складаються з одного слова, яке є однаковим у декількох мовах. Цей «чорний» список можна вимкнути за допомогою додавання прапорця `strict-same` до рядка або складника.

Дивись також:

Налаштування складників, Налаштування поведінки за допомогою прапорців

Небезпечний HTML-код

Нове в версії 3.9.

Підсумок

Переклад використовує небезпечну HTML-розмітку

Область

перекладені рядки

Клас перевірки

`weblate.checks.markup.SafeHTMLCheck`

Перевірити ідентифікатор

`safe-html`

Прапорець для вмикання

`safe-html`

Прапорець для нехтування

`ignore-safe-html`

Переклад використовує небезпечну розмітку HTML. Цю перевірку слід увімкнути за допомогою прапорця `safe-html` (див. *Налаштування поведінки за допомогою прапорців*). Також передбачено автоматичний засіб виправлення розмітки.

Підказка: Якщо також використано прапорець `md-text`, буде дозволено також посилання у стилі Markdown.

Дивись також:

Перевірка HTML виконується бібліотекою [Ammonia](#).

URL

Нове в версії 3.5.

Підсумок

Цей переклад не містить URL

Область

перекладені рядки

Клас перевірки

`weblate.checks.markup.URLCheck`

Перевірити ідентифікатор

`url`

Прапорець для вмикання

url

Прапорець для нехтування

ignore-url

У перекладі не міститься адреси. Ця перевірка виконується, лише якщо модуль перекладу позначено як такий, що містить адресу. У цьому випадку у перекладі також має бути коректна адреса.

Розмітка XML**Підсумок**

Теги XML у перекладі не збігаються з тегами XML у джерелі

Область

перекладені рядки

Клас перевірки

weblate.checks.markup.XMLTagsCheck

Перевірити ідентифікатор

xml-tags

Прапорець для нехтування

ignore-xml-tags

Це, зазвичай, означає, що остаточний результат виглядатиме інакше. У більшості цей результат є небажаним у перекладі, але іноді без зміни не обійтися.

Перевіряє, чи відтворено теги XML з початкового рядка у перекладі.

The check is automatically enabled for XML like strings. You might need to add `xml-text` flag in some cases to force turning it on.

Примітка: Цю перевірку вимкнено прапорцем `safe-html`, оскільки чищення коду HTML, яке вона виконує, може призвести до створення розмітки HTML, яка не є коректним кодом XML.

Синтаксис XML

Нове в версії 2.8.

Підсумок

Цей переклад не є правильним XML

Область

перекладені рядки

Клас перевірки

weblate.checks.markup.XMLValidityCheck

Перевірити ідентифікатор

xml-invalid

Прапорець для нехтування

ignore-xml-invalid

Розмітка XML є некоректною.

The check is automatically enabled for XML like strings. You might need to add `xml-text` flag in some cases to force turning it on.

Примітка: Цю перевірку вимкнено прапорцем `safe-html`, оскільки чищення коду HTML, яке вона виконує, може призвести до створення розмітки HTML, яка не є коректним кодом XML.

Простір нульової ширини

Підсумок

Переклад містить зайві символи пропуску нульової ширини

Область

перекладені рядки

Клас перевірки

`weblate.checks.chars.ZeroWidthSpaceCheck`

Перевірити ідентифікатор

`zero-width-space`

Прапорець для нехтування

`ignore-zero-width-space`

Символи пробілів нульової ширини (`<U+200B>`) використовуються для розриву рядків посередині слова (перенесення рядків).

Оскільки ці символи часто включають до перекладів помилково, ця перевірка вмикається, якщо відповідні символи буде знайдено у перекладі. У роботі деяких програм можуть виникати проблеми, якщо буде використано такі символи.

Дивись також:

[Пробіл нульової ширини у Вікіпедії](#)

1.6.4 Перевірки джерела

Перевірки початкових рядків можуть допомогти розробникам удосконалити початкові рядки.

Трикрапка

Підсумок

Рядок використовує три крапки (...) замість символу трикрапки (...)

Область

джерельні рядки

Клас перевірки

`weblate.checks.source.EllipsisCheck`

Перевірити ідентифікатор

`ellipsis`

Прапорець для нехтування

`ignore-ellipsis`

Ця перевірка вважається не пройденою, якщо у рядку використано три крапки (...) замість символу багатокрапки (...).

Використання символу Unicode у більшості випадків є кращим варіантом — символ виглядає краще у друкованому варіанті і краще озвучуються системами озвучення тексту.

Дивись також:

[Багатокрапка у Вікіпедії](#)

Синтаксис ICU MessageFormat

Нове в версії 4.9.

Підсумок

Помилки синтаксису в рядках ICU MessageFormat.

Область

джерельні рядки

Клас перевірки

`weblate.checks.icu.ICUSourceCheck`

Перевірити ідентифікатор

`icu_message_format_syntax`

Прапорець для вмикання

`icu-message-format`

Прапорець для нехтування

`ignore-icu-message-format`

Дивись також:

MessageFormat ICU

Тривалий час не перекладено

Нове в версії 4.1.

Підсумок

Цей рядок вже тривалий час не перекладено

Область

джерельні рядки

Клас перевірки

`weblate.checks.source.LongUntranslatedCheck`

Перевірити ідентифікатор

`long_untranslated`

Прапорець для нехтування

`ignore-long-untranslated`

Якщо рядок не було перекладено доволі давно, це може вказувати на проблему у рядку, яка робить його переклад утрудненим.

Кратні невдалі перевірки

Підсумок

Переклади на декількох мовах мають непройдені перевірки

Область

джерельні рядки

Клас перевірки

`weblate.checks.source.MultipleFailingCheck`

Перевірити ідентифікатор

`multiple_failures`

Прапорець для нехтування

`ignore-multiple-failures`

Багато перекладів цього рядка не проходять перевірок якості. Це, зазвичай, означає, що початковий рядок слід якимось чином удосконалити.

Поширеною причиною того, що рядок не пройшов перевірки, є нестача крапки наприкінці речення або подібні незначні проблеми, які перекладачі намагаються виправити у перекладі, хоча найкращим способом є виправлення початкового рядка.

Декілька змінних без назв

Нове в версії 4.1.

Підсумок

У рядку є декілька змінних без назв — перекладачі не зможуть змінити порядок цих змінних

Область

джерельні рядки

Клас перевірки

`weblate.checks.format.MultipleUnnamedFormatsCheck`

Перевірити ідентифікатор

`unnamed_format`

Прапорець для нехтування

`ignore-unnamed-format`

У рядку є декілька змінних без назв — перекладачі не зможуть змінити порядок цих змінних.

Вам варто використовувати іменовані змінні, щоб перекладачі могли переставляти їх у перекладі.

Немає форм множини

Підсумок

Рядок використовується як множина, але не використовує форми множини

Область

джерельні рядки

Клас перевірки

`weblate.checks.source.OptionalPluralCheck`

Перевірити ідентифікатор

`optional_plural`

Прапорець для нехтування

`ignore-optional-plural`

Рядок використано як рядок із формами множини, але у ньому не використовуються форми множини. Якщо у вашій системі перекладу передбачено підтримку форм множини, вам слід використовувати варіант рядка, який уможлиблює переклад за усіма правилами форм множини різних мов.

Наприклад, із Gettext у Python це може бути:

```
from gettext import ngettext

print(ngettext("Selected %d file", "Selected %d files", files) % files)
```

1.7 Пошук

Нове в версії 3.9.

Для пошуку потрібних вам рядків можна використовувати складні запити із використанням булевих операцій, дужок та фільтрів полів.

Якщо поле не визначено, пошук відбуватиметься у початкових рядках, перекладах та рядках контексту.

The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a light green bar with 'Watched translations 0', 'Suggested translations 0', 'Insights', and a 'Search' button. The main content area is titled 'Search' and contains a search bar with a dropdown menu set to 'All strings'. To the right of the search bar is a 'Sort By' dropdown. Below the search bar is an 'Advanced query builder' section with a 'Source strings' dropdown, a 'Search for...' input, an 'Exact' checkbox, and an 'Add' button. To the right of this is a 'Strings with suggestions' dropdown and another 'Add' button. Below the advanced query builder is a 'String changed after' dropdown with a date input 'mm/dd/yyyy' and an 'Add' button. The 'Query examples' section lists several pre-defined queries with their corresponding filter expressions and an 'Add' button for each:

Query examples	Filter expression	Action
Review strings changed by other users	<code>changed:>=2023-01-29 AND NOT changed_by:testuser</code>	Add
Translated strings	<code>state:>=translated</code>	Add
Strings with comments	<code>has:comment</code>	Add
Strings with any failing checks	<code>has:check</code>	Add
Strings with suggestions from others	<code>has:suggestion AND NOT suggestion_author:testuser</code>	Add
Approved strings with suggestions	<code>state:approved AND has:suggestion</code>	Add
All untranslated strings added the past month	<code>added:>=2023-01-29 AND state:<=needs-editing</code>	Add
Translated strings in a certain language	<code>is:translated AND language:cs</code>	Add

At the bottom of the search area is a large 'Search' button.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.7.1 Простий пошук

Усі фрази, які буде введено до поля для пошуку, буде поділено на слова. Буде показано усі рядки, які містять будь-яке зі слів. Щоб виконати пошук цілої фрази, візьміть «фразу для пошуку» у лапки (можна використувувати одинарні (") та подвійні (») лапки): "це рядок у лапках" або 'інший рядок у лапках'.

1.7.2 Поля

source:ТЕКСТ

Пошук у початковому рядку без урахування регістру символів.

target:ТЕКСТ

Пошук у рядку перекладу без урахування регістру символів.

context:ТЕКСТ

Пошук у рядку контексту без урахування регістру символів.

key:ТЕКСТ

Пошук у рядку ключа без урахування регістру символів.

note:ТЕКСТ

Пошук в описах початкових рядків без урахування регістру символів.

location:ТЕКСТ

Пошук у рядку розташування без урахування регістру символів.

priority:ЧИСЛО

Пріоритетність рядка.

id:ЧИСЛО

Унікальний ідентифікатор рядка.

added:ДАТА_ЧАС

Часова позначка додавання рядка до Weblate.

state:ТЕКСТ

Пошук за станом рядків (approved, translated, needs-editing, empty, read-only), передбачено підтримку *Оператори полів*.

pending:БУЛЕВЕ

Рядок у черзі на заливття до системи керування версіями.

has:ТЕКСТ

Шукати за атрибутами рядка - plural, context, suggestion, comment, check, dismissed-check, translation, variant, screenshot, flags, explanation, glossary, note, label.

is:ТЕКСТ

Пошук перекладів у черзі (pending). Може також шукати за усіма станами рядка (approved, translated, untranslated, needs-editing, read-only).

language:ТЕКСТ

Рядок цільової мови.

component:ТЕКСТ

Пошук ідентифікатора або назви складника незалежно від регістру, див. *Ідентифікатор складника* і *Назва складника*.

project:ТЕКСТ

Ідентифікатор проекту, див. *Ідентифікатор адреси*.

changed_by:ТЕКСТ

Рядок було змінено автором із вказаним ім'ям користувача.

changed:DATA_I_ЧАС

Вміст рядка було змінено у вказаний час, підтримує *Оператори полів*.

change_time:DATETIME

Рядок було змінено у вказаний час, підтримує *Оператори полів*. На відміну від `changed` включає подію, яка не змінює вмісту рядка. Ви також можете застосовувати нетипове фільтрування за діями за допомогою `change_action`.

change_action:ТЕКСТ

Фільтрує за дією зі зміни. Корисний у поєднанні із `change_time`. Приймає назву дії з внесення змін англійською, у лапках із пробілами або літерами нижнього регістру із пробілами, які замінено дефісами. Див. приклади для *Пошук змін*.

check:ТЕКСТ

Рядок не пройшов перевірки, див. *Перевірки і виправлення*, щоб дізнатися більше про перевірку ідентифікаторів.

dismissed_check:ТЕКСТ

Перевірку рядка пропущено, див. *Перевірки і виправлення*, щоб дізнатися більше про перевірку ідентифікаторів.

comment:ТЕКСТ

Шукати у коментарях користувачів.

resolved_comment:ТЕКСТ

Шукати у коментарях, проблеми з яких вирішено.

comment_author:ТЕКСТ

Фільтрувати за автором коментаря.

suggestion:ТЕКСТ

Шукати у пропозиціях.

suggestion_author:ТЕКСТ

Фільтрувати за автором пропозиції.

explanation:ТЕКСТ

Шукати у поясненнях.

label:ТЕКСТ

Пошук у мітках.

screenshot:ТЕКСТ

Пошук у знімках вікон.

1.7.3 Логічні (булеві) оператори

Ви можете поєднувати критерії пошуку за допомогою операторів AND, OR, NOT та дужок для формування складених запитів. Приклад: `state:translated AND (source:hello OR source:bar)`

1.7.4 Оператори полів

Ви можете вказати оператори, діапазони або часткові пошуки для пошуків за датою та числових пошуків:

state:>=translated

Стан `translated` або кращий (`approved`).

changed:2019

Змінено у 2019 році.

changed:[2019-03-01 to 2019-04-01]

Змінено між двома вказаними датами.

1.7.5 Точні оператори

Ви можете створювати запит щодо точної відповідності у інших полях рядка за допомогою оператора `=`. Наприклад, щоб виконати пошук усіх початкових рядків, у яких міститься `hello world`, скористайтеся таким запитом: `source:="hello world"`. Для пошуку виразів з окремих слів можна просто пропустити лапки. Наприклад, щоб знайти усі початкові рядки, у яких міститься `hello`, скористайтеся таким запитом: `source:=hello`.

1.7.6 Пошук змін

Нове в версії 4.4.

Пошук у журналі подій можна виконувати за допомогою операторів `change_action` і `change_time`.

Наприклад, пошук рядків, які було позначено для редагування у 2018 році, можна виконати за допомогою `change_time:2018 AND change_action:marked-for-edit` або `change_time:2018 AND change_action:"Marked for edit"`.

1.7.7 Формальні вирази

У будь-якому фрагменті тексту ви можете вказати формальний вираз як `r"формальний вираз"`.

Наприклад, для пошуку усіх початкових рядків, які містять будь-яку цифру від 2 до 5, скористайтеся таким виразом: `source:r"[2-5]"`.

1.7.8 Попередньо визначені запити

Ви можете вибрати попередньо визначені запити на сторінці пошуку. Це надає вам змогу швидко отримувати доступ до найпоширеніших варіантів запитів щодо пошуку:

The screenshot displays the Weblate web interface for a Django project. The top navigation bar includes links for Dashboard, Projects, Languages, and Checks. The sidebar on the left shows the project structure: WeblateOrg / Django / Czech / Translate. The main area features a custom search filter set to '%(count)s word'. A dropdown menu lists various string states: Untranslated strings (state:empty), Unfinished strings (state:<translated), Translated strings (state:>=translated), Strings marked for edit (state:needs-editing), Strings with suggestions (has:suggestion), Strings with variants (has:variant), Strings with screenshots (has:screenshot), Strings with labels (has:label), Strings with context (has:context), Unfinished strings without suggestions (state:<translated AND NOT has:suggestion), Strings with comments (has:comment), Strings with any failing checks (has:check), Approved strings (state:approved), and Strings waiting for review (state:translated). The right sidebar contains a Glossary section and a String information section with fields for Screenshot context, Explanation, Labels, Flags, Source string location, String age, Source string age, and Translation file. The bottom section includes a 'New comment' form and a 'Save' button.

Powered by Weblate 4.16 · [About Weblate](#) · [Legal](#) · [Contact](#) · [Documentation](#) · [Donate to Weblate](#)

1.7.9 Упорядковування результатів

Передбачено багато варіантів упорядкування, відповідно до ваших потреб:

The screenshot displays the Weblate web interface for managing translations. At the top, there's a navigation bar with links to Dashboard, Projects, Languages, and Checks. Below this, the breadcrumb trail shows 'WeblateOrg / Django / Czech / Translate'. A progress indicator shows 'translated 96%'. The main area features a translation form with fields for 'English' and 'Czech'. The English field contains the text 'The string uses three dots (...) instead of an ellipsis character (...)'. The Czech field is empty. Below the form are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. A dropdown menu is open over the 'Position and priority' button, listing various metadata fields. To the right, a sidebar contains a 'Glossary' section and a 'String information' section with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'. At the bottom, there's a 'New comment' section for users to provide feedback or discuss the translation.

1.8 Процеси перекладу

Використання Weblate є процесом, який наближає ваших користувачів до вас, наближаючи вас до ваших перекладачів. Ви можете самі вирішити, якими з можливостей Weblate ви скористаєтесь.

Нижче наведено неповний список способів налаштувати Weblate. Ви можете побудувати інші робочі процеси на основі типових прикладів з наведеного списку.

1.8.1 Доступ до перекладу

Питання *керування доступом* не займатимуть значного місця у обговоренні процесу перекладу, оскільки будь-який з варіантів керування доступом може бути застосований до будь-якого процесу перекладу. Будь ласка, зверніться до відповідної частини документації, щоб дізнатися про те, як керувати доступом до перекладів.

У наступних розділах під терміном *будь-який користувач* ми матимемо на увазі користувача, який має доступ до перекладу. Це може бути розпізнаний системою користувач, якщо проєкт є відкритим, або користувач, який має права доступу до проєкту рівня *Translate*.

1.8.2 Стани перекладу

Кожен перекладений рядок може перебувати в одному з таких станів:

Не перекладено

Переклад є порожнім — його, залежно від формату, може бути не збережено у файлі перекладу.

Потребує редагування

Переклад потребує редагування, зазвичай, у результаті внесення змін до початкового рядка, неузгодженості або дій перекладача. Переклад зберігається у файлі. Залежно від формату файлу, його може бути позначено таким, що потребує редагування (наприклад, встановлено прапорець *неточного перекладу* у файлі Gettext).

Очікує на перевірку

Переклад виконано, але не рецензовано. Переклад зберігається у файлі як чинний переклад.

Схвалено

Переклад було затверджено у процесі рецензування. Перекладачі вже не можуть змінити його, лише рецензенти. Перекладачі можуть лише подавати пропозиції щодо зміни перекладу.

Цей стан доступний, лише якщо увімкнено рецензування.

Пропозиції

Пропозиції зберігаються лише на Weblate, а не у файлі перекладу.

Стани представлено у файлах перекладу, коли це можливо.

Підказка: Якщо формат файлу, який ви використовуєте, не підтримує стани зберігання, можливо, ви захочете скористатися додатком *Позначити незмінні переклади як такі, що потребують редагування*, щоб позначити незмінні рядки такими, що потребують редагування.

Дивись також:

Можливості типів перекладу, Процеси перекладу

1.8.3 Безпосередній переклад

Цей найпоширеніший варіант для малих команд — безпосередньо перекладати може будь-хто. Це також типовий варіант для Weblate.

- *Будь-який користувач* може редагувати переклади.
- Пропозиції є додатковим способом подання змін, якщо перекладачі не певні щодо точності внесеної зміни.

Параметр	Значення	Примітка
Увімкнути рецензії	вимк.	Налаштовано на рівні проєкту.
Пропонувати переклад	Увімк.	Для користувачів можливість надання пропозицій корисна тим, що уможливорює переклад, якщо користувач не певен щодо його правильності.
Голосування за пропозицію	вимк.	
Автоматичне прийняття пропозиції	0	
Група перекладачів	Users	Або Translate із керуванням доступом на рівні проєкту.
Група рецензентів	Н/Д	Не використовується.

1.8.4 Експертне рецензування

У межах цього робочого процесу будь-хто може додавати пропозиції, які потребуватимуть затвердження від інших учасників команди, перш ніж їх буде прийнято як переклади.

- *Будь-який користувач* може додавати пропозиції.
- *Будь-який користувач* може голосувати за пропозиції.
- Пропозиції стають перекладами, якщо отримують попередньо встановлену кількість голосів.

Параметр	Значення	Примітка
Увімкнути рецензії	вимк.	Налаштовано на рівні проекту.
Пропонувати переклад	Увімк.	
Голосування за пропозицію	вимк.	
Автоматичне прийняття пропозиції	1	Ви можете встановити вище значення, щоб встановити запит щодо більшої кількості незалежних рецензій.
Група перекладачів	<i>Users</i>	Або <i>Translate</i> із <i>керуванням доступом на рівні проекту</i> .
Група рецензентів	Н/Д	Не використовується, рецензування усіх перекладачів.

1.8.5 Спеціалізовані рецензенти

Нове в версії 2.18: Підтримку належного робочого процесу рецензування реалізовано, починаючи з версії Weblate 2.18.

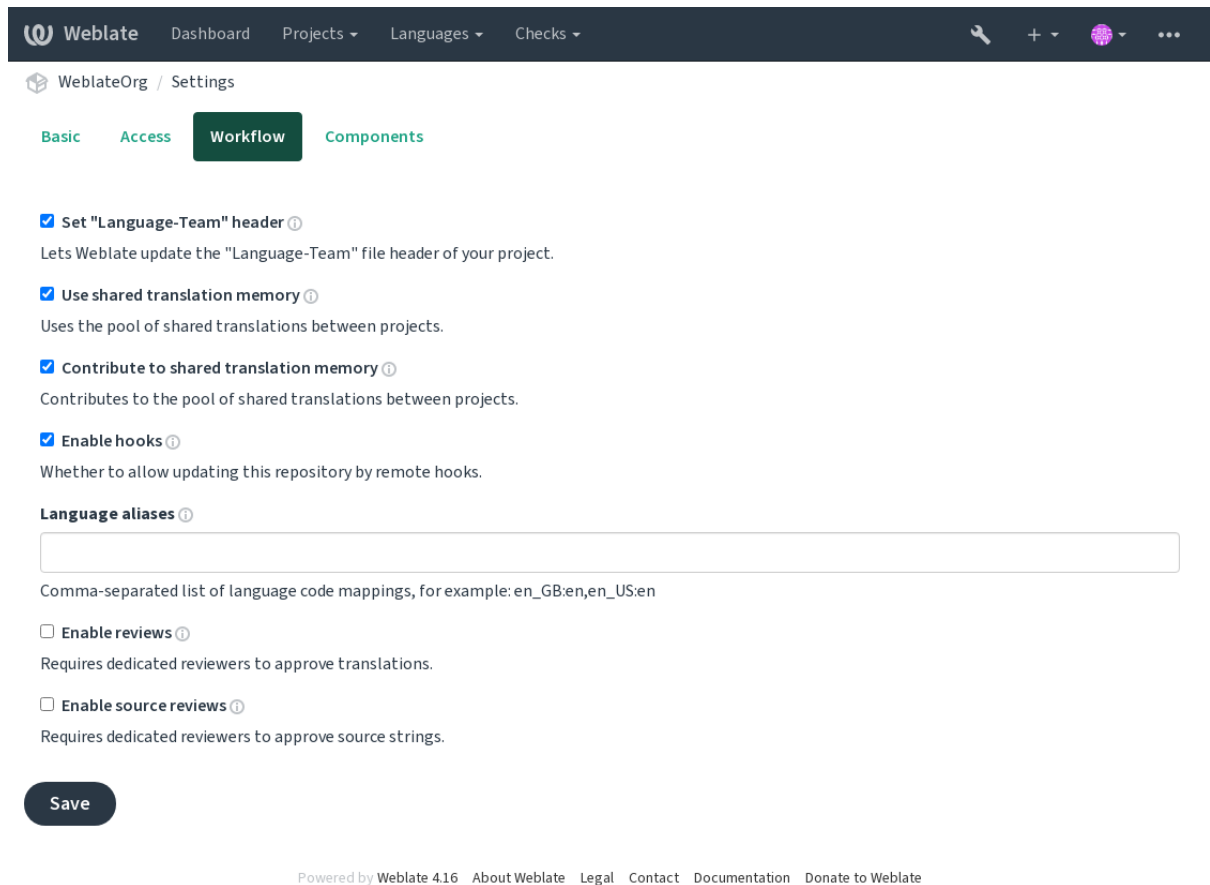
Якщо у проєкті є спеціалізовані рецензенти, ви матимете дві групи користувачів: одна зможе надсилати переклади, а інше рецензувати їх з метою забезпечення коректності та якості.

- *Будь-який користувач* може редагувати незатверджені переклади.
- *Рецензент* може затверджувати рядки та скасовувати затвердження рядків.
- *Рецензент* може редагувати усі переклади (зокрема й затверджені).
- Крім того, пропозиціями можна скористатися для пропонування змін до затверджених рядків.

Параметр	Значення	Примітка
Увімкнути рецензії	Увімк.	Налаштовано на рівні проекту.
Пропонувати переклад	вимк.	Для користувачів можливість надання пропозицій корисна тим, що уможливорює переклад, якщо користувач не певен щодо його правильності.
Голосування за пропозицію	вимк.	
Автоматичне прийняття пропозиції	0	
Група перекладачів	<i>Users</i>	Або <i>Translate</i> із <i>керуванням доступом на рівні проекту</i> .
Група рецензентів	<i>Рецензенти</i>	Або <i>Review</i> із <i>керуванням доступом на рівні проекту</i> .

1.8.6 Вмикання рецензування

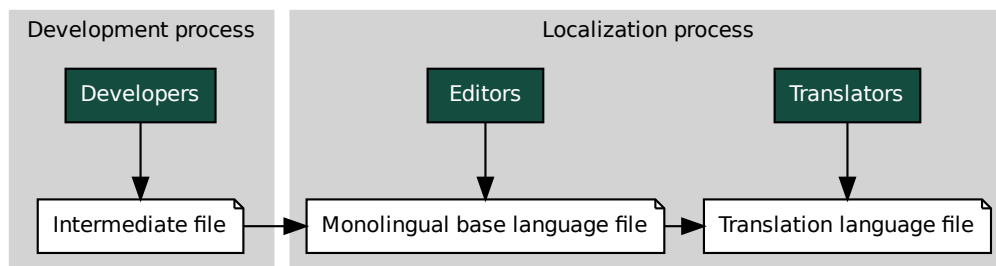
Рецензування можна увімкнути у налаштуваннях проєкту на сторінці *Процес* параметрів проєкту (відкрити її можна за допомогою меню *Керування* → *Параметри*):



1.8.7 Шлюз якості для початкових рядків

У багатьох випадках рядки початковою мовою походять від розробників, оскільки саме вони пишуть код і створюють початкові рядки. Втім, розробники часто не є носіями початкової мови і не забезпечують належної якості початкових рядків. Усунути проблему із початковими рядками може проміжний переклад — це додатковий бар'єр забезпечення якості між розробниками та перекладачами із користувачами.

Якщо встановити *проміжний мовний файл*, цей файл буде використано як джерело рядків, але його можна буде редагувати початковою мовою з метою удосконалення. Щойно рядок буде готовий початковою мовою, він стане доступним для перекладачів — його можна буде перекладати додатковими мовами.

**Дивись також:**

Проміжний мовний файл, Основний файл для одномовного перекладу, Двомовні і одномовні формати

1.8 Рецензування початкових рядків

Якщо увімкнути *рецензування початкових рядків*, до початкових рядків буде застосовано процес рецензування. Після вмикання рецензування користувачі зможуть повідомляти про проблеми із початковими рядками. Характеристики реального процесу залежать від того, який формат використовується, одномовний чи двомовний.

Для одномовних форматів рецензування початкового рядка працює подібно до процедури із *спеціалізованими рецензентами* — щойно буде повідомлено про проблему у початковому рядку, його буде позначено як такий, що *Потребує редагування*.

У двомовних форматах не можна безпосередньо редагувати початкові рядки (їх, зазвичай, видобувають безпосередньо з програмного коду). У цьому випадку до рядка, про який надходять скарги від перекладачів, додається мітка *Початкові рядки потребують рецензування*. Вам слід виконати рецензування таких рядків і або змінити їх у початковому коді, або вилучити мітку.

Дивись також:

Двомовні і одномовні формати, Спеціалізовані рецензенти, labels, Коментарі

1.9 Поширені питання та відповіді на них

1.9.1 Налаштування

Як створити автоматизовану процедуру?

Weblate може обробляти усі завдання, які пов'язано із перекладом, у напіваавтоматичному режимі. Якщо ви надасте Weblate доступ до запису до вашого репозиторію, переклад відбуватиметься без вашого втручання, якщо не станеться якихось конфліктів об'єднання.

1. Налаштуйте ваше репозиторій Git так, щоб Weblate отримував сповіщення при кожній зміні, див. *скрипти сповіщення*, щоб дізнатися більше про те, як це зробити.
2. Установіть адресу запису у розділі складника у Weblate, це надасть змогу Weblate записувати зміни до вашого репозиторію.
3. Увімкніть *запис після внеску* `<component-push_on_commit>` у параметрах вашого *складника* на Weblate — Weblate записуватиме зміни до вашого репозиторію, коли вони траплятимуться на Weblate.

Дивись також:*Безперервна локалізація, Уникання конфліктів об'єднання***Як отримувати доступ до репозиторію за допомогою SSH?**

Будь ласка, ознайомтеся із розділом *Доступ до репозиторіїв*, щоб дізнатися більше про налаштування ключів SSH.

Як виправляти конфлікти об'єднання у перекладах?

Час від часу трапляються конфлікти об'єднання, коли зміни до файла вносяться одночасно на Weblate і у основному репозиторії коду. Зазвичай, цих конфліктів можна уникнути об'єднанням перекладів Weblate з репозиторієм до внесення змін до файлів перекладів (наприклад, до запуску msgmerge). Просто накажіть Weblate вносити усі переклади з черги (зробити це можна за допомогою пункту *Синхронізувати репозиторію* у меню *Управління*) і виконайте об'єднання коду з репозиторієм (якщо не увімкнено автоматичний запис).

Якщо ви вже маєте конфлікт об'єднання, найпростішим способом вирішити проблему є усунення усіх конфліктів локально на вашій робочій станції — просто додайте Weblate як віддалений репозиторій, об'єднайте його із основною гілкою розробки і виправте усі конфлікти. Щойно ви запишете зміни до основного репозиторію, Weblate зможе використовувати об'єднану версію без будь-яких інших додаткових дій.

Примітка: Залежно від конфігурації, доступ до репозиторію Weblate може потребувати проходження розпізнавання. При використанні вбудованого *засобу експортування до git* у Weblate розпізнавання виконуватиметься за вашим іменем користувача та ключем до програмного інтерфейсу.

```
# Commit all pending changes in Weblate, you can do this in the UI as well:
wlc commit
# Lock the translation in Weblate, again this can be done in the UI as well:
wlc lock
# Add Weblate as remote:
git remote add weblate https://hosted.weblate.org/git/project/component/
# You might need to include credentials in some cases:
git remote add weblate https://username:APIKEY@hosted.weblate.org/git/project/
↪component/

# Update weblate remote:
git remote update weblate

# Merge Weblate changes:
git merge weblate/main

# Resolve conflicts:
edit ...
git add ...
...
git commit

# Rebase changes (if Weblate is configured to do rebases)
git rebase origin/main

# Push changes to upstream repository, Weblate will fetch merge from there:
git push

# Open Weblate for translation:
wlc unlock
```

Якщо ви використовуєте у Weblate кілька гілок, ви можете зробити те саме для кожної з них:

```
# Add and update Weblate remotes
git remote add weblate-one https://hosted.weblate.org/git/project/one/
git remote add weblate-second https://hosted.weblate.org/git/project/second/
git remote update weblate-one weblate-second

# Merge QA_4_7 branch:
git checkout QA_4_7
git merge weblate-one/QA_4_7
... # Resolve conflicts
git commit

# Merge main branch:
git checkout main
git merge weblate-second/main
... # Resolve conflicts
git commit

# Push changes to the upstream repository, Weblate will fetch the merge from there:
git push
```

У випадку файлів PO gettext існує спосіб розв'язувати усі конфлікти об'єднання у напівавтоматичному режимі:

Отримайте і збережіть локальний клон репозиторію Git Weblate. Також отримайте другий свіжий локальний клон основного репозиторію Git (тобто, вам знадобляться дві копії репозиторію Git: незмінена та робоча копії):

```
# Add remote:
git remote add weblate /path/to/weblate/snapshot/

# Update Weblate remote:
git remote update weblate

# Merge Weblate changes:
git merge weblate/main

# Resolve conflicts in the PO files:
for PO in `find . -name '*.po'`; do
    msgcat --use-first /path/to/weblate/snapshot/$PO\
                /path/to/upstream/snapshot/$PO -o $PO.merge
    msgmerge --previous --lang=${PO%.po} $PO.merge domain.pot -o $PO
    rm $PO.merge
    git add $PO
done
git commit

# Push changes to the upstream repository, Weblate will fetch merge from there:
git push
```

Дивись також:

Як експортувати репозиторій Git, який використовує Weblate?, Безперервна локалізація, Уникання конфліктів об'єднання, Клієнт Weblate

Як налаштувати одночасний переклад у декількох гілках розробки?

У Weblate передбачено підтримку запису змін у перекладах у межах одних *налаштувань проєкту*. Для кожних *налаштувань складника*, для якого ввімкнено запис (типова поведінка), внесені зміни автоматично поширюються на інші складники. У такий спосіб підтримується синхронізація перекладів, навіть якщо гілки значно різняться, і просте об'єднання файлів перекладів між ними є неможливим.

Щойно зміни з Weblate буде об'єднано, ви можете, вам, ймовірно, слід об'єднати ці гілки (залежно від вашої типової процедури розробки), відкинувши різниці:

```
git merge -s ours origin/maintenance
```

Дивись також:

Підтримання синхронізації перекладів у різних складниках

Як налаштувати переклад багатоплатформових проєктів?

У Weblate передбачено підтримку широкого діапазону форматів файлів (див. *Підтримувані формати файлів*), і найпростішим підходом є використання природного формату для кожної з платформ.

Після додавання всіх файлів перекладу для платформи як складників до одного проєкту (див. *Додавання проєктів і складників перекладу*) ви можете скористатися можливістю поширення перекладів (типово увімкнено, можна вимкнути на рівні *Налаштовування складників*) для перекладу рядків для всіх платформ одночасно.

Дивись також:

Підтримання синхронізації перекладів у різних складниках

Як експортувати репозиторій Git, який використовує Weblate?

У цьому репозиторії немає нічого особливого — воно зберігається у каталозі `DATA_DIR` і має назву `vcs/<проєкт>/<складник>/`. Якщо ви маєте SSH-доступ до відповідного комп'ютера, ви можете використувати репозиторій безпосередньо.

Для анонімного доступу вам варто запустити сервер Git і налаштувати його на обслуговування репозиторію для всіх інших користувачів.

Крім того, ви можете скористатися *Засіб експортування Git* у Weblate для автоматизації процесу.

Якими є варіанти надсилання змін назад до основного репозиторію?

Це значним чином залежить від ваших налаштувань — Weblate є доволі гнучким у цьому сенсі. Ось приклади деяких варіантів робочого процесу, яким можна скористатися у Weblate:

- Weblate автоматично надсилає і об'єднує зміни (див. *Як створити автоматизовану процедуру?*).
- Ви наказуєте Weblate надіслати зміни вручну (це потребує доступ до запису до основного репозиторію).
- Хтось вручну об'єднує зміни з репозиторію git Weblate до основного репозиторію.
- Хтось перезаписує журнал, який створюється Weblate (наприклад, шляхом вилучення внесків із об'єднанням), об'єднує зміни і повідомляє Weblate, що слід відновити початковий стан у основному репозиторії.

Звичайно ж, ви можете використовувати будь-яке поєднання описаних вище варіантів.

Як обмежити доступ Weblate лише перекладами без надання системі доступу до початкового коду?

Ви можете скористатися командою `git submodule` для відокремлення перекладів від початкового коду, лишаючи їх під керуванням системи керування версіями.

1. Створіть репозиторій із вашими файлами перекладів.
2. Додайте його як підлеглий модуль до вашого коду:

```
git submodule add git@example.com:project-translations.git path/to/translations
```

3. Пов'яжіть Weblate із цим репозиторієм — йому більше не знадобиться доступ до репозиторію, яке містить ваш початковий код.
4. Ви можете оновлювати основне репозиторію додаванням перекладів з Weblate за допомогою такої команди:

```
git submodule update --remote path/to/translations
```

Щоб дізнатися більше, будь ласка, ознайомтеся із документацією до `git submodule`.

Як перевірити, чи правильно налаштовано мій Weblate?

До складу Weblate включено набір перевірок налаштувань, підсумки яких можна бачити у адміністративному інтерфейсі — достатньо перейти за посиланням *Звіт щодо швидкодії* у адміністративному інтерфейсі або відкрити адресу, яка завершуватиметься `/manage/performance/`, безпосередньо.

Дивись також:

Спостереження за Weblate, Спостереження за станом Celery

Чому усі внески підписано Weblate <noreply@weblate.org>?

Типове ім'я автора внеску, визначається `DEFAULT_COMMITTER_EMAIL` і `DEFAULT_COMMITTER_NAME`.

Автором же кожного внеску (якщо у відповідній системі керування версіями передбачено підтримку авторів внесків) буде належним чином записано — ним буде користувач, яким виконано переклад.

Для внесків із невідомим авторством (наприклад, анонімних пропозицій або результатів комп'ютерного перекладу) авторство належатиме анонімному користувачеві (див. `ANONYMOUS_USER_NAME`). Ви можете змінити ім'я та адресу електронної пошти за допомогою інтерфейсу керування.

Дивись також:

Налаштовування складників

Як пересунути файли у сховищі без втрати журналу у Weblate?

Щоб зберегти журнал, коментарі або знімки вікон, які пов'язано із рядками, після зміни розташування файлів, вам слід забезпечити неможливість вилучення рядків на Weblate. Такі вилучення можуть статися, якщо сховище Weblate оновлено, але налаштування складника все ще вказує на старі файли. У таких ситуаціях Weblate припускає, що слід вилучити всі переклади.

Рішення полягає у виконанні синхронної дії у Weblate:

1. Заблокуйте відповідний складник у Weblate.
2. Внесіть усі зміни з черги і об'єднайте їх зі змінами в основному сховищі.
3. Вимкніть скрипти отримання *проекти*; це запобігатиме негайному виявленню змін у сховищі з боку Weblate.

4. Внесіть усі потрібні вам зміни до сховища (наприклад, за допомогою `git mv`), запишіть їх до основного сховища.
5. Внесіть зміни до *складника* так, щоб все збігалось із новою конфігурацією. Після зміни налаштувань Weblate отримає оновлене сховище і зареєструє змінені місця, зберігши наявні рядки.
6. Розблокуйте складник і повторно увімкніть скрипти у налаштуваннях проекту.

1.9.2 Користування

Як рецензувати переклади інших користувачів?

- У Weblate передбачено підтримку кількох робочих процесів з рецензування, див. *Процеси перекладу*.
- Ви можете підписатися на усі зміни у розділі *Сповіщення*, а потім перевіряти внески інших учасників перекладу, коли сповіщення надходять вам електронною поштою.
- Передбачено особливий інструмент, панель його розташовано у нижній частині панелі перекладу. На цій панелі ви можете вибрати навігацію перекладами, які виконано іншими користувачами, починаючи з вказаної дати.

Дивись також:

Процеси перекладу

Як надавати відгуки щодо початкового рядка?

На контекстних вкладках під перекладом ви можете скористатися вкладкою *Коментарі*, щоб надати відгук щодо початкового рядка або обговорити його із іншими перекладачами.

Дивись також:

report-source, *Коментарі*

Як скористатися наявними перекладами під час перекладу?

- Усі переклади у межах Weblate можна використовувати завдяки спільній пам'яті перекладів.
- Ви можете імпортувати до Weblate наявні файли пам'яті перекладів.
- Скористайтесь функціональними можливостями з імпортування для завантаження копії перекладів, пропозицій або перекладів, які потребують рецензування. Це оптимальний підхід для одноразового перекладу на основі копії або подібної бази даних перекладів.
- Ви можете налаштувати *tmserver* із усіма базами даних перекладів, які у вас є, і надати змогу Weblate користуватися ними. Це корисно, якщо ви хочете скористатися ними декілька разів під час перекладу.
- Іншим варіантом є переклад усіх пов'язаних перекладів в одному екземплярі Weblate, що забезпечить автоматичне поширення перекладів з одного складника до інших.

Дивись також:

Налаштовування автоматичних пропозицій, Автоматичні пропозиції, Пам'ять перекладів

Чи оновлює Weblate файли перекладів між перекладами?

Weblate намагається звести до мінімуму зміни у файлах перекладів. Для деяких форматів файлів це, на жаль, може призвести до переформатування файла. Якщо ви хочете зберегти бажане для вас форматування, будь ласка, скористайтеся для цього скриптом попередньої обробки даних.

Дивись також:

updating-target-files

Звідки беруться визначення мов і як можна додати власне визначення?

Базовий набір визначень мов включено у пакунки Weblate і Translate-toolkit. У цьому наборі є дані понад 150 мов разом із даними щодо форм множини та напрямку запису тексту.

Ви можете визначати власні мови за допомогою адміністративного інтерфейсу — вам достатньо вказати дані щодо цієї мови.

Дивись також:

Визначення мов

Чи може Weblate підсвічувати зміни у неточно перекладеному рядку?

У Weblate передбачено підтримку цієї можливості, але програмі потрібні дані для показу відмінностей.

Для файлів PO Gettext вам слід передати параметр `--previous` до команди **msgmerge** при оновленні файлів PO. Приклад:

```
msgmerge --previous -U po/cs.po po/phpmyadmin.pot
```

Для одномовних перекладів Weblate може знаходити попередній рядок за ідентифікатором, тому програма показує різниці автоматично.

Чому Weblate показує застарілі рядки перекладу, хоча шаблон перекладу було оновлено?

Weblate не виконує ніяких дій із файлами перекладу, окрім надання перекладачам можливості перекладати ці файли. Тому файли перекладів не буде оновлено, якщо буде оновлено шаблон або початковий код, на основі якого формується шаблон. Вам просто слід зробити це вручну і записати зміни до репозиторію, Weblate підхопить зміни автоматично.

Примітка: Зазвичай, варто об'єднувати зміни, які було внесено у Weblate до оновлення файлів перекладу, інакше, зазвичай, виникатимуть конфлікти, які доведеться усувати.

Наприклад, із файлами PO gettext ви можете оновити файли перекладів за допомогою інструмента **msgmerge**:

```
msgmerge -U locale/cs/LC_MESSAGES/django.mo locale/django.pot
```

Якщо ви хочете, щоб оновлення відбувалося автоматично, ви можете встановити додаток *Оновити файли po, порівнявши із pot (msgmerge)*.

Дивись також:

updating-target-files

How to handle renaming translation files?

When renaming files in the repository, it can happen that Weblate sees this as removal and adding of the files. This can lead to losing strings history, comments and suggestions.

To avoid that, perform renaming in following steps:

1. Lock the translation component in manage-vcs.
2. Commit pending changes in manage-vcs.
3. Merge Weblate changes to the upstream repository.
4. Disable receiving updates via hooks using *Увімкнути обробники*.
5. Perform the renaming of the files in the repository.
6. Update the component configuration to match new file names.
7. Enable update hooks and unlock the component.

1.9.3 Вирішення проблем

Іноді запити не виконуються із повідомленням про помилку «відкрито забагато файлів»

Таке трапляється, якщо ваш репозиторій Git розростається й у вас стає забагато файлів. Стискання репозиторію Git має усунути проблему.

Найпростішим способом зробити це є така команда:

```
# Go to DATA_DIR directory
cd data/vcs
# Compress all Git repositories
for d in */* ; do
    pushd $d
    git gc
    popd
done
```

Дивись також:

DATA_DIR

При спробі доступу до сайту браузер показує повідомлення про помилку «Bad Request (400)»

Найімовірнішою причиною є помилкове значення змінної *ALLOWED_HOSTS*. У цій змінній має містити список усіх назв вузлів, яким ви хочете дозволити доступ до вашого Weblate. Приклад:

```
ALLOWED_HOSTS = ["weblate.example.com", "weblate", "localhost"]
```

Дивись також:

Налаштовування дозволених вузлів

Що означає повідомлення «Знайдено кілька файлів для однієї мови (en)»?

Таке типowo трапляється, якщо у вас є файл перекладу початковою мовою. Weblate стежить за початковими рядками і резервує для цього каталог початкової мови. Додатковий файл початковою мовою не обробляється.

- Якщо переклад початковою мовою є бажаним, будь ласка, змініть початкову мову `<component-source_language>` у налаштуваннях складника. Вам варто скористатися ``English (Developer)` як початковою мовою або використати *Шлюз якості для початкових рядків*.
- Якщо файл перекладу для початкової мови є непотрібним, будь ласка вилучіть його з репозиторію.
- Якщо файл перекладу для початкової мови є бажаним, але Weblate має його ігнорувати, будь ласка, скоригуйте *Фільтр мов*, щоб виключити його.

Підказка: Ви можете отримати подібне повідомлення про помилку і для інших мов. У такому випадку найімовірнішою причиною є те, що з однією мовою у Weblate пов'язано кілька файлів.

Це може бути наслідком того, що паралельно використано застарілі і нові коди мов (`ja` і `jp` для японської) або паралельно вжито код мови із кодом країни і загальний код (`fr` і `fr_FR`). Докладніше про це у розділі *Обробка кодів мов*.

1.9.4 Можливості

Чи передбачено у Weblate підтримку інших систем керування версіями, окрім Git і Mercurial?

У поточній версії Weblate не передбачено вбудованої підтримки нічого, окрім *Git* (із розширеною підтримкою *Заняти щодо злиття GitHub, Gerrit і Subversion*) та `ref:vcs-mercurial`. Втім, можливе написання модулів для інших систем керування версіями.

Крім того, ви можете скористатися *Віддалені допоміжні засоби Git* у *Git* для доступу до інших систем керування версіями.

Крім того, у Weblate передбачено підтримку дій без системи керування версіями, див. *Локальні файли*.

Примітка: Для вбудованої підтримки інших систем керування версіями Weblate потрібне використання розподіленої системи керування версіями. Ймовірно, Weblate може працювати із будь-чим, окрім *Git* та *Mercurial*, але хтось має реалізувати підтримку відповідної системи керування версіями.

Дивись також:

Налаштування інтеграції із керуванням версіями

Як Weblate зберігає авторство перекладів?

Усі зміни, які внесено на Weblate, надсилаються до системи керування версіями від імені перекладачів. У такий спосіб забезпечується належне авторство кожної окремої зміни. Ви можете стежити за змінами за допомогою стандартних інструментів вашої системи керування версіями точно так само, як ви це робите з кодом.

Крім того, якщо таку підтримку передбачено у форматі файлів перекладу, буде оновлено заголовки файла — до них буде включено ім'я перекладача.

Дивись також:

`list_translators`, `../devel/reporting`

Чому Weblate примусово показує усі файли PO у одній ієрархії?

Weblate було розроблено із припущенням, що кожен файл PO представлено у системі окремим складником. Це зручно для перекладачів — вони знають, що саме перекладають.

Змінено в версії 4.2: Перекладачі можуть перекласти всі складники проєкту певною мовою в цілому.

Чому Weblate використовує дивні коди мов, зокрема `sr_Latn` та `zh_Hant`?

Ці коди мови визначено [RFC 5646](#) для удосконалення індикації різних мов. Раніше для цього помилково використовувалися модифікатори (для варіантів `@latin`) або коди країн (для китайської).

Weblate може працювати із застарілими кодами мов і пов'язувати їх із кодами поточної версії. Наприклад, `sr@latin` буде оброблено як `sr_Latn`, а `zh@CN` як `zh_Hans`.

Примітка: Типово, Weblate використовує коди мов у стилі POSIX, із підкреслюваннями. Див. [Визначення мов](#), щоб дізнатися більше.

Дивись також:

[Визначення мов](#), [Стиль коду мови](#), [Додавання нових перекладів](#)

1.10 Підтримувані формати файлів

У Weblate передбачено підтримку більшості форматів перекладів, які може обробляти [translate-toolkit](#). Втім, у кожного формату є свої особливості, тому можуть виникати певні проблеми з форматами, які не було належним чином тестовано.

Дивись також:

[Translation Related File Formats](#)

Примітка: При виборі формату файлів для вашого застосунку слід надавати перевагу типовим форматам на платформі або у наборі інструментів, яким ви користуєтесь. Якщо ви це зробите, перекладачі зможуть користуватися знайомими їм інструментами, що збільшить ймовірність того, що вони візьмуть участь у вашому проєкті.

1.10.1 Двомовні і одномовні формати

Передбачено підтримку як одномовних, так і двомовних форматів. У двомовних форматах зберігаються повідомлення двома мовами в одному файлі — початковий рядок і переклад (типовими прикладами є [GNU Gettext](#), [XLIFF](#) або [Рядки Apple iOS](#)). З іншого боку, в одномовних форматах передбачено ідентифікацію рядка за ідентифікатором, а кожен мовний файл містить лише прив'язку цих ідентифікаторів до перекладів кожного з мов (типовим прикладом є [Рядкові ресурси Android](#)). Деякі формати файлів використовуються у обох варіантах. Докладніший опис можна знайти нижче.

Для правильного використання одномовних файлів Weblate потрібен доступ до файла, у якому міститься повний список рядків, які слід перекласти із початковими даними — цей файл у Weblate має назву [Основний файл для одномовного перекладу](#) (назва може бути дещо іншою у вашій парадигмі).

Крім того, робочий процес може бути розширено використанням [проміжного мовного файла](#), до якого буде включено рядки, які надано розробниками, але які не буде використано без обробки у остаточних рядках перекладу.

1.10.2 Автоматичне виявлення

Weblate може автоматично визначати декілька поширених форматів файлів, але це визначення може значно вплинути на швидкодію і обмежить можливості списком можливостей, які є специфічними для заданого формату файлів (наприклад, автоматичним додаванням нових перекладів).

1.10.3 Можливості типів перекладу

Table 1: Можливості всіх підтримуваних форматів

Формат	Кількість мов ^{с. 80, 1}	Форми множини ^{с. 80, 2}	Опис ^{с. 80, 3}	Кон-текст ^{с. 80, 4}	Розташування ^{с. 80, 5}	Прапорці ^{с. 80, 8}	Додаткові стани ^{с. 80, 6}
<i>GNU Gettext</i>	двомовний	так	так	так	так	так ⁹	потребує редагування
<i>Одномовний gettext</i>	одномовний	так	так	так	так	так ⁹	потребує редагування
<i>XLIFF</i>	обидва	так	так	так	так	так ¹⁰	потребує редагування, затверджено
<i>Властивості Java</i>	обидва	ні	так	ні	ні	ні	
<i>Мовні файли m18n</i>	одномовний	ні	так	ні	ні	ні	
<i>Властивості GWT</i>	одномовний	так	так	ні	ні	ні	
<i>Переклади Joomla</i>	одномовний	ні	так	ні	так	ні	
<i>Qt Linguist .ts</i>	обидва	так	так	ні	так	так ^{с. 80, 10}	потребує редагування
<i>Рядкові ресурси Android</i>	одномовний	так	так ⁷	ні	ні	так ^{с. 80, 10}	
<i>Рядки Apple iOS</i>	обидва	ні	так	ні	ні	ні	
<i>Рядки PHP</i>	одномовний	ні ¹¹	так	ні	ні	ні	
<i>файли JSON</i>	одномовний	ні	ні	ні	ні	ні	
<i>Файли JSON i18next</i>	одномовний	так	ні	ні	ні	ні	
<i>Файли JSON go-i18n</i>	одномовний	так	так	ні	ні	ні	

continues on next page

Table 1 – continued from previous page

Формат	Кількість мов ^{с. 80, 1}	Форми множини ²	Описи ³	Контекст ⁴	Розташування ⁵	Прапорці ⁸	Додаткові стани ⁶
<i>Файли JSON gotext</i>	одномовний	так	так	ні	так	ні	
<i>Файл ARB</i>	одномовний	так	так	ні	ні	ні	
<i>JSON WebExtensio.</i>	одномовний	так	так	ні	ні	ні	
<i>Файли ресурсів .XML</i>	одномовний	ні	так	ні	ні	так ^{с. 80, 10}	
<i>Файли ResourceDict</i>	одномовний	ні	ні	ні	ні	так ^{с. 80, 10}	
<i>Файли CSV</i>	обидва	ні	так	так	так	ні	потребує редагування
<i>Файли YAML</i>	одномовний	ні	так	ні	ні	ні	
<i>Файли YAML Ruby</i>	одномовний	так	так	ні	ні	ні	
<i>Файли DTD</i>	одномовний	ні	ні	ні	ні	ні	
<i>Прості файли XML</i>	одномовний	ні	ні	ні	ні	так ^{с. 80, 10}	
<i>Файли RC Windows</i>	одномовний	ні	так	ні	ні	ні	
<i>Excel Open XML</i>	одномовний	ні	так	так	так	ні	потребує редагування
<i>Файли мета-даних крамниці застосунків</i>	одномовний	ні	ні	ні	ні	ні	
<i>Файли субтитрів</i>	одномовний	ні	ні	ні	так	ні	
<i>Файли HTML</i>	одномовний	ні	ні	ні	ні	ні	
<i>Формат OpenDocument</i>	одномовний	ні	ні	ні	ні	ні	
<i>Формат IDML</i>	одномовний	ні	ні	ні	ні	ні	
<i>Переклади INI</i>	одномовний	ні	ні	ні	ні	ні	
<i>Переклади INI Inno Setup</i>	одномовний	ні	ні	ні	ні	ні	
<i>Формат TermBase eXchange</i>	двомовний	ні	так	ні	ні	так ^{с. 80, 10}	
<i>Текстові файли</i>	одномовний	ні	ні	ні	ні	ні	

continues on next page

Table 1 – continued from previous page

Формат	Кількість мов ^{с, 80, 1}	Форми множини ²	Описи ³	Контекст ⁴	Розташування ⁵	Прапорці ⁸	Додаткові стани ⁶
<i>Формат Stringsdict</i>	одномовний	так	так	ні	ні	ні	
<i>Формат Fluent</i>	одномовний	ні ¹²	так	ні	ні	ні	

Рядки лише для читання

Нове в версії 3.10.

До файла буде включено призначені лише для читання рядки з файлів перекладу, але ці рядки не можна бути редагувати у Weblate. Придатні лише до читання рядки передбачено у деяких форматах (*XLIFF* і *Рядкові ресурси Android*), але її можна емулювати у інших форматах додаванням прапорця `read-only`, див. *Налаштовування поведінки за допомогою прапорців*.

1.10.4 GNU Gettext

Найвикористовуваніший формат для перекладу вільного ПЗ.

Підтримку контекстних даних, які зберігаються у файлі, реалізовано коригуванням його заголовків та прив'язкою до відповідних файлів початкового коду.

Типовий двомовний файл PO gettext виглядає ось так:

```
#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "Monday"
msgstr "Pondělí"

#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "Tuesday"
msgstr "Úterý"

#: weblate/accounts/avatar.py:163
msgctxt "No known user"
msgid "None"
msgstr "Žádný"
```

¹ Див. *Двомовні і одномовні формати*

² Форми множини потрібні для належної локалізації рядків із змінними-лічильниками.

³ Описами початкових рядків можна скористатися для передавання додаткових відомостей щодо рядка, який слід перекласти.

⁴ Контекст використовується для розрізнення ідентичних рядків, які використовуються у різних парадигмах (наприклад, *Sin* може бути використано як скорочення дня тижня «Sunday» або як назва найближчої до нас зірки).

⁵ Дані щодо розташування рядка у початковому коді можуть допомогти досвідченим перекладачам визначити, у який спосіб використовується рядок.

⁸ Див. *Налаштовування поведінки за допомогою прапорців*

⁶ Форматом передбачено підтримку додаткових станів рядка, окрім станів «Неперекладений» та «Перекладено».

⁹ Коментарі типу gettext використовуються як прапорці.

¹⁰ Прапорці видобуваються з нестандартного атрибута `weblate-flags` для всіх заснованих на XML форматів. Крім того, передбачено підтримку `max-length:N` шляхом обробки атрибута `maxwidth` у формі, яку визначено у стандарті XLIFF, див. *Визначення прапорців перекладу*.

⁷ Коментар XML, який додано перед елементом `<string>`, буде оброблено як опис початкового рядка.

¹¹ Підтримку форм множини передбачено лише для Laravel, який використовує синтаксис рядків для визначення цих форм, див. *Локалізація у Laravel*.

¹² Форми множини обробляються у синтаксисі рядків і не показуються як форми множини на Weblate.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	po/* .po
Основний файл для одномовного перекладу	<i>Порожньо</i>
Шаблон для нових перекладів	po/messages .pot
Формат файлу	<i>файл PO Gettext</i>

Дивись також:

devel/gettext, devel/sphinx, Gettext у Вікіпедії, PO Files, *Оновити змінну ALL_LINGUAS у файлі «configure»*, *Налаштувати виведення gettext*, *Оновити файл LINGUAS*, *Генерувати то-файли*, *Оновити файли po, порівнявши із pot (msgmerge)*

Одномовний gettext

У деяких проєктах розробники використовують gettext як одномовний формат — у початковому коді є лише ідентифікатори, а рядок має бути перекладено усіма мовами, зокрема і англійською. Підтримку таких форматів передбачено, але цей формат слід вибрати явним чином при імпортуванні складників до Weblate.

Типовий одномовний файл PO gettext виглядає ось так:

```
#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "day-monday"
msgstr "Pondělí"

#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "day-tuesday"
msgstr "Úterý"

#: weblate/accounts/avatar.py:163
msgid "none-user"
msgstr "Žádný"
```

А базовий мовний файл ось такий:

```
#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "day-monday"
msgstr "Monday"

#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "day-tuesday"
msgstr "Tuesday"

#: weblate/accounts/avatar.py:163
msgid "none-user"
msgstr "None"
```

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	po/* .po
Основний файл для одномовного перекладу	po/en .po
Шаблон для нових перекладів	po/messages .pot
Формат файлу	<i>файл PO Gettext (одномовний)</i>

1.10.5 XLIFF

Формат на основі XML створено для стандартизації файлів перекладу, але, врешті, він став лише одним із багатьох стандартів у цій області.

XML Localization Interchange File Format (XLIFF) зазвичай використовується як двомовний, але у Weblate передбачено підтримку і одномовних варіантів.

У Weblate передбачено підтримку декількох варіантів XLIFF:

файл перекладу XLIFF

Простий файл XLIFF, де вміст елементів зберігається у формі простого тексту (усі елементи XML екрановано).

XLIFF із підтримкою заміників

Стандартний XLIFF із підтримкою заміників та інших елементів XML.

XLIFF із розширеннями gettext

XLIFF, удосконалений за [XLIFF 1.2 Representation Guide for Gettext PO](#) з метою забезпечення підтримки форм множини.

Дивись також:

Специфікація *XML Localization Interchange File Format (XLIFF)*, [Настанови щодо представлення XLIFF 1.2 для Gettext PO](#)

Стани перекладу

Змінено в версії 3.3: Weblate ігнорував атрибут стану до версії 3.3.

Дані атрибута `state` у файлі частково обробляються і пов'язуються зі станом «Потребує редагування» у Weblate (для позначення рядка як такого, що потребує редагування, використовується такі стани, якщо вони є у записі перекладу: `new`, `needs-translation`, `needs-adaptation`, `needs-l10n`). Якщо атрибут `state` не вказано, рядок вважатиметься перекладеним, якщо існує елемент `<target>`.

Якщо рядок перекладу має атрибут `approved="yes"`, його також буде імпортовано до Weblate як «Підтверджений». Усі інші переклади буде імпортовано як «Очікують на рецензування» (що відповідає специфікації XLIFF).

Під час збереження Weblate не додає ці атрибути, якщо вони не є потрібними:

- Атрибут `state` додається, лише якщо рядок позначено як такий, який потребує редагування.
- Атрибут `approved` буде додано, лише якщо рядок було рецензовано.
- У інших випадках атрибути не додаються, але їх буде оновлено, якщо вони були.

Це означає, що при використанні формату XLIFF, наполегливо рекомендуємо увімкнути процес рецензування Weblate, щоб переглядати і змінювати стан підтвердження рядків.

Так само, під час імпортування таких файлів (у вивантаженій формі) ви можете вибрати варіант *Імпортувати як перекладено* у розділі *Обробка рядків, які потребують рецензування*.

Дивись також:

[Спеціалізовані рецензенти](#)

Пробіли і символи перенесення рядків у XLIFF

Загалом, типи або кількість пробілів у форматах XML не враховуються. Якщо ви хочете зберегти пробіли, вам слід додати прапорець `xml:space="preserve"` до рядка.

Приклад:

```
<trans-unit id="10" approved="yes">
  <source xml:space="preserve">hello</source>
  <target xml:space="preserve">Hello, world!
</target>
</trans-unit>
```

Визначення прапорців перекладу

Ви можете вказати додаткові прапорці перекладу (див. *Налаштовування поведінки за допомогою прапорців*) за допомогою атрибута `weblate-flags`. Крім того, Weblate розуміє атрибути `maxwidth` і `font` зі специфікації XLIFF:

```
<trans-unit id="10" maxwidth="100" size-unit="pixel" font="ubuntu;22:bold">
  <source>Hello %s</source>
</trans-unit>
<trans-unit id="20" maxwidth="100" size-unit="char" weblate-flags="c-format">
  <source>Hello %s</source>
</trans-unit>
```

Атрибут `font` обробляється для отримання сімейства шрифтів, розміру та ваги. У наведеному вище прикладі показано усе це, хоча потрібним є лише сімейство шрифтів. Усі пробіли у записі сімейства шрифтів буде перетворено на підкреслювання, отже `Source Sans Pro` буде перетворено `Source_Sans_Pro`. Будь ласка, зверніть увагу на це при іменуванні групи шрифтів (див. *Керування шрифтами*).

Ключі рядків

Weblate визначає модулі у файлі XLIFF за допомогою атрибута `resname`, якщо його вказано у файлі та використовує резервний атрибут `id` (разом з міткою `file`, якщо її вказано).

Вмістом `resname` має бути зручний для читання ідентифікатор модуля, який варто показати у Weblate замість `id`. Вміст `resname` не повинен повторюватися у межах файла XLIFF. Це вимога Weblate, яка не є частиною стандарту XLIFF. Стандарт не накладає вимоги унікальності на цей атрибут.

Типовий <i>Налаштовування складників</i> Weblate для двомовних XLIFF	
Маска файлів	<code>localizations/*.xliff</code>
Основний файл для одномовного перекладу	<i>Порожньо</i>
Шаблон для нових перекладів	<code>localizations/en-US.xliff</code>
Формат файлу	<i>файл перекладу XLIFF</i>

Типовий <i>Налаштовування складників</i> Weblate для одномовних XLIFF	
Маска файлів	<code>localizations/*.xliff</code>
Основний файл для одномовного перекладу	<code>localizations/en-US.xliff</code>
Шаблон для нових перекладів	<code>localizations/en-US.xliff</code>
Формат файлу	<i>файл перекладу XLIFF</i>

Дивись також:

[XLIFF у Вікіпедії](#), [XLIFF](#), [Атрибут font у XLIFF 1.2](#), [Атрибут maxwidth у XLIFF 1.2](#)

1.10.6 Властивості Java

Власний формат Java для перекладів.

Властивості Java, зазвичай, використовуються як одномовні переклади.

У Weblate передбачено підтримку варіантів ISO-8859-1, UTF-8 і UTF-16 цього формату. У всіх з цих форматів передбачено збереження усіх символів Unicode, лише у різних кодуваннях. В ISO-8859-1 використовуються екрановані послідовності Unicode (наприклад `zkou\u0161ka`). У інших варіантах символи кодуються безпосередньо, у UTF-8 або UTF-16.

Примітка: Завантаження екранованих послідовностей працює також і у режимі UTF-8, тому, будь ласка, будьте обережними із вибором належного кодування, яке відповідатиме потребам вашого застосунку.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	<code>src/app/Bundle_*.properties</code>
Основний файл для одномовного перекладу	<code>src/app/Bundle.properties</code>
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>Властивості Java (ISO-8859-1)</i>

Дивись також:

Властивості Java у Вікіпедії, Mozilla and Java properties files, *Мовні файли mi18n*, *updating-target-files*, *Форматувати файл властивостей Java*, *Очищення файлів перекладу*

1.10.7 Мовні файли mi18n

Нове в версії 4.7.

Формат файлів, який використовують для локалізації коду JavaScript за допомогою *mi18n*. Синтаксично збігається із *Властивості Java*.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	<code>*.lang</code>
Основний файл для одномовного перекладу	<code>en-US.lang</code>
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>мовний файл mi18n</i>

Дивись також:

mi18n, Mozilla and Java properties files, *Властивості Java*, *updating-target-files*, *Форматувати файл властивостей Java*, *Очищення файлів перекладу*

1.10.8 Властивості GWT

Власний формат GWT для перекладів.

Властивості GWT, зазвичай, використовуються як одномовні переклади.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	<code>src/app/Bundle_*.properties</code>
Основний файл для одномовного перекладу	<code>src/app/Bundle.properties</code>
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>Властивості GWT</i>

Дивись також:

Підручник з локалізації GWT, Настанови з перекладу GWT, Mozilla and Java properties files, updating-target-files, *Форматувати файл властивостей Java*, *Очищення файлів перекладу*

1.10.9 Переклади INI

Нове в версії 4.1.

Формат файлів INI для перекладу.

Переклади INI, зазвичай, використовуються як одномовні переклади.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	language/*.ini
Основний файл для одномовного перекладу	language/en.ini
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл INI</i>

Примітка: Weblate видобуває ключові слова лише з розділів у файлі INI. Якщо у файлі INI немає розділів, вам варто скористатися форматом *Переклади Joomla* або *Властивості Java*.

Дивись також:

INI Files, *Властивості Java*, *Переклади Joomla*, *Переклади INI Inno Setup*

1.10.10 Переклади INI Inno Setup

Нове в версії 4.1.

Формат файлів INI Inno Setup для перекладу.

Переклади INI Inno Setup, зазвичай, використовуються як одномовні переклади.

Примітка: Єдиною помітною відмінністю від *Переклади INI* є підтримка замінників %n і %t для розриву рядка і табуляції.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	language/*.isl
Основний файл для одномовного перекладу	language/en.isl
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл INI Inno Setup</i>

Примітка: У поточній версії передбачено підтримку лише файлів Unicode (.isl), підтримки варіанта ANSI (.isl) у поточній версії не передбачено.

Дивись також:

INI Files, *Переклади Joomla*, *Переклади INI*

1.10.11 Переклади Joomla

Нове в версії 2.12.

Власний формат Joomla для перекладів.

Переклади Joomla, зазвичай, використовуються як одномовні переклади.

Типовий <i>Налаштування складників</i> Weblate	
Маска файлів	language/*/com_foobar.ini
Основний файл для одномовного перекладу	language/en-GB/com_foobar.ini
Шаблон для нових перекладів	Порожньо
Формат файлу	мовний файл Joomla

Дивись також:

Mozilla and Java properties files, *Переклади INI*, *Переклади INI Inno Setup*

1.10.12 Qt Linguist .ts

Формат перекладів, який використовується у перекладах застосунків на основі Qt.

Файли Qt Linguist використовуються і як двомовні, і як одномовні переклади.

Типовий <i>Налаштування складників</i> Weblate при використанні як двомовного файла	
Маска файлів	i18n/app.*.ts
Основний файл для одномовного перекладу	Порожньо
Шаблон для нових перекладів	i18n/app.de.ts
Формат файлу	файл перекладу Qt Linguist

Типовий <i>Налаштування складників</i> Weblate при використанні як одномовного файла	
Маска файлів	i18n/app.*.ts
Основний файл для одномовного перекладу	i18n/app.en.ts
Шаблон для нових перекладів	i18n/app.en.ts
Формат файлу	файл перекладу Qt Linguist

Дивись також:

Підручник з Qt Linguist, Qt .ts, *Двомовні і одномовні формати*

1.10.13 Рядкові ресурси Android

Специфічний для Android формат файлів для перекладу застосунків.

Рядкові ресурси Android є одномовними. Файл *Основний файл для одномовного перекладу* зберігається у іншому місці, якщо порівнювати із іншими файлами — res/values/strings.xml.

Типовий <i>Налаштування складників</i> Weblate	
Маска файлів	res/values-*/strings.xml
Основний файл для одномовного перекладу	res/values/strings.xml
Шаблон для нових перекладів	Порожньо
Формат файлу	рядковий ресурс Android

Дивись також:

Документація для рядкових ресурсів Android, [Android string resources](#)

Примітка: У поточній версії підтримки *string-array* Android не передбачено. Щоб обійти це обмеження, ви можете розбити ваші рядкові масиви на частини:

```
<string-array name="several_strings">
  <item>First string</item>
  <item>Second string</item>
</string-array>
```

стане:

```
<string-array name="several_strings">
  <item>@string/several_strings_0</item>
  <item>@string/several_strings_1</item>
</string-array>
<string name="several_strings_0">First string</string>
<string name="several_strings_1">Second string</string>
```

string-array, який вказує на елементи *string*, має зберігатися в іншому файлі і не повинен бути доступним для перекладу.

Цей скрипт може допомогти у попередній обробці наявних файлів *strings.xml* і перекладів: <https://gist.github.com/paour/11291062>

Підказка: Щоб уникнути перекладу деяких рядків, їх можна позначити як непридатні до перекладу. Це може бути особливо корисно для посилань:

```
<string name="foobar" translatable="false">@string/foo</string>
```

1.10.14 Рядки Apple iOS

Формат файлів, який типово використовують для перекладу застосунків Apple iOS, але також стандартизований PWG 5100.13 і використаний у NeXTSTEP/OpenSTEP.

Рядки Apple iOS, зазвичай, використовуються як одномовні переклади.

Типовий <i>Налаштування складників</i> Weblate	
Маска файлів	Resources/*.lproj/Localizable.strings
Основний файл для одномовного перекладу	Resources/en.lproj/Localizable.strings або Resources/Base.lproj/Localizable.strings
Шаблон для нових перекладів	Порожньо
Формат файлу	рядки iOS (UTF-8)

Дивись також:

Документація із `.ref:stringsdict`, «рядкових файлів» Apple <<https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/BPInternational/MaintainingYourOwnStringsFiles/MaintainingYourOwnStringsFiles.html>>, формат файлів каталогів повідомлень у PWG 5100.13, Mac OS X strings

1.10.15 Рядки PHP

Переклади PHP є, зазвичай, одномовними, тому рекомендуємо вказувати базовий файл, яким (здебільшого) є файл з рядками англійською мовою.

Приклад файла:

```
<?php
$LANG['foo'] = 'bar';
$LANG['foo1'] = 'foo bar';
$LANG['foo2'] = 'foo bar baz';
$LANG['foo3'] = 'foo bar baz bag';
```

Типовий *Налаштовування складників Weblate*

Маска файлів	lang/*/texts.php
Основний файл для одномовного перекладу	lang/en/texts.php
Шаблон для нових перекладів	lang/en/texts.php
Формат файлу	<i>Рядки PHP</i>

Рядки PHP Laravel

Змінено в версії 4.1.

Передбачено підтримку файлів локалізації PHP Laravel, а також форм множини:

```
<?php
return [
    'welcome' => 'Welcome to our application',
    'apples' => 'There is one apple|There are many apples',
];
```

Дивись також:

PHP, Локалізація у Laravel

1.10.16 файли JSON

Нове в версії 2.0.

Змінено в версії 2.16: З версії Weblate 2.16 і принаймні з версії [translate-toolkit 2.2.4](#) передбачено підтримку і вкладеної структури файлів JSON.

Змінено в версії 4.3: Програма належним чином зберігає структуру файла JSON, навіть для складних випадків, які було важко обробити у попередніх випусках.

Формат JSON використовується здебільшого для перекладу програм, які реалізовано мовою JavaScript.

У поточній версії Weblate передбачено підтримку декількох варіантів перекладів JSON:

- Прості файли ключ-значення, які використовують, наприклад, у *vue-i18n* або *react-intl*.
- Файли із вкладеними ключами.
- *Файли JSON i18next*
- *Файли JSON go-i18n*
- *Файли JSON gotext*
- *JSON WebExtension*
- *Файл ARB*

Переклади JSON є, зазвичай, одномовними, тому рекомендуємо вказувати базовий файл, яким (здебільшого) є файл з рядками англійською мовою.

Приклад файла:

```
{
  "Hello, world!\n": "Ahoj světe!\n",
  "Orangutan has %d banana.\n": "",
  "Try Weblate at https://demo.weblate.org/!\n": "",
  "Thank you for using Weblate.": ""
}
```

Підтримку файлів із вкладеністю також реалізовано (див. вимоги вище). Такий файл може виглядати ось так:

```
{
  "weblate": {
    "hello": "Ahoj světe!\n",
    "orangutan": "",
    "try": "",
    "thanks": ""
  }
}
```

Підказка: Варіанти *файл JSON* та *файл JSON із вкладеною структурою* призначено для файлів одного типу. Використання обох варіантів надає змогу зберегти наявну структуру JSON під час перекладу.

Єдиною відмінністю є обробка додавання нових рядків за допомогою Weblate. У варіанті із вкладеністю програма робить спробу обробити ключ і вставити новий рядок у відповідну структуру. Наприклад, ключ `app.name` буде вставлено так:

```
{
  "app": {
    "name": "Weblate"
  }
}
```

Типовий <i>Налаштовування складників</i> Weblate	
Маска файлів	<code>langs/translation-*.json</code>
Основний файл для одномовного перекладу	<code>langs/translation-en.json</code>
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл JSON із вкладеною структурою</i>

Дивись також:

[JSON](#), [updating-target-files](#), [Налагодити виведення JSON](#), [Очищення файлів перекладу](#),

1.10.17 Файли JSON i18next

Змінено в версії 2.17: З версії Weblate 2.17 і принаймні з версії [translate-toolkit 2.2.5](#) передбачено підтримку файлів JSON i18next із формами множини.

Змінено в версії 4.15.1: Додано підтримку версії 4 цього формату.

Підказка: Якщо ви користуєтеся формами множини, рекомендуємо скористатися версією 4, оскільки у ній вирівняно з CLDR обробку таких форм. У попередніх версіях для деяких мови використовувалися інші форми множини, що не є правильним.

i18next — бібліотека для перекладу, яку написано на JavaScript і для JavaScript. У Weblate передбачено підтримку файлів локалізації цієї бібліотеки із різними можливостями, зокрема формами множини.

Переклади i18next є одномовними, тому рекомендуємо вказувати базовий файл, яким (здебільшого) є файл з рядками англійською мовою.

Примітка: У Weblate передбачено підтримку формату версій 3 і 4 JSON i18next. Будь ласка, виберіть належний формат файлів, що відповідає вашому середовищу.

Версії 2 і 1 є здебільшого сумісними з версією 3, за винятком способу обробки форм множини.

Приклад файла:

```
{
  "hello": "Hello",
  "apple": "I have an apple",
  "apple_plural": "I have {{count}} apples",
  "apple_negative": "I have no apples"
}
```

Типовий <i>Налаштовування складників</i> Weblate	
Маска файлів	langs/*.json
Основний файл для одномовного перекладу	langs/en.json
Шаблон для нових перекладів	Порожньо
Формат файлу	Файл JSON i18next v3

Дивись також:

JSON, формат JSON i18next, updating-target-files, *Налагодити виведення JSON*, *Очищення файлів перекладу*

1.10.18 Файли JSON go-i18n

Нове в версії 4.1.

Змінено в версії 4.16: Додано підтримку версії 2 цього формату.

Переклади go-i18n є одномовними, тому рекомендуємо вказувати базовий файл, яким (здебільшого) є файл з рядками англійською мовою.

Примітка: У Weblate передбачено підтримку формату версій 1 і 2 JSON go-i18n. Будь ласка, виберіть належний формат файлів, що відповідає вашому середовищу.

Типовий <i>Налаштовування складників</i> Weblate для версії 1	
Маска файлів	langs/*.json
Основний файл для одномовного перекладу	langs/en.json
Шаблон для нових перекладів	Порожньо
Формат файлу	файл JSON go-i18n версії 1

Типовий <i>Налаштовування складників</i> Weblate для версії 2	
Маска файлів	langs/*.json
Основний файл для одномовного перекладу	langs/en.json
Шаблон для нових перекладів	Порожньо
Формат файлу	файл JSON go-i18n версії 2

Дивись також:

[JSON](#), [go-i18n](#), [updating-target-files](#), [Налагодити виведення JSON](#), [Очищення файлів перекладу](#),

1.10.19 Файли JSON gotext

Нове в версії 4.15.1.

Переклади gotext є одномовними, тому рекомендуємо вказувати базовий файл, яким (здебільшого) є файл з рядками англійською мовою.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	<code>internal/translations/locales/*/messages.gotext.json</code>
Основний файл для одномовного перекладу	<code>internal/translations/locales/en-GB/messages.gotext.json</code>
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл JSON gotext</i>

Дивись також:

[JSON](#), [I18n у Go](#): керування перекладами, [updating-target-files](#), [Налагодити виведення JSON](#), [Очищення файлів перекладу](#)

1.10.20 Файл ARB

Нове в версії 4.1.

Переклади ARB є одномовними, тому рекомендуємо вказувати базовий файл, яким (здебільшого) є файл з рядками англійською мовою.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	<code>lib/l10n/intl_*.arb</code>
Основний файл для одномовного перекладу	<code>lib/l10n/intl_en.arb</code>
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл ARB</i>

Дивись також:

[JSON](#), [Специфікація пакунків ресурсів застосунків](#), [Інтернаціоналізація застосунків Flutter](#), [updating-target-files](#), [Налагодити виведення JSON](#), [Очищення файлів перекладу](#)

1.10.21 JSON WebExtension

Нове в версії 2.16: Підтримку цих файлів передбачено з версії Weblate 2.16 і принаймні версії [translate-toolkit 2.2.4](#).

Формат файлів, який використовується для перекладу розширень до Mozilla Firefox або Google Chromium.

Примітка: Хоча цей формат має назву JSON, його специфікацією передбачено можливість включення коментарів, які не є частиною специфікації JSON. У поточній версії Weblate не передбачено підтримки файлів із коментарями.

Приклад файла:

```
{
  "hello": {
    "message": "Ahoj světe!\n",
    "description": "Description",
    "placeholders": {
      "url": {
        "content": "$1",
        "example": "https://developer.mozilla.org"
      }
    }
  },
  "orangutan": {
    "message": "Orangutan has $count$ bananas",
    "description": "Description",
    "placeholders": {
      "count": {
        "content": "$1",
        "example": "5"
      }
    }
  },
  "try": {
    "message": "",
    "description": "Description"
  },
  "thanks": {
    "message": "",
    "description": "Description"
  }
}
```

Типовий <i>Налаштування складників Weblate</i>	
Маска файлів	<code>_locales/*/messages.json</code>
Основний файл для одномовного перекладу	<code>_locales/en/messages.json</code>
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл JSON WebExtension</i>

Дивись також:

JSON, Google chrome.i18n, Переклад розширень Mozilla

1.10.22 Файли ресурсів .XML

Нове в версії 2.3.

Файл ресурсів .XML (.resx) використовується як одномовний формат файлів XML у застосунках Microsoft .NET. Він є взаємозамінним із .resw, якщо використовується ідентичний до .resx синтаксис.

Типовий <i>Налаштування складників Weblate</i>	
Маска файлів	<code>Resources/Language.*.resx</code>
Основний файл для одномовного перекладу	<code>Resources/Language.resx</code>
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл ресурсів .NET</i>

Дивись також:

.NET Resource files (.resx), updating-target-files, *Очищення файлів перекладу*

1.10.23 Файли ResourceDictionary

Нове в версії 4.13.

ResourceDictionary є форматом одномовних файлів XML, який використовують для пакування ресурсів придатних до перекладу рядків для застосунків Windows Presentation Foundation (WPF).

Типовий <i>Налаштовування складників</i> Weblate	
Маска файлів	Languages/* .xaml
Основний файл для одномовного перекладу	Language/en .xaml
Шаблон для нових перекладів	Порожньо
Формат файлу	файл ResourceDictionary

Дивись також:

Flat XML, Прості файли XML, updating-target-files, Очищення файлів перекладу

1.10.24 Файли CSV

Нове в версії 2.4.

У файлах CSV можуть міститися прості списки початкових рядків і рядків перекладів. У Weblate передбачено підтримку таких файлів:

- Файли із визначенням полів у заголовку (location, source, target, ID, fuzzy, context, translator_comments, developer_comments). Цей підхід є рекомендованим, оскільки зменшує ймовірність помилки. Виберіть формат файлів *файл CSV*.
- Файли із двома полями — початковим рядком і перекладом (саме у такому порядку) — виберіть *Простий файл CSV* у полі формату файлів.
- Файли без заголовків із полями у порядку визначеному в [translate-toolkit](#): location, source, target, ID, fuzzy, context, translator_comments, developer_comments. Виберіть формат файлів *файл CSV*.
- Не забудьте визначити *Основний файл для одномовного перекладу* коли ваші файли одномовні (гляньте *Двомовні і одномовні формати*).

Підказка: Типово, у форматі CSV не передбачено автоматичного виявлення кодування символів у файлі. Іноді таке автоматичне виявлення не спрацьовує і спричиняє проблеми зі швидкодією. Будь ласка, виберіть варіант форматування файла з кодуванням, щоб уникнути цього (приклад: *файл CSV (UTF-8)*).

Попередження: Обробник формату CSV у поточній версії автоматично виявляє діалект файла CSV. У деяких випадках автоматичне виявлення може бути невдалим — ви отримаєте неоднозначні підсумки. Це, зокрема, стосується файлів CSV із символами нових рядків у значеннях. Щоб обійти проблеми, рекомендуємо уникати використання символів лапок.

Приклад файла:

Thank you for using Weblate.,Děkujeme za použití Weblate.

Типово <i>Налаштовування складників</i> Weblate для двомовних CSV	
Маска файлів	locale/*.csv
Основний файл для одномовного перекладу	<i>Порожньо</i>
Шаблон для нових перекладів	locale/en.csv
Формат файлу	<i>Файл CSV</i>

Типово <i>Налаштовування складників</i> Weblate для одномовних CSV	
Маска файлів	locale/*.csv
Основний файл для одномовного перекладу	locale/en.csv
Шаблон для нових перекладів	locale/en.csv
Формат файлу	<i>Простий файл CSV</i>

Багатозначний файл CSV

Нове в версії 4.13.

У цьому варіанті файлів CSV уможливлено зберігання декількох перекладів одного рядка.

Дивись також:

[CSV](#)

1.10.25 Файли YAML

Нове в версії 2.9.

Звичайні файли YAML із рядковими ключами і значеннями. Крім того, Weblate видобуває рядки зі списків або словників.

Приклад файла YAML:

```
weblate:
  hello: ""
  orangutan: ""
  try: ""
  thanks: ""
```

Типовий <i>Налаштовування складників</i> Weblate	
Маска файлів	translations/messages/*.yaml
Основний файл для одномовного перекладу	translations/messages.en.yaml
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл YAML</i>

Дивись також:

[YAML](#), [Файли YAML Ruby](#)

1.10.26 Файли YAML Ruby

Нове в версії 2.9.

Файли YAML інтернаціоналізації Ruby із кореневим вузлом «language».

Приклад файла перекладу YAML Ruby:

```
cs:
  weblate:
    hello: ""
    orangutan: ""
    try: ""
    thanks: ""
```

Типовий <i>Налаштовування складників</i> Weblate	
Маска файлів	translations/messages/*.yaml
Основний файл для одномовного перекладу	translations/messages.en.yaml
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл YAML Ruby</i>

Дивись також:

[YAML](#), [Файли YAML](#)

1.10.27 Файли DTD

Нове в версії 2.18.

Приклад файла DTD:

```
<!ENTITY hello "">
<!ENTITY orangutan "">
<!ENTITY try "">
<!ENTITY thanks "">
```

Типовий <i>Налаштовування складників</i> Weblate	
Маска файлів	locale/*.dtd
Основний файл для одномовного перекладу	locale/en.dtd
Шаблон для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл DTD</i>

Дивись також:

[Mozilla DTD format](#)

1.10.28 Прості файли XML

Нове в версії 3.9.

Приклад простого файлу XML:

```
<?xml version='1.0' encoding='UTF-8'?>
<root>
  <str key="hello_world">Hello World!</str>
  <str key="resource_key">Translated value.</str>
</root>
```

Типовий *Налаштовування складників* Weblate

Маска файлів	locale/*.xml
Основний файл для одномовного перекладу	locale/en.xml
Шаблон для нових перекладів	Порожньо
Формат файлу	простий файл XML

Дивись також:

[Flat XML](#)

1.10.29 Файли RC Windows

Змінено в версії 4.1: Підтримку файлів RC Windows було переписано.

Примітка: Підтримка цього формату зараз перебуває на тестовому етапі. Будемо раді тестуванню та відгукам.

Приклад файлу RC Windows:

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT

STRINGTABLE
BEGIN
    IDS_MSG1          "Hello, world!\n"
    IDS_MSG2          "Orangutan has %d banana.\n"
    IDS_MSG3          "Try Weblate at http://demo.weblate.org/!\n"
    IDS_MSG4          "Thank you for using Weblate."
END
```

Типовий *Налаштовування складників* Weblate

Маска файлів	lang/*.rc
Основний файл для одномовного перекладу	lang/en-US.rc
Шаблон для нових перекладів	lang/en-US.rc
Формат файлу	Файл RC

Дивись також:

[Windows RC files](#)

1.10.30 Файли метаданих крамниці застосунків

Нове в версії 3.5.

Метадані, що використовуються для публікації застосунків у різних крамницях застосунків, можна перекладати. У поточній версії сумісними є такі інструменти:

- [Triple-T gradle-play-publisher](#)
- [Fastlane](#)
- [F-Droid](#)

Метадані складаються із декількох текстових файлів, які Weblate показуватиме як окремі рядки для перекладу.

Типовий <i>Налаштовування складників</i> Weblate	
Маска файлів	fastlane/android/metadata/*
Основний файл для одномовного перекладу	fastlane/android/metadata/en-US
Шаблон для нових перекладів	fastlane/android/metadata/en-US
Формат файлу	„Файли метаданих крамниці застосунків“

Підказка: Якщо ви не хочете, щоб певні рядки було перекладено (наприклад, усунути з перекладу записи журналу змін), позначте ці рядки як придатні лише для читання (див. *Налаштовування поведінки за допомогою прапорців*). Процедру можна автоматизувати за допомогою *Масове редагування*.

1.10.31 Файли субтитрів

Нове в версії 3.7.

За допомогою Weblate можна перекладати різноманітні файли субтитрів:

- Файл субтитрів SubRip (*.srt)
- Файл субтитрів MicroDVD (*.sub)
- Файл субтитрів Advanced Substation Alpha (*.ass)
- Файл субтитрів Substation Alpha (*.ssa)

Типовий <i>Налаштовування складників</i> Weblate	
Маска файлів	path/*.srt
Основний файл для одномовного перекладу	path/en.srt
Шаблон для нових перекладів	path/en.srt
Формат файлу	<i>файл субтитрів SubRip</i>

Дивись також:

[Subtitles](#)

1.10.32 Excel Open XML

Нове в версії 3.2.

Файли Excel Open XML (.xlsx) може бути імпортовано і експортовано.

При вивантаженні файлів XLSX для перекладу слід зважати на те, що оброблятиметься лише активний робочий аркуш. У файлі має бути принаймні один стовпчик із назвою `source` (у якому міститься початковий рядок) і стовпчик із назвою `target` (у якому міститься переклад). Крім того, має бути стовпчик із назвою `context` (у якому міститься контекстний шлях рядка перекладу). Якщо ви використовуєте отримання XLSX для експортування перекладів до робочої книги Excel, у вас вже є файл у належному форматі.

1.10.33 Файли HTML

Нове в версії 4.1.

Примітка: Підтримка цього формату зараз перебуває на тестовому етапі. Будемо раді тестуванню та відгукам.

Придатні до перекладу дані буде видобуто з файлів HTML і запропоновано до перекладу.

Дивись також:

[HTML](#)

1.10.34 Текстові файли

Нове в версії 4.6.

Примітка: Підтримка цього формату зараз перебуває на тестовому етапі. Будемо раді тестуванню та відгукам.

Придатні до перекладу дані буде видобуто з простих текстових файлів і запропоновано до перекладу. Кожен абзац перекладається як окремий рядок.

Існує три різновиди цього формату:

- Звичайний текстовий файл
- Текстовий файл DokuWiki
- Текстовий файл MediaWiki

Дивись також:

[Simple Text Documents](#)

1.10.35 Формат OpenDocument

Нове в версії 4.1.

Примітка: Підтримка цього формату зараз перебуває на тестовому етапі. Будемо раді тестуванню та відгукам.

Придатні до перекладу дані буде видобуто з файлів OpenDocument і запропоновано до перекладу.

Дивись також:

[OpenDocument Format](#)

1.10.36 Формат IDML

Нове в версії 4.1.

Примітка: Підтримка цього формату зараз перебуває на тестовому етапі. Будемо раді тестуванню та відгукам.

Придатні до перекладу дані буде видобуто з файлів мови розмітки Adobe InDesign і запропоновано до перекладу.

1.10.37 Формат TermBase eXchange

Нове в версії 4.5.

TBX — це формат XML для обміну термінологічними даними.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	tbx/*.*tbx
Основний файл для одномовного перекладу	Порожньо
Шаблон для нових перекладів	Порожньо
Формат файлу	Файл TermBase eXchange

Дивись також:

[TBX у Вікіпедії](#), [TBX](#), [Глосарій](#)

1.10.38 Формат Stringsdict

Нове в версії 4.8.

Примітка: Підтримка цього формату зараз перебуває на тестовому етапі. Будемо раді тестуванню та відгукам.

Заснований на XML формат, який використовує Apple і у якому передбачено можливість зберігання форм множини рядка.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файлів	Resources/*.*lproj/Localizable.stringsdict
Основний файл для одномовного перекладу	Resources/en.lproj/Localizable.stringsdict або Resources/Base.lproj/Localizable.stringsdict
Шаблон для нових перекладів	Порожньо
Формат файлу	Файл Stringsdict

Дивись також:

[Рядки Apple iOS](#), [файловий формат Stringsdict](#)

1.10.39 Формат Fluent

Нове в версії 4.8.

Примітка: Підтримка цього формату зараз перебуває на тестовому етапі. Будемо раді тестуванню та відгукам.

Fluent — одномовний текстовий формат, увагу у якому зосереджено на асиметричній локалізації: простий рядок однією мовою може відповідати складному багатоваріантному перекладу іншою мовою.

Типовий <i>Налаштовування складників</i> Weblate	
Маска файлів	locales/*/messages.ftl
Основний файл для одномовного перекладу	locales/en/messages.ftl
Шаблон для нових перекладів	Порожньо
Формат файлу	Файл <i>Fluent</i>

Дивись також:

[Сайт проєкту Fluent](#)

1.10.40 Підтримка інших форматів

Реалізувати підтримку більшості форматів, підтримку яких реалізовано у [translate-toolkit](#) і у яких передбачено підтримку перетворення у послідовну форму, доволі просто, але цю підтримку ще не достатньо перевірено. У більшості випадків потрібен певний тонкий шар у Weblate для приховування відмінностей у поведінці різних репозиторіїв даних [translate-toolkit](#).

Рекомендованим способом додавання нових форматів є реалізація їхньої підтримки у [translate-toolkit](#).

Дивись також:

[Translation Related File Formats](#)

1.11 Налаштування інтеграції із керуванням версіями

У поточній версії Weblate передбачено підтримку модулів керування версіями *Git* (із розширеною підтримкою *Запити щодо злиття GitHub*, *Запити щодо об'єднання GitLab*, *Запити щодо злиття Gitea*, *Gerrit*, *Subversion* і *Запит щодо злиття для сервера Bitbucket*) та *Mercurial*.

1.11.1 Доступ до репозиторіїв

Репозиторій системи керування версіями, яким ви хочете користуватися, має бути доступним для Weblate. Якщо репозиторій загальнодоступний, вам достатньо ввести правильну адресу (наприклад <https://github.com/WeblateOrg/weblate.git>), але для приватних репозиторіїв або адрес для запису налаштування є складнішим і вимагає розпізнавання користувача.

Доступ до репозиторіїв з Hosted Weblate

Для Hosted Weblate на GitHub, Bitbucket, Codeberg і GitLab зареєстровано особливого користувача для запису внесків (назва облікового запису — *weblate*, адреса електронної пошти — `hosted@weblate.org`, ім'я користувача *Weblate push user*). Вам слід додати цього користувача як учасника розробки і надати йому відповідні права доступу до вашого репозиторію (для клонування достатньо прав на читання, для запису потрібні права запису). Залежно від служби і параметрів вашої організації надання прав буде здійснено негайно або потребуватиме підтвердження з боку Weblate.

Запрошення для користувача `:guilabel:weblate`` на GitHub приймаються автоматично протягом п'яти хвилин. У інших службах може знадобитися обробка вручну, тому, будь ласка, майте терпіння.

Щойно користувача *weblate* буде додано, ви можете налаштувати *репозиторій початкового коду* і *адресу запису до репозиторію* з використання протоколу SSH (наприклад, `git@github.com:WeblateOrg/weblate.git`).


Репозиторії із доступом за SSH

Найпоширенішим способом доступу до приватних репозиторіїв є спосіб, який засновано на SSH. Уповноважте відкритий ключ SSH Weblate (див. *Ключ SSH Weblate*) на доступ до основного репозиторію у цей спосіб.

Попередження: На GitHub кожен ключ може бути використано лише один раз, див. *Репозиторії GitHub* і *Доступ до репозиторіїв з Hosted Weblate*.

Крім того, Weblate зберігає відбиток ключа вузла при першому з'єднанні і не може з'єднатися із вузлом, якщо ключ буде пізніше змінено (див. *Перевірка ключів SSH вузла*).

Якщо потрібні якісь коригування, виконайте їх за допомогою адміністративного інтерфейсу Weblate:


Weblate
Dashboard
Projects ▾
Languages ▾
Checks ▾

Manage / SSH keys

Weblate status
Backups
Translation memory
Performance report
SSH keys
Alerts
Repositories
Users
Teams

Appearance
Tools
Automatic suggestions
Billing

Public SSH key

Weblate uses SSH key to access remote repositories. The corresponding public key is found below, you can use it to grant Weblate access to a repository.

```
ssh-rsa
AAAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNZ8fKPCbs6hiYpnovamGbWdXygsRjmbGwJv0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhLwp1uYlW
Weblate
```

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-ed25519	+DIY3wvW6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

To access SSH hosts, its host key needs to be verified. You can get the host key by entering a domain name or IP for the host in the form below.

Hostname

Port

Submit

Powered by Weblate 4.16
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Ключ SSH Weblate

Відкритий ключ Weblate доступний до перегляду усіма користувачами на сторінці *Про програму*.

Адміністратори можуть створити або переглянути відкритий ключ, який використовується Weblate для з'єднання (з розділу *Ключі SSH*) на основній сторінці адміністративного інтерфейсу.

Примітка: У поточній версії відповідний закритий ключ SSH не захищено паролем, тому вам слід переко-
нати, що його добре захищено.

Підказка: Створіть резервну копію створено закритого ключа SSH Weblate.

Перевірка ключів SSH вузла

Weblate автоматично зберігає ключі SSH вузла за першого доступу та використовує їх у подальшому.

Якщо вам захочеться перевірити відбиток ключа до встановлення з'єднання з репозиторієм, додайте ключі SSH вузлів серверів, до яких ви збираєтеся отримати доступ у пункті *Додати ключ вузла* того самого розділу адміністративного інтерфейсу. Введіть назву вузла, до якого ви маєте намір отримати доступ (наприклад, `gitlab.com`) і натисніть кнопку *Надіслати*. Перевірте, чи відбиток збігається з доданим вами сервером.

Додані ключі із відбитками буде показано у повідомленні підтвердження:

[Dashboard](#)
[Projects](#)
[Languages](#)
[Checks](#)

[Manage / SSH keys](#)

Added host key for github.com with fingerprint p2QAMXNIC1TJYWeIOtrVc98/R1BUFWu3/LiyKgUfQM (ecdsa-sha2-nistp256), please verify that it is correct.

Added host key for github.com with fingerprint nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8 (ssh-rsa), please verify that it is correct.

Added host key for github.com with fingerprint +DiY3wvW6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU (ssh-ed25519), please verify that it is correct.

[Weblate status](#)
[Backups](#)
[Translation memory](#)
[Performance report](#)
[SSH keys](#)
[Alerts](#)
[Repositories](#)
[Users](#)
[Teams](#)

[Appearance](#)
[Tools](#)
[Automatic suggestions](#)
[Billing](#)

Public SSH key

Weblate uses SSH key to access remote repositories. The corresponding public key is found below, you can use it to grant Weblate access to a repository.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNZ8fKPCbs6hiYpnovamGbWDxygRSjmbGwjV0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhlLwp1uYlW
Weblate
```

[Download private key](#)

Known host keys

Hostname	Key type	Fingerprint
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOtrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-ed25519	+DiY3wvW6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

To access SSH hosts, its host key needs to be verified. You can get the host key by entering a domain name or IP for the host in the form below.

Hostname
Port

[Submit](#)

Репозиторії GitHub

Можливий доступ з використанням SSH (див. *Репозиторії із доступом за SSH*), але якщо вам потрібен доступ до кількох репозиторіїв, ви зіткнетеся з обмеженням GitHub на використання ключів SSH (оскільки кожен ключ можна використати лише один раз).

Якщо *гілку запису* не встановлено, буде створено відгалуження проекту, і зміни буде записано до відгалуження. Якщо гілку запису встановлено, зміни буде записано до основного репозиторію і вибраної гілки.

Для варіантів із малою кількістю проектів скористайтесь розпізнаванням HTTPS із особистим жетоном доступу та вашим обліковим записом GitHub, див. *Creating an access token for command-line use*.

Для варіантів із великою кількістю проектів зазвичай краще створити окремого користувача для Weblate, пов'язати його із відкритим ключем SSH, який створено на Weblate (див. *Ключ SSH Weblate*) і надати йому доступ до усіх репозиторіїв, які ви хочете перекладати. Цей підхід використано і на Hosted Weblate — маємо особливого користувача *weblate*.

Дивись також:

Доступ до репозиторіїв з Hosted Weblate

Внутрішні адреси Weblate

Щоб скористатися одним спільним репозиторієм для різних складників ви можете використати особливу адресу, `weblate://проект/складник` в інших (пов'язаних) складниках. Тоді, складник матиме спільні налаштування репозиторію системи керування версіями з еталонним складником.

Попередження: Вилучення основного складника призводить до вилучення пов'язаних складників.

Weblate автоматично коригує адресу репозиторію при створенні складника, якщо виявляє складник із відповідними налаштуваннями репозиторію. Ви можете перевизначити цю поведінку на останньому кроці налаштування складника.

Причини для використання внутрішніх адрес:

- Заощаджує місце на диску сервера — репозиторій зберігається лише один раз.
- Пришвидшує оновлення — оновлюється лише один репозиторій.
- Наявність лише одного експортованого репозиторію із перекладами Weblate (див. *Засіб експортування Git*).
- Деякі додатки можуть працювати із декількома складниками, які мають єдине спільний репозиторій, наприклад *Сполучити Git подання*.

Репозиторії HTTPS

Щоб отримати доступ до захищених репозиторіїв HTTPS, включіть до адреси ім'я користувача і пароль. Не слід хвилюватися — Weblate вилучить ці дані при показі адреси користувачам (якщо взагалі показуватиме їм адресу репозиторію).

Наприклад, адреса GitHub із доданими даними розпізнавання може виглядати так: `https://користувач:ваш_жетон_доступу@github.com/WeblateOrg/weblate.git`.

Примітка: Якщо ваше ім'я користувача і пароль містять спеціальні символи, їх слід записувати у URL-кодуванні, наприклад `https://user%40example.com:%24password%23@bitbucket.org/...`

Використання проксі-сервера

Якщо вам потрібен доступ HTTP/HTTPS до репозиторіїв системи керування версіями з використанням проксі-сервера, налаштуйте систему керування версіями на його використання.

Зробити це можна за допомогою змінних середовища `http_proxy`, `https_proxy` і `all_proxy` (як це описано у [документації до cURL](#)) або примусовим визначенням у налаштуваннях системи керування версіями. Приклад:

```
git config --global http.proxy http://user:password@proxy.example.com:80
```

Примітка: Налаштовування проксі слід виконати для користувача, від імені якого запущено Weblate (див. також [Права доступу у файловій системі](#)) і з `HOME=$DATA_DIR/home` (див. [DATA_DIR](#)), інакше Git, який запускатиме Weblate, не використовуватиме ці налаштування.

Дивись також:

Сторінка підручника cURL, Документація з налаштувань Git

1.11.2 Git

Підказка: Для роботи Weblate потрібен Git 2.12 або новіша версія.

Дивись також:

Див. розділ [Доступ до репозиторіїв](#), щоб дізнатися більше про доступ до репозиторіїв різних типів.

Git з примусовим «push»

Це працює як і сам Git. Єдиною відмінністю є те, що запис завжди виконується примусово. Такий режим призначено лише для випадку використання окремого репозиторію для перекладів.

Попередження: Будьте обережні із використанням, оскільки так можна дуже просто втратити внески до вашого основного репозиторію.

Налаштовування Git

Weblate викликає усі команди VCS з використанням `HOME=$DATA_DIR/home` (див. [DATA_DIR](#)), тому редагування налаштувань користувача слід виконувати у `DATA_DIR/home/.git`.

Віддалені допоміжні засоби Git

Ви можете скористатися [remote helpers](#) Git для додаткової підтримки інших систем керування версіями, але вам слід бути готовими до діагностики проблем, до яких може призвести використання цих засобів.

У поточній версії доступні допоміжні засоби для Bazaar і Mercurial у окремих репозиторіях на GitHub: [git-remote-hg](#) і [git-remote-bzr](#). Отримайте їх вміст вручну і запишіть до ваших шляхів пошуку (наприклад, до `~/bin`). Переконайтеся, що у вашій системі встановлено відповідні засоби систем керування версіями.

Щойно ці засоби буде встановлено, такими віддаленими засобами можна скористатися для визначення репозиторію у Weblate.

Клонування проекту `gnuhello` з Launchpad за допомогою Bazaar:

```
bzr::lp:gnuhello
```

Для репозиторію `hello` з `selenic.com` з використанням Mercurial:

```
hg::http://selenic.com/repo/hello
```

Попередження: Незручності використання віддалених допоміжних засобів Git виникають, наприклад, з Mercurial, віддалений допоміжний засіб іноді створює новий пункт при записі змін до основної репозиторію.

1.11.3 Запити щодо злиття GitHub

Нове в версії 2.3.

Це додає тонкий шар над [Git](#) з використанням інструмента [GitHub API](#), щоб уможливити запис змін у перекладах як запитів щодо об'єднання, замість безпосереднього запису до репозиторію.

[Git](#) записує зміни безпосередньо до репозиторію, а [Запити щодо злиття GitHub](#) створює запити щодо об'єднання. Останній не потрібен для простого доступу до репозиторіїв Git.

Щоб це спрацювало, вам слід налаштувати реєстраційні дані для програмного інтерфейсу (`GITHUB_CREDENTIALS`) у параметрах Weblate. Після налаштування ви побачите пункт [GitHub](#) при виборі *системи керування версіями складника*.

Дивись також:

[Записування змін з Weblate, `GITHUB_CREDENTIALS`](#)

1.11.4 Запити щодо об'єднання GitLab

Нове в версії 3.9.

Це просто додає тонкий шар над [Git](#) з використанням інструмента [GitLab API](#), щоб уможливити запис змін у перекладах як запитів щодо об'єднання, замість безпосереднього запису до репозиторію.

Потреби у використанні цього інструмента для доступу до репозиторіїв Git, звичайний [Git](#) працює так само. Єдиною відмінністю є спосіб обробки запису до репозиторію. З [Git](#) зміни записуються безпосередньо до репозиторію, а [Запити щодо об'єднання GitLab](#) створює запит щодо об'єднання.

Щоб це спрацювало, вам слід налаштувати реєстраційні дані для програмного інтерфейсу (`GITLAB_CREDENTIALS`) у параметрах Weblate. Після налаштування ви побачите пункт [GitLab](#) при виборі *системи керування версіями складника*.

Дивись також:

[Записування змін з Weblate, `GITLAB_CREDENTIALS`](#)

1.11.5 Запити щодо злиття Gitea

Нове в версії 4.12.

Це додає тонкий шар над [Git](#) з використанням інструмента [Gitea API](#), щоб уможливити запис змін у перекладах як запитів щодо об'єднання, замість безпосереднього запису до репозиторію.

Потреби у використанні цього інструмента для доступу до репозиторіїв Git, звичайний [Git](#) працює так само. Єдиною відмінністю є спосіб обробки запису до репозиторію. З [Git](#) зміни записуються безпосередньо до репозиторію, а [Запити щодо злиття Gitea](#) створює запит щодо об'єднання.

Щоб це спрацювало, вам слід налаштувати реєстраційні дані для програмного інтерфейсу (*GITEA_CREDENTIALS*) у параметрах Weblate. Після налаштування ви побачите пункт *Gitea* при виборі *системи керування версіями складника*.

Дивись також:

Записування змін з Weblate, GITEA_CREDENTIALS

1.11.6 Запит щодо злиття для сервера Bitbucket

Нове в версії 4.16.

Це додає тонкий шар над *Git* з використанням інструмента програмного інтерфейсу Bitbucket, щоб уможливити запис змін у перекладах як запитів щодо об'єднання, замість безпосереднього запису до репозиторію.

Попередження: Тут не передбачено підтримки хмарного програмного інтерфейсу Bitbucket.

Потреби у використанні цього інструмента для доступу до репозиторіїв Git, звичайний *Git* працює так само. Єдиною відмінністю є спосіб обробки запису до репозиторію. З *Git* зміни записуються безпосередньо до репозиторію, а *Запит щодо злиття для сервера Bitbucket* створює запит щодо об'єднання.

Щоб це спрацювало, вам слід налаштувати реєстраційні дані для програмного інтерфейсу (*BITBUCKETSERVER_CREDENTIALS*) у параметрах Weblate. Після налаштування ви побачите пункт *сервер Bitbucket* при виборі *системи керування версіями складника*.

Дивись також:

Записування змін з Weblate, BITBUCKETSERVER_CREDENTIALS

1.11.7 Запити щодо об'єднання Pagure

Нове в версії 4.3.2.

Це просто додає тонкий шар над *Git* з використанням інструмента *Pagure API*, щоб уможливити запис змін у перекладах як запитів щодо об'єднання, замість безпосереднього запису до репозиторію.

Потреби у використанні цього інструмента для доступу до репозиторіїв Git, звичайний *Git* працює так само. Єдиною відмінністю є спосіб обробки запису до репозиторію. З *Git* зміни записуються безпосередньо до репозиторію, а *Запити щодо об'єднання Pagure* створює запит щодо об'єднання.

Щоб це спрацювало, вам слід налаштувати реєстраційні дані для програмного інтерфейсу (*PAGURE_CREDENTIALS*) у параметрах Weblate. Після налаштування ви побачите пункт *Pagure* при виборі *системи керування версіями складника*.

Дивись також:

Записування змін з Weblate, PAGURE_CREDENTIALS

1.11.8 Gerrit

Нове в версії 2.2.

Це додає тонкий шар над *Git* з використанням інструмента *git-review*, щоб уможливити запис змін у перекладах як запитів щодо рецензування Gerrit, замість безпосереднього запису до репозиторію.

У документації до Gerrit наведено подробиці щодо налаштувань, потрібних для таких репозиторіїв.

1.11.9 Mercurial

Нове в версії 2.1.

Ще однією системою керування версіями, якою ви можете користуватися безпосередньо у Weblate є Mercurial.

Примітка: Це має працювати із будь-якою версією Mercurial, але іноді можна зіткнутися із несумісними змінами у інтерфейсі командного рядка, які шкодять інтеграції із Weblate.

Дивись також:

Див. розділ *Доступ до репозиторіїв*, щоб дізнатися більше про доступ до репозиторіїв різних типів.

1.11.10 Subversion

Нове в версії 2.8.

Weblate використовує для взаємодії з репозиторіями [subversion git-svn](#). Це скрипт мовою Perl, який надає змогу користуватися subversion за допомогою клієнта Git, уможливаючи для користувачів супровід повного клону внутрішнього репозиторію і локальні внески.

Примітка: Weblate намагається виявити компонування репозиторію Subversion автоматично — передбачено підтримку як безпосередніх адрес для гілок, так і репозиторію зі стандартним компонуванням (branches/, tags/ і trunk/). Докладніше про це можна дізнатися з [документації до git-svn](#). Якщо компонування вашого репозиторію не є стандартним і ви стикаєтеся з помилками, спробуйте включити назву гілки у адресу репозиторію і лишити гілку порожньою.

Змінено в версії 2.19: До цього було передбачено лише підтримку репозиторіїв зі стандартним компонуванням.

Реєстраційні дані Subversion

Weblate очікує, що ви прийняли сертифікат і, якщо потрібно, вказали ваші реєстраційні дані. Weblate вставить їх до каталогу [DATA_DIR](#). Прийміть сертифікат за допомогою одноразового запуску `svn` зі змінною середовища `$HOME`, яка збігається за значенням з [DATA_DIR](#):

```
# Use DATA_DIR as configured in Weblate settings.py, it is /app/data in the Docker
HOME=${DATA_DIR}/home svn co https://svn.example.com/example
```

Дивись також:

[DATA_DIR](#)

1.11.11 Локальні файли

1.11.12 Git

Підказка: Нижче використано [Git](#). Для роботи слід встановити Git. Уможливило перехід на безпосереднє використання Git із повним журналом перекладів.

Нове в версії 3.8.

Крім того, Weblate може працювати без віддаленої системи керування версіями. Початкові переклади імпортуються безпосереднім вивантаженням. Пізніше ви можете замінити окремі файли вивантаженням нових версій або додати рядки перекладу безпосередньо з Weblate (у поточній версії ця можливість реалізовано лише для одномовних перекладів).

За лаштунками, Weblate створює репозиторій Git і стежить за усіма змінами за його допомогою. Якщо ви згодом вирішите скористатися системою керування версіями для зберігання ваших перекладів, у вас вже буде репозиторій у Weblate, на основі якого ви зможете побудувати інтеграцію.

1.12 Програмний інтерфейс REST Weblate

Нове в версії 2.6: Програмний інтерфейс REST доступний починаючи з версії Weblate 2.6.

Програмний інтерфейс доступний через `/api/` в URL і побудовано на [бібліотеці REST Django](#). Можете користуватися ним безпосередньо або через [Клієнт Weblate](#).

1.12.1 Розпізнавання і типові параметри

Відкритий програмний інтерфейс проекту доступний без розпізнавання користувача, хоча частоту запитів користувачів, які не пройшли розпізнавання, значно зменшено (типово, до 100 запитів на добу). Через це рекомендуємо вам користуватися розпізнаванням. У розпізнаванні використовується ключ, який ви можете отримати у вашому профілі. Використовуйте його в заголовку `Authorization`:

ANY /

Типова поведінка запиту для програмного інтерфейсу, заголовків, кодів стану та параметрів у цьому розділі стосується також усі інших кінцевих точок.

Параметри запиту

- **format** – Формат відповіді (перевизначає `Accept`). Можливі значення залежать від налаштування бібліотеки REST. Типово, передбачено підтримку `json` і `api`. Останнє надає інтерфейсу браузера для програмного інтерфейсу.
- **page** – Повертає вказану сторінку розбитих на сторінки результатів (скористайтесь полями `next` у `previous` у відповіді, щоб автоматизувати навігацію).

Заголовки запиту

- `Accept` – тип вмісту відповіді залежить від заголовка `Accept`
- `Authorization` – необов'язковий ключ для розпізнавання як-от `Authorization: Token YOUR-TOKEN`

Заголовки відповіді

- `Content-Type` – це залежить від заголовка `Accept` запиту
- `Allow` – список дозволених методів HTTP на об'єкті

Об'єкт відповіді JSON

- **detail** (*string*) – докладний опис результатів (для кодів станів HTTP, відмінних від 200 OK)
- **count** (*int*) – загальна кількість записів для списків об'єктів
- **next** (*string*) – адреса наступної сторінки для списків об'єктів
- **previous** (*string*) – адреса попередньої сторінки для списків об'єктів
- **results** (*array*) – підсумки для списку об'єктів
- **url** (*string*) – Адреса для доступу до цього ресурсу за допомогою програмного інтерфейсу

- **web_url** (*string*) – Адреса для доступу до цього ресурсу за допомогою браузера

Коди станів

- **200 OK** – якщо запит було оброблено коректно
- **201 Created** – якщо новий об'єкт було успішно створено
- **204 No Content** – якщо об'єкт було успішно вилучено
- **400 Bad Request** – якщо не вказано параметри форми
- **403 Forbidden** – якщо заборонено доступ
- **429 Too Many Requests** – якщо увімкнено обмеження швидкості

Жетони розпізнавання

Змінено в версії 4.10: Жетони рівня проекту було впроваджено у випуску 4.10.

У кожного користувача є власний жетон доступу, який можна отримати зі сторінки профілю користувача. Новостворені жетони користувачів мають префікс `wlu_`.

Передбачено можливість створення жетонів для доступу до програмного інтерфейсу на рівні певного проєкту. Ці жетони можна розпізнати за префіксом `wlp_`.

Приклади розпізнавання

Приклад запиту:

```
GET /api/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
Authorization: Token YOUR-TOKEN
```

Приклад відповіді:

```
HTTP/1.0 200 OK
Date: Fri, 25 Mar 2016 09:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, HEAD, OPTIONS

{
  "projects": "http://example.com/api/projects/",
  "components": "http://example.com/api/components/",
  "translations": "http://example.com/api/translations/",
  "languages": "http://example.com/api/languages/"
}
```

Приклад для CURL:

```
curl \
  -H "Authorization: Token TOKEN" \
  https://example.com/api/
```

Приклади передавання параметрів

Для методу `POST` параметри можна вказати як подану форму (*application/x-www-form-urlencoded*) або як JSON (*application/json*).

Приклад запиту форми:

```
POST /api/projects/hello/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Token TOKEN

operation=pull
```

Приклад запиту JSON:

```
POST /api/projects/hello/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{"operation": "pull"}
```

Приклад для CURL:

```
curl \
  -d operation=pull \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/components/hello/weblate/repository/
```

Приклад JSON для CURL:

```
curl \
  --data-binary '{"operation": "pull"}' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/components/hello/weblate/repository/
```

Обмеження частоти у програмному інтерфейсі

Запити до програмного інтерфейсу обмежено за частотою: за типових налаштувань можна виконувати не більше 100 запитів на день для анонімних користувачів і не більше 5000 запитів на годину для користувачів, які пройшли розпізнавання у системі.

Обмеження за частотою можна скоригувати у файлі `settings.py`; див. [дроселювання у документації до бібліотек REST Django](#), щоб дізнатися більше про те, як налаштувати це обмеження.

У контейнері Docker домен сайту можна налаштувати за допомогою `WEBLATE_API_RATELIMIT_ANON` та `WEBLATE_API_RATELIMIT_USER`.

Стан обмеження за частотою повідомляється за допомогою таких частин заголовка:

X-RateLimit-Limit	Обмеження за частотою запитів
X-RateLimit-Remaining	Залишок до обмеження на запити
X-RateLimit-Reset	Кількість секунд до скидання обмеження за частотою у вікні обробки

Змінено в версії 4.1: Додано частини заголовка, пов'язані зі станом засобу обмеження за частотою.

Дивись також:

Обмеження частоти, Обмеження частоти, `WEBLATE_API_RATELIMIT_ANON`,
`WEBLATE_API_RATELIMIT_USER`

1.12.2 Точка входу до програмного інтерфейсу

GET `/api/`

Коренева точка входу до програмного інтерфейсу.

Приклад запиту:

```
GET /api/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
Authorization: Token YOUR-TOKEN
```

Приклад відповіді:

```
HTTP/1.0 200 OK
Date: Fri, 25 Mar 2016 09:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, HEAD, OPTIONS

{
  "projects": "http://example.com/api/projects/",
  "components": "http://example.com/api/components/",
  "translations": "http://example.com/api/translations/",
  "languages": "http://example.com/api/languages/"
}
```

1.12.3 Користувачі

Нове в версії 4.0.

GET `/api/users/`

Повертає список користувачів, якщо у вас є права доступу до перегляду керування користувачами. Якщо у вас немає таких прав, ви зможете переглядати подробиці лише щодо свого облікового запису.

Дивись також:

Атрибути об'єкта користувачів описано у `GET /api/users/(str:username)/`.

POST `/api/users/`

Створює обліковий запис користувача.

Параметри

- **username** (*string*) – Ім'я користувача
- **full_name** (*string*) – Повне ім'я користувача
- **email** (*string*) – Адреса електронної пошти користувача
- **is_superuser** (*boolean*) – Чи є користувач надкористувачем? (необов'язковий)
- **is_active** (*boolean*) – Чи є користувач активним? (необов'язковий)

- **is_bot** (*boolean*) – Чи є користувач ботом? (необов'язкове) (використовується для жетонів області проекту)

GET `/api/users/ (str: username) /`

Повертає відомості щодо користувачів.

Параметри

- **username** (*string*) – Назва облікового запису користувача

Об'єкт відповіді JSON

- **username** (*string*) – назва запису користувача
- **full_name** (*string*) – повне ім'я користувача
- **email** (*string*) – адреса електронної пошти користувача
- **is_superuser** (*boolean*) – чи є користувач надкористувачем
- **is_active** (*boolean*) – чи є користувач активним
- **is_bot** (*boolean*) – чи є користувач ботом (використовується для жетонів рівня проекту)
- **date_joined** (*string*) – дата створення запису користувача
- **groups** (*array*) – посилання на пов'язані групи; див. `GET /api/groups/ (int:id) /`

Приклад даних JSON:

```
{
  "email": "user@example.com",
  "full_name": "Example User",
  "username": "exampleusername",
  "groups": [
    "http://example.com/api/groups/2/",
    "http://example.com/api/groups/3/"
  ],
  "is_superuser": true,
  "is_active": true,
  "is_bot": false,
  "date_joined": "2020-03-29T18:42:42.617681Z",
  "url": "http://example.com/api/users/exampleusername/",
  "statistics_url": "http://example.com/api/users/exampleusername/statistics/"
}
```

PUT `/api/users/ (str: username) /`

Змінює параметри користувача.

Параметри

- **username** (*string*) – Назва облікового запису користувача

Об'єкт відповіді JSON

- **username** (*string*) – назва запису користувача
- **full_name** (*string*) – повне ім'я користувача
- **email** (*string*) – адреса електронної пошти користувача
- **is_superuser** (*boolean*) – чи є користувач надкористувачем
- **is_active** (*boolean*) – чи є користувач активним
- **is_bot** (*boolean*) – чи є користувач ботом (використовується для жетонів рівня проекту)

- **date_joined** (*string*) – дата створення запису користувача

PATCH /api/users/ (str: username) /

Змінює параметри користувача.

Параметри

- **username** (*string*) – Назва облікового запису користувача

Об'єкт відповіді JSON

- **username** (*string*) – назва запису користувача
- **full_name** (*string*) – повне ім'я користувача
- **email** (*string*) – адреса електронної пошти користувача
- **is_superuser** (*boolean*) – чи є користувач надкористувачем
- **is_active** (*boolean*) – чи є користувач активним
- **is_bot** (*boolean*) – чи є користувач ботом (використовується для жетонів рівня проєкту)
- **date_joined** (*string*) – дата створення запису користувача

DELETE /api/users/ (str: username) /

Вилучає усі дані користувача і позначає користувача як неактивного.

Параметри

- **username** (*string*) – Назва облікового запису користувача

POST /api/users/ (str: username) /groups/

Пов'язати групи з користувачем.

Параметри

- **username** (*string*) – Назва облікового запису користувача

Параметри форми

- **string group_id** – Унікальний ідентифікатор групи

DELETE /api/users/ (str: username) /groups/

Нове в версії 4.13.1.

Вилучити користувача з групи.

Параметри

- **username** (*string*) – Назва облікового запису користувача

Параметри форми

- **string group_id** – Унікальний ідентифікатор групи

GET /api/users/ (str: username) /statistics/

Список статистичних даних користувача.

Параметри

- **username** (*string*) – Назва облікового запису користувача

Об'єкт відповіді JSON

- **translated** (*int*) – Кількість перекладів, автором яких є користувач
- **suggested** (*int*) – Кількість пропозицій, автором яких є користувач
- **uploaded** (*int*) – Кількість вивантажень, автором яких є користувач
- **commented** (*int*) – Кількість коментарів, автором яких є користувач

- **languages** (*int*) – Кількість мов, якими може перекладати користувач

GET /api/users/ (*str: username*) /notifications/

Список передплат користувача.

Параметри

- **username** (*string*) – Назва облікового запису користувача

POST /api/users/ (*str: username*) /notifications/

Пов'язати передплату із користувачем.

Параметри

- **username** (*string*) – Назва облікового запису користувача

Надіслати запит щодо об'єкта JSON

- **notification** (*string*) – Назва зареєстрованого сповіщення
- **scope** (*int*) – Область сповіщень з доступних варіантів
- **frequency** (*int*) – Варіанти частоти сповіщення

GET /api/users/ (*str: username*) /notifications/
int: subscription_id/

Отримати передплату, яку пов'язано із користувачем.

Параметри

- **username** (*string*) – Назва облікового запису користувача
- **subscription_id** (*int*) – Ідентифікатор зареєстрованого сповіщення

PUT /api/users/ (*str: username*) /notifications/
int: subscription_id/

Редагувати передплату, яку пов'язано із користувачем.

Параметри

- **username** (*string*) – Назва облікового запису користувача
- **subscription_id** (*int*) – Ідентифікатор зареєстрованого сповіщення

Надіслати запит щодо об'єкта JSON

- **notification** (*string*) – Назва зареєстрованого сповіщення
- **scope** (*int*) – Область сповіщень з доступних варіантів
- **frequency** (*int*) – Варіанти частоти сповіщення

PATCH /api/users/ (*str: username*) /notifications/
int: subscription_id/

Редагувати передплату, яку пов'язано із користувачем.

Параметри

- **username** (*string*) – Назва облікового запису користувача
- **subscription_id** (*int*) – Ідентифікатор зареєстрованого сповіщення

Надіслати запит щодо об'єкта JSON

- **notification** (*string*) – Назва зареєстрованого сповіщення
- **scope** (*int*) – Область сповіщень з доступних варіантів
- **frequency** (*int*) – Варіанти частоти сповіщення

DELETE /api/users/ (*str: username*) /notifications/
int: subscription_id/

Вилучити передплату, яку пов'язано із користувачем.

Параметри

- **username** (*string*) – Назва облікового запису користувача
- **subscription_id** – Назва зареєстрованого сповіщення
- **subscription_id** – int

1.12.4 Групи

Нове в версії 4.0.

GET /api/groups/

Повертає список груп, якщо у вас є права доступу для перегляду групи керування. Якщо прав немає, ви зможете переглянути лише групи, частиною яких є користувач.

Дивись також:

Атрибути об'єкта групи описано у [GET /api/groups/\(int:id\)/](#).

POST /api/groups/

Створює групу.

Параметри

- **name** (*string*) – Назва групи
- **project_selection** (*int*) – Вибір групи проекту за вказаними параметрами
- **language_selection** (*int*) – Група мов, яку вибрано за вказаними параметрами
- **defining_project** (*str*) – посилання на визначальний проект, використовується для *Керування правами доступу на рівні проекту*; див. [GET /api/projects/\(string:project\)/](#)

GET /api/groups/(int: id) /

Повертає відомості щодо групи.

Параметри

- **id** (*int*) – Ідентифікатор групи

Об'єкт відповіді JSON

- **name** (*string*) – назва групи
- **project_selection** (*int*) – ціле число, що відповідає групі проектів
- **language_selection** (*int*) – ціле число, що відповідає групі мов
- **roles** (*array*) – посилання на пов'язані ролі; див. [GET /api/roles/\(int:id\)/](#)
- **projects** (*array*) – посилання на пов'язані проекти; див. [GET /api/projects/\(string:project\)/](#)
- **components** (*array*) – посилання на пов'язані складники; див. [GET /api/components/\(string:project\)/\(string:component\)/](#)
- **componentlists** (*array*) – посилання на пов'язаний список складників; див. [GET /api/component-lists/\(str:slug\)/](#)
- **defining_project** (*str*) – посилання на визначальний проект, використовується для *Керування правами доступу на рівні проекту*; див. [GET /api/projects/\(string:project\)/](#)

Приклад даних JSON:

```
{
  "name": "Guests",
  "defining_project": null,
  "project_selection": 3,
  "language_selection": 1,
  "url": "http://example.com/api/groups/1/",
  "roles": [
    "http://example.com/api/roles/1/",
    "http://example.com/api/roles/2/"
  ],
  "languages": [
    "http://example.com/api/languages/en/",
    "http://example.com/api/languages/cs/"
  ],
  "projects": [
    "http://example.com/api/projects/demo1/",
    "http://example.com/api/projects/demo/"
  ],
  "componentlist": "http://example.com/api/component-lists/new/",
  "components": [
    "http://example.com/api/components/demo/weblate/"
  ]
}
```

PUT /api/groups/(int: id) /

Змінює параметри групи.

Параметри

- **id** (*int*) – Ідентифікатор групи

Об'єкт відповіді JSON

- **name** (*string*) – назва групи
- **project_selection** (*int*) – ціле число, що відповідає групі проєктів
- **language_selection** (*int*) – ціле число, що відповідає групі мов

PATCH /api/groups/(int: id) /

Змінює параметри групи.

Параметри

- **id** (*int*) – Ідентифікатор групи

Об'єкт відповіді JSON

- **name** (*string*) – назва групи
- **project_selection** (*int*) – ціле число, що відповідає групі проєктів
- **language_selection** (*int*) – ціле число, що відповідає групі мов

DELETE /api/groups/(int: id) /

Вилучає групу.

Параметри

- **id** (*int*) – Ідентифікатор групи

POST /api/groups/(int: id)/roles/

Пов'язати ролі з групою.

Параметри

- **id** (*int*) – Ідентифікатор групи

Параметри форми

- **string role_id** – Унікальний ідентифікатор ролі

POST /api/groups/(int: id)/components/

Пов'язати складники з групою.

Параметри

- **id** (int) – Ідентифікатор групи

Параметри форми

- **string component_id** – Унікальний ідентифікатор складника

DELETE /api/groups/(int: id)/components/

int: component_id

Вилучити складник з групи.

Параметри

- **id** (int) – Ідентифікатор групи
- **component_id** (int) – Унікальний ідентифікатор складника

POST /api/groups/(int: id)/projects/

Пов'язати проекти з групою.

Параметри

- **id** (int) – Ідентифікатор групи

Параметри форми

- **string project_id** – Унікальний ідентифікатор проекту

DELETE /api/groups/(int: id)/projects/

int: project_id

Вилучити проєкт з групи.

Параметри

- **id** (int) – Ідентифікатор групи
- **project_id** (int) – Унікальний ідентифікатор проекту

POST /api/groups/(int: id)/languages/

Пов'язати мови з групою.

Параметри

- **id** (int) – Ідентифікатор групи

Параметри форми

- **string language_code** – Унікальний код мови

DELETE /api/groups/(int: id)/languages/

string: language_code

Вилучити мову з групи.

Параметри

- **id** (int) – Ідентифікатор групи
- **language_code** (string) – Унікальний код мови

POST /api/groups/(int: id)/componentlists/

Пов'язати списки складників з групою.

Параметри

- **id** (*int*) – Ідентифікатор групи

Параметри форми

- **string component_list_id** – Унікальний ідентифікатор списку складників

DELETE /api/groups/(int: id)/componentlists/
int: component_list_id

Вилучає список складників з групи.

Параметри

- **id** (*int*) – Ідентифікатор групи
- **component_list_id** (*int*) – Унікальний ідентифікатор списку складників

1.12.5 Ролі

GET /api/roles/

Повертає список усіх ролей, які пов'язано із користувачем. Якщо користувач є надкористувачем, буде повернуто список усіх наявних ролей.

Дивись також:

Атрибути об'єкта ролей документовано у `GET /api/roles/(int:id)/`.

POST /api/roles/

Створює роль.

Параметри

- **name** (*string*) – Назва ролі
- **permissions** (*array*) – Список кодових назв прав доступу

GET /api/roles/(int: id) /

Повертає відомості щодо ролі.

Параметри

- **id** (*int*) – Ідентифікатор ролі

Об'єкт відповіді JSON

- **name** (*string*) – Назва ролі
- **permissions** (*array*) – список кодових назв прав доступу

Приклад даних JSON:

```
{
  "name": "Access repository",
  "permissions": [
    "vcs.access",
    "vcs.view"
  ],
  "url": "http://example.com/api/roles/1/",
}
```

PUT /api/roles/(int: id) /

Змінює параметри ролі.

Параметри

- **id** (*int*) – Ідентифікатор ролі

Об'єкт відповіді JSON

- **name** (*string*) – Назва ролі
- **permissions** (*array*) – список кодових назв прав доступу

PATCH /api/roles/ (int: id) /

Змінює параметри ролі.

Параметри

- **id** (*int*) – Ідентифікатор ролі

Об'єкт відповіді JSON

- **name** (*string*) – Назва ролі
- **permissions** (*array*) – список кодових назв прав доступу

DELETE /api/roles/ (int: id) /

Вилучає роль.

Параметри

- **id** (*int*) – Ідентифікатор ролі

1.12.6 Мови

GET /api/languages/

Повертає список усіх мов.

Дивись також:

Атрибути об'єкта ролі описано у [GET /api/languages/ \(string: language\) /](#).

POST /api/languages/

Створює мову.

Параметри

- **code** (*string*) – Назва мови
- **name** (*string*) – Назва мови
- **direction** (*string*) – Напрямок тексту
- **population** (*int*) – Кількість мовців
- **plural** (*object*) – Кількість і формула форм множини мови

GET /api/languages/ (string: language) /

Повертає відомості щодо мови.

Параметри

- **language** (*string*) – Код мови

Об'єкт відповіді JSON

- **code** (*string*) – Код мови
- **direction** (*string*) – Напрямок тексту
- **plural** (*object*) – Об'єкт даних щодо форм множини мови
- **aliases** (*array*) – Масив альтернативних назв мови

Надіслати запит щодо об'єкта JSON

- **population** (*int*) – Кількість мовців

Приклад даних JSON:

```
{
  "code": "en",
  "direction": "ltr",
  "name": "English",
  "population": 159034349015,
  "plural": {
    "id": 75,
    "source": 0,
    "number": 2,
    "formula": "n != 1",
    "type": 1
  },
  "aliases": [
    "english",
    "en_en",
    "base",
    "source",
    "eng"
  ],
  "url": "http://example.com/api/languages/en/",
  "web_url": "http://example.com/languages/en/",
  "statistics_url": "http://example.com/api/languages/en/statistics/"
}
```

PUT /api/languages/(string: language) /

Зміни у параметрах мови.

Параметри

- **language** (string) – Код мови

Надіслати запит щодо об'єкта JSON

- **name** (string) – Назва мови
- **direction** (string) – Напрямок тексту
- **population** (int) – Кількість мовців
- **plural** (object) – Подробиці щодо форм множини мови

PATCH /api/languages/(string: language) /

Зміни у параметрах мови.

Параметри

- **language** (string) – Код мови

Надіслати запит щодо об'єкта JSON

- **name** (string) – Назва мови
- **direction** (string) – Напрямок тексту
- **population** (int) – Кількість мовців
- **plural** (object) – Подробиці щодо форм множини мови

DELETE /api/languages/(string: language) /

Вилучає мову.

Параметри

- **language** (string) – Код мови

GET /api/languages/(string: language)/statistics/

Повертає статистику щодо мови.

Параметри

- **language** (*string*) – Код мови

Об'єкт відповіді JSON

- **total** (*int*) – загальна кількість рядків
- **total_words** (*int*) – загальна кількість слів
- **last_change** (*timestamp*) – останні зміни у мові
- **recent_changes** (*int*) – загальна кількість змін
- **translated** (*int*) – кількість перекладених рядків
- **translated_percent** (*float*) – частка перекладених рядків
- **translated_words** (*int*) – кількість перекладених слів
- **translated_words_percent** (*int*) – частка перекладених слів
- **translated_chars** (*int*) – кількість перекладених символів
- **translated_chars_percent** (*int*) – частка перекладених символів
- **total_chars** (*int*) – загальна кількість символів
- **fuzzy** (*int*) – кількість неточно перекладених (позначених для редагування) рядків
- **fuzzy_percent** (*int*) – частка неточно перекладених (позначених для редагування) рядків
- **failing** (*int*) – кількість рядків, які не пройшли перевірки
- **failing** – відсоток рядків, які не пройшли перевірки

1.12.7 Проєкти

GET /api/projects/

Повертає список усіх проєктів.

Дивись також:

Атрибути об'єкта проєкту описано у [GET /api/projects/\(string:project\)/](#).

POST /api/projects/

Нове в версії 3.9.

Створює проєкт.

Параметри

- **name** (*string*) – Назва проєкту
- **slug** (*string*) – Ідентифікатор проєкту
- **web** (*string*) – Вебсайт проєкту

GET /api/projects/(string: project) /

Повертає відомості щодо проєкту.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту

Об'єкт відповіді JSON

- **name** (*string*) – назва проєкту
- **slug** (*string*) – ідентифікатор проєкту

- **web** (*string*) – вебсайт проекту
- **components_list_url** (*string*) – адреса списку складників, див. `GET /api/projects/(string:project)/components/`
- **repository_url** (*string*) – адреса стану репозиторію, див. `GET /api/projects/(string:project)/repository/`
- **changes_list_url** (*string*) – адреса списку змін, див. `GET /api/projects/(string:project)/changes/`
- **translation_review** (*boolean*) – Увімкнути рецензії
- **source_review** (*boolean*) – Увімкнути рецензування початкових даних
- **set_language_team** (*boolean*) – Установлення заголовка «Language-Team»
- **enable_hooks** (*boolean*) – Увімкнути обробники
- **instructions** (*string*) – Вказівки щодо перекладу
- **language_aliases** (*string*) – Альтернативні назви мови

Приклад даних JSON:

```
{
  "name": "Hello",
  "slug": "hello",
  "url": "http://example.com/api/projects/hello/",
  "web": "https://weblate.org/",
  "web_url": "http://example.com/projects/hello/"
}
```

PATCH `/api/projects/(string: project) /`

Нове в версії 4.3.

Редагувати проект розміщенням запиту **PATCH**.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

PUT `/api/projects/(string: project) /`

Нове в версії 4.3.

Редагувати проект розміщенням запиту **PUT**.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту

DELETE `/api/projects/(string: project) /`

Нове в версії 3.9.

Вилучає проект.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту

GET `/api/projects/(string: project) /changes/`

Повертає список змін у проєкті. Це, по суті, `GET /api/changes/` області видимості проєкту, які приймають ті самі параметри.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту

Об'єкт відповіді JSON

- **results** (*array*) – список об'єктів складника, див. `GET /api/changes/(int:id)/`

GET /api/projects/(string: project)/repository/

Повертає відомості про стан репозиторію системи керування версіями. Ця кінцева точка містить лише загальне резюме для всіх репозиторіїв проекту. Щоб отримати докладні дані про стан скористайтесь `GET /api/components/(string:project)/(string:component)/repository/`.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту

Об'єкт відповіді JSON

- **needs_commit** (*boolean*) – визначає, чи є якісь зміни у черзі внесків
- **needs_merge** (*boolean*) – визначає, чи є якісь зміни у основному репозиторії для злиття
- **needs_push** (*boolean*) – визначає, чи є якісь локальні зміни для запису

Приклад даних JSON:

```
{
  "needs_commit": true,
  "needs_merge": false,
  "needs_push": true
}
```

POST /api/projects/(string: project)/repository/

Виконує вказану дію над репозиторієм системи керування версіями.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту

Надіслати запит щодо об'єкта JSON

- **operation** (*string*) – Дія, яку слід виконати. Одне з таких значень: push, pull, commit, reset, cleanup, file-sync

Об'єкт відповіді JSON

- **result** (*boolean*) – результат дії

Приклад для CURL:

```
curl \
  -d operation=pull \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/repository/
```

Приклад запиту JSON:

```
POST /api/projects/hello/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{"operation": "pull"}
```

Приклад відповіді JSON:


```

HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"result":true}

```

GET `/api/projects/(string: project)/components/`

Повертає список складників перекладу у вказаному проєкті.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту

Об'єкт відповіді JSON

- **results** (*array*) – масив об'єктів складника, див. `GET /api/components/(string:project)/(string:component)/`

POST `/api/projects/(string: project)/components/`

Нове в версії 3.9.

Змінено в версії 4.3: У новій версії параметри `zipfile` і `docfile` приймаються меншою кількістю складників системи керування версіями, див. *Локальні файли*.

Змінено в версії 4.6: У нових версіях клоновані репозиторію автоматично спільно використовуються у межах проєкту з використанням *Внутрішні адреси Weblate*. Скористайтесь `disable_autoshare`, щоб вимкнути таку поведінку.

Створює складники перекладу у вказаному проєкті.

Підказка: Використовуйте *Внутрішні адреси Weblate* під час створення кількох складників з одного репозиторію VCS.

Примітка: Більша частина дій зі створення складника виконується у фоновому режимі. Ознайомтеся із документацією до атрибута `task_url` створеного складника і стежте за поступом за його допомогою.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту

Параметри форми

- **file zipfile** – Файл ZIP, який слід вивантажити до Weblate для ініціалізації перекладів
- **file docfile** – Документ для перекладу
- **boolean disable_autoshare** – Вимикає автоматичне спільне використання репозиторіїв за допомогою *Внутрішні адреси Weblate*.

Надіслати запит щодо об'єкта JSON

- **object** – Параметри складника; див. `GET /api/components/(string:project)/(string:component)/`

Об'єкт відповіді JSON

- **result** (*object*) – Створений об'єкт складника; див. `GET /api/components/(string:project)/(string:component)/`

JSON не можна використовувати при вивантаженні файлів з використанням параметрів `zipfile` і `docfile`. Дані слід вивантажувати як `multipart/form-data`.

Приклад запиту форми CURL:

```
curl \
  --form docfile=@strings.html \
  --form name=Weblate \
  --form slug=weblate \
  --form file_format=html \
  --form new_lang=add \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

Приклад запиту JSON CURL:

```
curl \
  --data-binary '{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "name": "Weblate",
    "slug": "weblate",
    "repo": "https://github.com/WeblateOrg/hello.git",
    "template": "",
    "new_base": "po/hello.pot",
    "vcs": "git"
  }' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

Запит JSON для створення складника з Git:

```
POST /api/projects/hello/components/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "name": "Weblate",
  "slug": "weblate",
  "repo": "https://github.com/WeblateOrg/hello.git",
  "template": "",
  "new_base": "po/hello.pot",
  "vcs": "git"
}
```

Запит JSON для створення складника на основі іншого складника:

```
POST /api/projects/hello/components/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```

Authorization: Token TOKEN
Content-Length: 20

{
  "file_format": "po",
  "filemask": "po/*.po",
  "name": "Weblate",
  "slug": "weblate",
  "repo": "weblate://weblate/hello",
  "template": "",
  "new_base": "po/hello.pot",
  "vcs": "git"
}

```

Приклад відповіді JSON:

```

HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "population": 159034349015,
      "name": "English",
      "url": "http://example.com/api/languages/en/",
      "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
  },
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "url": "http://example.com/api/components/hello/weblate/",
  "vcs": "git",
  "web_url": "http://example.com/projects/hello/weblate/"
}

```

GET /api/projects/(string: project)/languages/

Повертає поділену на сторінки статистику для всіх мов у проєкті.

Нове в версії 3.8.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту

Об'єкт відповіді JSON

- **results** (*array*) – масив об'єктів статистики перекладу
- **language** (*string*) – назва мови
- **code** (*string*) – код мови
- **total** (*int*) – загальна кількість рядків
- **translated** (*int*) – кількість перекладених рядків
- **translated_percent** (*float*) – частка перекладених рядків
- **total_words** (*int*) – загальна кількість слів
- **translated_words** (*int*) – кількість перекладених слів
- **words_percent** (*float*) – частка перекладених слів

GET /api/projects/(string: project)/statistics/

Повертає статистику проекту.

Нове в версії 3.8.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту

Об'єкт відповіді JSON

- **total** (*int*) – загальна кількість рядків
- **translated** (*int*) – кількість перекладених рядків
- **translated_percent** (*float*) – частка перекладених рядків
- **total_words** (*int*) – загальна кількість слів
- **translated_words** (*int*) – кількість перекладених слів
- **words_percent** (*float*) – частка перекладених слів

1.12.8 Складники

Підказка: Для створення компонентів скористайтеся `POST /api/projects/(string:project)/components/`.

GET /api/components/

Повертає список складників перекладу.

Дивись також:

Атрибути об'єкта складника документовано у `GET /api/components/(string:project)/(string:component)/`.

GET /api/components/(string: project) /
string: component/

Повертає дані щодо складника перекладу.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Об'єкт відповіді JSON

- **project** (object) – проект перекладу, див. `GET /api/projects/(string:project)/`
- **name** (string) – Назва складника
- **slug** (string) – Ідентифікатор складника
- **vcs** (string) – Система керування версіями
- **repo** (string) – Репозиторій з джерелами
- **git_export** (string) – Експортований URL репозиторію
- **branch** (string) – Гілка репозиторію
- **push_branch** (string) – Гілка для запису
- **filemask** (string) – Маска файлів
- **template** (string) – Основний файл для одномовного перекладу
- **edit_template** (string) – Змінити основний файл
- **intermediate** (string) – Проміжний мовний файл
- **new_base** (string) – Шаблон для нових перекладів
- **file_format** (string) – Формат файлу
- **license** (string) – Ліцензія перекладу
- **agreement** (string) – Угода із учасником
- **new_lang** (string) – Додавання нового перекладу
- **language_code_style** (string) – Стиль коду мови
- **source_language** (object) – об'єкт початкової мови, див. `GET /api/languages/(string:language)/`
- **push** (string) – Адреса для записування до репозиторію
- **check_flags** (string) – Прапорці перекладу
- **priority** (string) – Пріоритет
- **enforced_checks** (string) – Примусові перевірки
- **restricted** (string) – Обмежений доступ
- **repoweb** (string) – Переглядач репозиторію
- **report_source_bugs** (string) – Адреса для повідомлень про помилки у рядках
- **merge_style** (string) – Стиль злиття
- **commit_message** (string) – Повідомлення щодо внеску, додавання, вилучення, повідомлення додатків та запитів щодо об'єднання
- **add_message** (string) – Повідомлення щодо внеску, додавання, вилучення, повідомлення додатків та запитів щодо об'єднання
- **delete_message** (string) – Повідомлення щодо внеску, додавання, вилучення, повідомлення додатків та запитів щодо об'єднання
- **merge_message** (string) – Повідомлення щодо внеску, додавання, вилучення, повідомлення додатків та запитів щодо об'єднання
- **addon_message** (string) – Повідомлення щодо внеску, додавання, вилучення, повідомлення додатків та запитів щодо об'єднання
- **pull_message** (string) – Повідомлення щодо внеску, додавання, вилучення, повідомлення додатків та запитів щодо об'єднання

- **allow_translation_propagation** (*string*) – Дозволити поширення перекладу
- **enable_suggestions** (*string*) – Пропонувати переклад
- **suggestion_voting** (*string*) – Голосування за пропозицію
- **suggestion_autoaccept** (*string*) – Автоматичне прийняття пропозицій
- **push_on_commit** (*string*) – Відправляти при поданні
- **commit_pending_age** (*string*) – Вік змін для подання
- **auto_lock_error** (*string*) – Блокувати при помилці
- **language_regex** (*string*) – Фільтр мов
- **variant_regex** (*string*) – Формальний вираз варіантів
- **repository_url** (*string*) – Адреса стану репозиторію, див. `GET /api/components/(string:project)/(string:component)/repository/`
- **translations_url** (*string*) – Адреса списку перекладів, див. `GET /api/components/(string:project)/(string:component)/translations/`
- **lock_url** (*string*) – Адреса для стану блокування, див. `GET /api/components/(string:project)/(string:component)/lock/`
- **changes_list_url** (*string*) – Адреса списку змін; див. `GET /api/components/(string:project)/(string:component)/changes/`
- **task_url** (*string*) – Адреса фонового завдання (якщо таке виконується); див. `GET /api/tasks/(str:uuid)/`

Приклад даних JSON:

```
{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "population": 159034349015,
      "name": "English",
      "url": "http://example.com/api/languages/en/",
      "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
  },
  "source_language": {
    "code": "en",
    "direction": "ltr",
    "population": 159034349015,
    "name": "English",
    "url": "http://example.com/api/languages/en/",

```

(continues on next page)

(continued from previous page)

```

    "web_url": "http://example.com/languages/en/"
  },
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "url": "http://example.com/api/components/hello/weblate/",
  "vcs": "git",
  "web_url": "http://example.com/projects/hello/weblate/"
}

```

PATCH `/api/components/(string: project) /`
string: `component/`

Редагувати складник розміщенням запиту **PATCH**.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту
- **component** (*string*) – Ідентифікатор адрес складника
- **source_language** (*string*) – Код початкової мови проєкту (необов'язковий)

Надіслати запит щодо об'єкта JSON

- **name** (*string*) – назва складника
- **slug** (*string*) – ідентифікатор складника
- **repo** (*string*) – адреса репозиторію системи керування версіями

Приклад для CURL:

```

curl \
  --data-binary '{"name": "new name"}' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  PATCH http://example.com/api/projects/hello/components/

```

Приклад запиту JSON:

```

PATCH /api/projects/hello/components/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{
  "name": "new name"
}

```

Приклад відповіді JSON:

```

HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{

```

(continues on next page)

(continued from previous page)

```

"branch": "main",
"file_format": "po",
"filemask": "po/*.po",
"git_export": "",
"license": "",
"license_url": "",
"name": "new name",
"slug": "weblate",
"project": {
  "name": "Hello",
  "slug": "hello",
  "source_language": {
    "code": "en",
    "direction": "ltr",
    "population": 159034349015,
    "name": "English",
    "url": "http://example.com/api/languages/en/",
    "web_url": "http://example.com/languages/en/"
  },
  "url": "http://example.com/api/projects/hello/",
  "web": "https://weblate.org/",
  "web_url": "http://example.com/projects/hello/"
},
"repo": "file:///home/nijel/work/weblate-hello",
"template": "",
"new_base": "",
"url": "http://example.com/api/components/hello/weblate/",
"vcs": "git",
"web_url": "http://example.com/projects/hello/weblate/"
}

```

PUT `/api/components/(string: project) /`
string: `component/`

Редагувати складник розміщенням запиту **PUT**.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Надіслати запит щодо об'єкта JSON

- **branch** (*string*) – гілка репозиторію системи керування версіями
- **file_format** (*string*) – формат файлів перекладів
- **filemask** (*string*) – маска файлів перекладів у репозиторії
- **name** (*string*) – назва складника
- **slug** (*string*) – ідентифікатор складника
- **repo** (*string*) – адреса репозиторію системи керування версіями
- **template** (*string*) – базовий файл для одномовних перекладів
- **new_base** (*string*) – базовий файл для додавання нових перекладів
- **vcs** (*string*) – система керування версіями

DELETE `/api/components/(string: project) /`
string: `component/`

Нове в версії 3.9.

Вилучає складник.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

GET /api/components/ (**string:** *project*) /
string: *component/changes/*

Повертає список змін у складнику. Це, по суті, *GET /api/changes/* області видимості складника, які приймають ті самі параметри.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Об'єкт відповіді JSON

- **results** (*array*) – список об'єктів складника, див. *GET /api/changes/ (int:id)/*

GET /api/components/ (**string:** *project*) /
string: *component/file/*

Нове в версії 4.9.

Отримує усі доступні переклади, які пов'язано зі складником, як архів у вказаному форматі.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Параметри запиту

- **format** (*string*) – Формат архіву, яким слід скористатися. Якщо не вказано, типовим є *zip*. Підтримувані формати: *zip*

GET /api/components/ (**string:** *project*) /
string: *component/screenshots/*

Повертає список знімків вікон складника.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Об'єкт відповіді JSON

- **results** (*array*) – масив знімків вікон складника; див. *GET /api/screenshots/ (int:id)/*

GET /api/components/ (**string:** *project*) /
string: *component/lock/*

Повертає стан блокування складника.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Об'єкт відповіді JSON

- **locked** (*boolean*) – визначає, чи заблоковано складник від оновлень

Приклад даних JSON:

```
{
  "locked": false
}
```

POST `/api/components/(string: project) /`
string: `component/lock/`

Установлює стан блокування складника.

Відповідь є тією самою, що і `GET /api/components/(string:project)/(string:component)/lock/`.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Надіслати запит щодо об'єкта JSON

- **lock** – Булеве значення, яке визначає, замкнено репозиторій чи ні.

Приклад для CURL:

```
curl \
  -d lock=true \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/components/hello/weblate/repository/
```

Приклад запиту JSON:

```
POST /api/components/hello/weblate/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{"lock": true}
```

Приклад відповіді JSON:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"locked":true}
```

GET `/api/components/(string: project) /`
string: `component/repository/`

Повертає дані щодо стану репозиторію системи керування версіями.

Відповідь є тією самою, що і для `GET /api/projects/(string:project)/repository/`.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Об'єкт відповіді JSON

- **needs_commit** (*boolean*) – визначає, чи є якісь зміни у черзі внесків
- **needs_merge** (*boolean*) – визначає, чи є якісь зміни у основному репозиторії для злиття
- **needs_push** (*boolean*) – визначає, чи є якісь локальні зміни для запису
- **remote_commit** (*string*) – Дані віддаленого внеску
- **status** (*string*) – Стан репозиторію системи керування версіями, який повідомляється системою керування версіями
- **merge_failure** – Текст, що описує помилку злиття або pull, якщо помилки немає

POST /api/components/(*string: project*) /
string: component/repository/

Виконує вказану дію над репозиторієм системи керування версіями.

Див. документацію у **POST** /api/projects/(*string:project*)/repository/.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту
- **component** (*string*) – Ідентифікатор адрес складника

Надіслати запит щодо об'єкта JSON

- **operation** (*string*) – Дія, яку слід виконати. Одне з таких значень: push, pull, commit, reset, cleanup

Об'єкт відповіді JSON

- **result** (*boolean*) – результат дії

Приклад для CURL:

```
curl \
  -d operation=pull \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/components/hello/weblate/repository/
```

Приклад запиту JSON:

```
POST /api/components/hello/weblate/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{"operation":"pull"}
```

Приклад відповіді JSON:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"result":true}
```

GET /api/components/(*string: project*) /
string: component/monolingual_base/

Завантажує базовий файл для одномовних перекладів.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

GET /api/components/ (**string**: *project*) /
string: *component/new_template/*

Завантажує файл шаблону для нових перекладів.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

GET /api/components/ (**string**: *project*) /
string: *component/translations/*

Повертає список об'єктів перекладу у вказаному складнику.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Об'єкт відповіді JSON

- **results** (*array*) – масив об'єктів перекладу, див. *GET /api/translations/ (string:project) / (string:component) / (string:language) /*

POST /api/components/ (**string**: *project*) /
string: *component/translations/*

Створює переклад у вказаному складнику.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Надіслати запит щодо об'єкта JSON

- **language_code** (*string*) – код мови перекладу; див. *GET /api/languages/ (string:language) /*

Об'єкт відповіді JSON

- **result** (*object*) – створено об'єкт перекладу

Приклад для CURL:

```
curl \
  -d language_code=cs \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

Приклад запиту JSON:

```
POST /api/projects/hello/components/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20
```

(continues on next page)

(continued from previous page)

```
{"language_code": "cs"}
```

Приклад відповіді JSON:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
  "failing_checks": 0,
  "failing_checks_percent": 0,
  "failing_checks_words": 0,
  "filename": "po/cs.po",
  "fuzzy": 0,
  "fuzzy_percent": 0.0,
  "fuzzy_words": 0,
  "have_comment": 0,
  "have_suggestion": 0,
  "is_template": false,
  "is_source": false,
  "language": {
    "code": "cs",
    "direction": "ltr",
    "population": 1303174280,
    "name": "Czech",
    "url": "http://example.com/api/languages/cs/",
    "web_url": "http://example.com/languages/cs/"
  },
  "language_code": "cs",
  "id": 125,
  "last_author": null,
  "last_change": null,
  "share_url": "http://example.com/engage/hello/cs/",
  "total": 4,
  "total_words": 15,
  "translate_url": "http://example.com/translate/hello/weblate/cs/",
  "translated": 0,
  "translated_percent": 0.0,
  "translated_words": 0,
  "url": "http://example.com/api/translations/hello/weblate/cs/",
  "web_url": "http://example.com/projects/hello/weblate/cs/"
}
```

GET /api/components/(string: project) /
string: component/statistics/

Повертає поділену на сторінки статистику для всіх перекладів складника.

Нове в версії 2.7.

Параметри

- **project** (string) – Ідентифікатор адреси проекту
- **component** (string) – Ідентифікатор адрес складника

Об'єкт відповіді JSON

- **results** (*array*) – масив об'єктів статистики перекладу, див. `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/`

GET `/api/components/(string: project) /string: component/links/`

Повертає проєкт, який пов'язано зі складником.

Нове в версії 4.5.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту
- **component** (*string*) – Ідентифікатор адрес складника

Об'єкт відповіді JSON

- **projects** (*array*) – пов'язані проєкти; див. `GET /api/projects/(string:project)/`

POST `/api/components/(string: project) /string: component/links/`

Пов'язати проєкт зі складником.

Нове в версії 4.5.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту
- **component** (*string*) – Ідентифікатор адрес складника

Параметри форми

- **string project_slug** – Ідентифікатор проєкту

DELETE `/api/components/(string: project) /string: component/links/string: project_slug/`

Вилучити прив'язку проєкту до складника.

Нове в версії 4.5.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту
- **component** (*string*) – Ідентифікатор адрес складника
- **project_slug** (*string*) – Ідентифікатор проєкту, який слід вилучити

1.12.9 Переклади

GET `/api/translations/`

Повертає список перекладів.

Дивись також:

Атрибути об'єкта перекладу документовано у `GET /api/translations/(string:project)/(string:component)/(string:language)/`.

GET `/api/translations/(string: project) /string: component/string: language/`

Повертає відомості щодо перекладу.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту

- **component** (*string*) – Ідентифікатор адрес складника
- **language** (*string*) – Код мови перекладу

Об'єкт відповіді JSON

- **component** (*object*) – об'єкт складника; див. `GET /api/components/(string:project)/(string:component)/`
- **failing_checks** (*int*) – кількість рядків, які не пройшли перевірки
- **failing_checks_percent** (*float*) – відсоток рядків, які не пройшли перевірки
- **failing_checks_words** (*int*) – кількість слів у рядках, які не пройшли перевірки
- **filename** (*string*) – назва файла перекладу
- **fuzzy** (*int*) – кількість неточно перекладених (позначених для редагування) рядків
- **fuzzy_percent** (*float*) – частка неточно перекладених (позначених для редагування) рядків
- **fuzzy_words** (*int*) – кількість слів у неточно перекладених (позначених для редагування) рядках
- **have_comment** (*int*) – кількість рядків із коментарями
- **have_suggestion** (*int*) – кількість рядків із пропозиціями
- **is_template** (*boolean*) – чи переклад має одномовну основу
- **language** (*object*) – об'єкт початкової мови, див. `GET /api/languages/(string:language)/`
- **language_code** (*string*) – код мови, використаний у репозиторії; може відрізнятися від коду мови у об'єкті мови
- **last_author** (*string*) – ім'я останнього автора
- **last_change** (*timestamp*) – часова позначка останньої зміни
- **revision** (*string*) – хеш модифікації для файла
- **share_url** (*string*) – Адреса для спільного ресурсу, що веде до сторінки участі
- **total** (*int*) – загальна кількість рядків
- **total_words** (*int*) – загальна кількість слів
- **translate_url** (*string*) – Адреса для перекладу
- **translated** (*int*) – кількість перекладених рядків
- **translated_percent** (*float*) – частка перекладених рядків
- **translated_words** (*int*) – кількість перекладених слів
- **repository_url** (*string*) – Адреса стану репозиторію, див. `GET /api/translations/(string:project)/(string:component)/(string:language)/repository/`
- **file_url** (*string*) – Адреса об'єкта файла, див. `GET /api/translations/(string:project)/(string:component)/(string:language)/file/`
- **changes_list_url** (*string*) – Адреса списку змін, див. `GET /api/translations/(string:project)/(string:component)/(string:language)/changes/`

- `units_list_url (string)` – Адреса списку рядків, див. `GET /api/translations/(string:project)/(string:component)/(string:language)/units/`

Приклад даних JSON:

```
{
  "component": {
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "new_base": "",
    "project": {
      "name": "Hello",
      "slug": "hello",
      "source_language": {
        "code": "en",
        "direction": "ltr",
        "population": 159034349015,
        "name": "English",
        "url": "http://example.com/api/languages/en/",
        "web_url": "http://example.com/languages/en/"
      },
      "url": "http://example.com/api/projects/hello/",
      "web": "https://weblate.org/",
      "web_url": "http://example.com/projects/hello/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "slug": "weblate",
    "template": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
  },
  "failing_checks": 3,
  "failing_checks_percent": 75.0,
  "failing_checks_words": 11,
  "filename": "po/cs.po",
  "fuzzy": 0,
  "fuzzy_percent": 0.0,
  "fuzzy_words": 0,
  "have_comment": 0,
  "have_suggestion": 0,
  "is_template": false,
  "language": {
    "code": "cs",
    "direction": "ltr",
    "population": 1303174280,
    "name": "Czech",
    "url": "http://example.com/api/languages/cs/",
    "web_url": "http://example.com/languages/cs/"
  },
  "language_code": "cs",
  "last_author": "Weblate Admin",
  "last_change": "2016-03-07T10:20:05.499",
  "revision": "7ddfafe6daaf57fc8654cc852ea6be212b015792",
  "share_url": "http://example.com/engage/hello/cs/",
  "total": 4,
  "total_words": 15,
}
```

(continues on next page)

(continued from previous page)

```

"translate_url": "http://example.com/translate/hello/weblate/cs/",
"translated": 4,
"translated_percent": 100.0,
"translated_words": 15,
"url": "http://example.com/api/translations/hello/weblate/cs/",
"web_url": "http://example.com/projects/hello/weblate/cs/"
}

```

DELETE /api/translations/(string: project) /
 string: component/string: language/

Нове в версії 3.9.

Вилучає переклад.

Параметри

- **project** (string) – Ідентифікатор адреси проекту
- **component** (string) – Ідентифікатор адрес складника
- **language** (string) – Код мови перекладу

GET /api/translations/(string: project) /
 string: component/string: language/changes/

Повертає список змін у перекладі. Це, по суті, *GET* /api/changes/ області видимості перекладу, які приймають ті самі параметри.

Параметри

- **project** (string) – Ідентифікатор адреси проекту
- **component** (string) – Ідентифікатор адрес складника
- **language** (string) – Код мови перекладу

Об'єкт відповіді JSON

- **results** (array) – список об'єктів складника, див. *GET* /api/changes/(int:id)/

GET /api/translations/(string: project) /
 string: component/string: language/units/

Повертає список модулів перекладу.

Параметри

- **project** (string) – Ідентифікатор адреси проекту
- **component** (string) – Ідентифікатор адрес складника
- **language** (string) – Код мови перекладу
- **q** (string) – Рядок пошукового запиту *Пошук* (необов'язково)

Об'єкт відповіді JSON

- **results** (array) – масив об'єктів складників; див. *GET* /api/units/(int:id)/

POST /api/translations/(string: project) /
 string: component/string: language/units/

Додати новий модуль.

Параметри

- **project** (string) – Ідентифікатор адреси проекту
- **component** (string) – Ідентифікатор адрес складника

- **language** (*string*) – Код мови перекладу

Надіслати запит щодо об'єкта JSON

- **key** (*string*) – Назва модуля перекладу (використовують як ключ або контекст)
- **value** (*array*) – Початкові рядки (один рядок, якщо не створено форм множини)
- **state** (*int*) – Стан рядка, див. *GET /api/units/(int:id)/*

Об'єкт відповіді JSON

- **unit** (*object*) – новостворений модуль; див. *GET /api/units/(int:id)/*

Дивись також:

Керування рядками, *adding-new-strings*

POST /api/translations/(string: project) /
string: component/string: language/autotranslate/
Перемкнути автоматичний переклад.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника
- **language** (*string*) – Код мови перекладу

Надіслати запит щодо об'єкта JSON

- **mode** (*string*) – Режим автоматичного перекладу
- **filter_type** (*string*) – Автоматичний тип фільтрування перекладів
- **auto_source** (*string*) – Джерело автоматичного перекладу — *mt* або *others*
- **component** (*string*) – Щоб отримати доступ до додаткових складників, увімкніть спільний внесок у пам'ять перекладів для проекту.
- **engines** (*array*) – Рушії комп'ютерного перекладу
- **threshold** (*string*) – Поріг оцінки

GET /api/translations/(string: project) /
string: component/string: language/file/

Отримати поточний файл перекладу у форматі, в якому він зберігається у системі керування версіями, (без параметра *format*) або у перетвореному форматі (див *Отримання перекладів*).

Примітка: Ця кінцева точка програмного інтерфейсу використовує іншу логіку для виведення даних, якщо порівнювати із рештою програмного інтерфейсу, оскільки працює із цілим файлом, а не із даними цього файла. Набір прийнятних значень параметра `format` є іншим. Без цього параметра ви отримаєте файл перекладу у тому форматі, у якому він зберігається у системі керування версіями.

Параметри запиту

- **format** – Формат файлів, яким слід скористатися; якщо не вказано, перетворення форматів не виконуватиметься; підтримувані формати файлів: *po*, *mo*, *xliff*, *xliff11*, *tbx*, *tmx*, *csv*, *xlsx*, *json*, *aresource*, *strings*
- **q** (*string*) – Фільтрувати отримані рядки, див. *search*, можна застосувати, лише якщо перетворення відбувається на місці (вказано *format*).

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

- **language** (*string*) – Код мови перекладу

POST /api/translations/(string: project) /
string: component/string: language/file/

Вивантажити новий файл із перекладами.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника
- **language** (*string*) – Код мови перекладу

Параметри форми

- **string conflicts** – Спосіб обробки конфліктів (ignore, replace-translated або replace-approved)
- **file file** – Вивантажений файл
- **string email** – Адреса ел. пошти автора
- **string author** – Ім'я автора
- **string method** – Спосіб вивантаження (translate, approve, suggest, fuzzy, replace, source, add), див. *Способи імпортування*
- **string fuzzy** – Обробка неточних (позначених для редагування) перекладів (empty, process, approve)

Приклад для CURL:

```
curl -X POST \
  -F file=@strings.xml \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/translations/hello/android/cs/file/
```

GET /api/translations/(string: project) /
string: component/string: language/repository/

Повертає дані щодо стану репозиторію системи керування версіями.

Відповідь така сама, як для *GET* /api/components/(string:project) /
(string:component)/repository/.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника
- **language** (*string*) – Код мови перекладу

POST /api/translations/(string: project) /
string: component/string: language/repository/

Виконує вказану дію над репозиторієм системи керування версіями.

Див. документацію у *POST* /api/projects/(string:project)/repository/.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника
- **language** (*string*) – Код мови перекладу

Надіслати запит щодо об'єкта JSON

- **operation** (*string*) – Дія, яку слід виконати. Одне з таких значень: `push`, `pull`, `commit`, `reset`, `cleanup`

Об'єкт відповіді JSON

- **result** (*boolean*) – результат дії

GET `/api/translations/` (**string**: *project*) /
string: *component* / **string**: *language* / **statistics** /

Повертає докладну статистику щодо перекладу.

Нове в версії 2.7.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника
- **language** (*string*) – Код мови перекладу

Об'єкт відповіді JSON

- **code** (*string*) – код мови
- **failing** (*int*) – кількість рядків, які не пройшли перевірки
- **failing_percent** (*float*) – частка рядків, які не пройшли перевірки
- **fuzzy** (*int*) – кількість неточно перекладених (позначених для редагування) рядків
- **fuzzy_percent** (*float*) – частка неточно перекладених (позначених для редагування) рядків
- **total_words** (*int*) – загальна кількість слів
- **translated_words** (*int*) – кількість перекладених слів
- **last_author** (*string*) – ім'я останнього автора
- **last_change** (*timestamp*) – дата останньої зміни
- **name** (*string*) – назва мови
- **total** (*int*) – загальна кількість рядків
- **translated** (*int*) – кількість перекладених рядків
- **translated_percent** (*float*) – частка перекладених рядків
- **url** (*string*) – адреса для доступу до перекладу (адреса запрошення)
- **url_translate** (*string*) – адреса доступу до перекладу (справжня адреса перекладу)

1.12.10 Пам'ять

Нове в версії 4.14.

GET `/api/memory/`

Повертає список результатів з пам'яті.

DELETE `/api/memory/` (**int**: *memory_object_id*) /

Вилучає об'єкт пам'яті

Параметри

- **memory_object_id** – Ідентифікатор об'єкта пам'яті

1.12.11 Модулі

Модуль — окрема одиниця перекладу, яка поєднує початковий рядок із відповідним рядком перекладу, а також містить деякі пов'язані із ними метадані. Термін походить з [Translate Toolkit](#) та XLIFF.

Нове в версії 2.10.

GET `/api/units/`

Повертає список модулів перекладу.

Параметри

- **q** (*string*) – Рядок пошукового запиту *Пошук* (необов'язково)

Дивись також:

Атрибути об'єкта модуля документовано у `GET /api/units/(int:id)/`.

GET `/api/units/(int: id) /`

Змінено в версії 4.3: У нових версіях `target` і `source` є масивами для належного зберігання форм множини.

Повертає відомості щодо модуля перекладу.

Параметри

- **id** (*int*) – Ід. модуля

Об'єкт відповіді JSON

- **translation** (*string*) – Адреса пов'язаного об'єкта перекладу
- **source** (*array*) – початковий рядок
- **previous_source** (*string*) – попередній початковий рядок, який використано для встановлення приблизної відповідності
- **target** (*array*) – рядок перекладу
- **id_hash** (*string*) – унікальний ідентифікатор модуля
- **content_hash** (*string*) – унікальний ідентифікатор початкового рядка
- **location** (*string*) – розташування модуля у початковому рядку
- **context** (*string*) – контекст модуля перекладу
- **note** (*string*) – нотатка модуля перекладу
- **flags** (*string*) – прапорці модуля перекладу
- **labels** (*array*) – мітки модулів перекладу, доступні у початкових модулях
- **state** (*int*) – стан модуля, 0 — не перекладено, 10 — потребує редагування, 20 — перекладено, 30 — затверджено, 100 — лише читання
- **fuzzy** (*boolean*) – чи позначено модуль як неточний або такий, який потребує рецензування
- **translated** (*boolean*) – чи є модуль перекладеним
- **approved** (*boolean*) – чи затверджено переклад
- **position** (*int*) – позиція модуля у файлі перекладу
- **has_suggestion** (*boolean*) – чи має модуль пропозиції перекладу
- **has_comment** (*boolean*) – чи має модуль коментарі
- **has_failing_check** (*boolean*) – чи пройшов модуль усі перевірки
- **num_words** (*int*) – кількість початкових слів
- **priority** (*int*) – пріоритетність перекладу; типове значення — 100

- **id** (*int*) – ідентифікатор модуля
- **explanation** (*string*) – Пояснення щодо рядка, доступне у початкових модулях, див. [Додаткові відомості щодо початкових рядків](#)
- **extra_flags** (*string*) – Додаткові прапорці рядків, доступні у початкових модулях, див. [Налаштування поведінки за допомогою прапорців](#)
- **web_url** (*string*) – адреса, за якою можна редагувати модуль
- **source_unit** (*string*) – Посилання на модуль початкового коду; див. [GET /api/units/\(int:id\)/](#)
- **pending** (*boolean*) – чи є модуль частиною черги на запис
- **timestamp** (*timestamp*) – вік рядків

PATCH /api/units/(int: id) /

Нове в версії 4.3.

Виконує часткове оновлення модуля перекладу.

Параметри

- **id** (*int*) – Ід. модуля

Надіслати запит щодо об'єкта JSON

- **state** (*int*) – стан модуля, 0 — не перекладено, 10 — потребує редагування, 20 — перекладено, 30 — затверджено (має бути увімкнено процедуру перекладу з рецензуванням, див. [Спеціалізовані рецензенти](#))
- **target** (*array*) – рядок перекладу
- **explanation** (*string*) – Пояснення щодо рядка, доступне у початкових модулях, див. [Додаткові відомості щодо початкових рядків](#)
- **extra_flags** (*string*) – Додаткові прапорці рядків, доступні у початкових модулях, див. [Налаштування поведінки за допомогою прапорців](#)

Об'єкт відповіді JSON

- **labels** (*array*) – мітки, доступні у початкових модулях

PUT /api/units/(int: id) /

Нове в версії 4.3.

Виконує повне оновлення модуля перекладу.

Параметри

- **id** (*int*) – Ід. модуля

Надіслати запит щодо об'єкта JSON

- **state** (*int*) – стан модуля, 0 — не перекладено, 10 — потребує редагування, 20 — перекладено, 30 — затверджено (має бути увімкнено процедуру перекладу з рецензуванням, див. [Спеціалізовані рецензенти](#))
- **target** (*array*) – рядок перекладу
- **explanation** (*string*) – Пояснення щодо рядка, доступне у початкових модулях, див. [Додаткові відомості щодо початкових рядків](#)
- **extra_flags** (*string*) – Додаткові прапорці рядків, доступні у початкових модулях, див. [Налаштування поведінки за допомогою прапорців](#)

Об'єкт відповіді JSON

- **labels** (*array*) – мітки, доступні у початкових модулях

DELETE `/api/units/(int: id) /`

Нове в версії 4.3.

Вилучає модуль перекладу.

Параметри

- **id** (*int*) – Ід. модуля

1.12.12 Зміни

Нове в версії 2.10.

GET `/api/changes/`

Змінено в версії 4.1: Фільтрування змін було впроваджено у випуску 4.1.

Повертає список змін у перекладі.

Дивись також:

Атрибути об'єкта зміни документовано у `GET /api/units/(int:id) /`.

Параметри запиту

- **user** (*string*) – Ім'я користувача для фільтрів
- **action** (*int*) – Дія для фільтрування, можна використовувати декілька разів
- **timestamp_after** (*timestamp*) – Часова позначка у форматі ISO 8601 для показу списку наступних змін
- **timestamp_before** (*timestamp*) – Часова позначка у форматі ISO 8601 для показу списку попередніх змін

GET `/api/changes/(int: id) /`

Повертає відомості щодо зміни перекладу.

Параметри

- **id** (*int*) – Ідентифікатор зміни

Об'єкт відповіді JSON

- **unit** (*string*) – адреса пов'язаного об'єкта модуля
- **translation** (*string*) – Адреса пов'язаного об'єкта перекладу
- **component** (*string*) – адреса пов'язаного об'єкта складника
- **user** (*string*) – адреса пов'язаного об'єкта користувача
- **author** (*string*) – адреса або пов'язаний об'єкт автора
- **timestamp** (*timestamp*) – часова позначка події
- **action** (*int*) – числовий ідентифікатор дії
- **action_name** (*string*) – текстовий опис дії
- **target** (*string*) – змінений текст або подробиці події
- **id** (*int*) – змінити ідентифікатор

1.12.13 Знімки екрана

Нове в версії 2.14.

GET /api/screenshots/

Повертає список даних рядків знімків вікон.

Дивись також:

Атрибути об'єкта знімка вікна документовано у `GET /api/screenshots/(int:id)/`.

GET /api/screenshots/(int: id) /

Повертає дані щодо даних знімка вікна.

Параметри

- **id** (*int*) – Ідентифікатор знімка вікна

Об'єкт відповіді JSON

- **name** (*string*) – назва знімка вікна
- **component** (*string*) – адреса пов'язаного об'єкта складника
- **file_url** (*string*) – адреса для отримання файла; див. `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – пов'язати із відповідними даними початкового рядка, див. `GET /api/units/(int:id)/`

GET /api/screenshots/(int: id) /file/

Отримати зображення знімка вікна.

Параметри

- **id** (*int*) – Ідентифікатор знімка вікна

POST /api/screenshots/(int: id) /file/

Замінити зображення знімка вікна.

Параметри

- **id** (*int*) – Ідентифікатор знімка вікна

Параметри форми

- **file image** – Вивантажений файл

Приклад для CURL:

```
curl -X POST \
  -F image=@image.png \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/screenshots/1/file/
```

POST /api/screenshots/(int: id) /units/

Пов'язати початковий рядок зі знімком вікна.

Параметри

- **id** (*int*) – Ідентифікатор знімка вікна

Параметри форми

- **string unit_id** – Ід. модуля

Об'єкт відповіді JSON

- **name** (*string*) – назва знімка вікна
- **translation** (*string*) – Адреса пов'язаного об'єкта перекладу

- **file_url** (*string*) – адреса для отримання файла; див. *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – пов'язати із відповідними даними початкового рядка, див. *GET /api/units/(int:id)/*

**DELETE /api/screenshots/(int: id)/units/
int: unit_id**

Вилучити прив'язку початкового рядка до знімка вікна.

Параметри

- **id** (*int*) – Ідентифікатор знімка вікна
- **unit_id** – Ідентифікатор модуля початкового рядка

POST /api/screenshots/

Створює знімок вікна.

Параметри форми

- **file image** – Вивантажений файл
- **string name** – Назва знімка екрана
- **string project_slug** – Ідентифікатор проєкту
- **string component_slug** – Ідентифікатор складника
- **string language_code** – Код мови

Об'єкт відповіді JSON

- **name** (*string*) – назва знімка вікна
- **component** (*string*) – адреса пов'язаного об'єкта складника
- **file_url** (*string*) – адреса для отримання файла; див. *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – пов'язати із відповідними даними початкового рядка, див. *GET /api/units/(int:id)/*

PATCH /api/screenshots/(int: id) /

Редагувати часткові відомості щодо знімка вікна.

Параметри

- **id** (*int*) – Ідентифікатор знімка вікна

Об'єкт відповіді JSON

- **name** (*string*) – назва знімка вікна
- **component** (*string*) – адреса пов'язаного об'єкта складника
- **file_url** (*string*) – адреса для отримання файла; див. *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – пов'язати із відповідними даними початкового рядка, див. *GET /api/units/(int:id)/*

PUT /api/screenshots/(int: id) /

Редагувати повні відомості щодо знімка вікна.

Параметри

- **id** (*int*) – Ідентифікатор знімка вікна

Об'єкт відповіді JSON

- **name** (*string*) – назва знімка вікна

- **component** (*string*) – адреса пов'язаного об'єкта складника
- **file_url** (*string*) – адреса для отримання файла; див. `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – пов'язати із відповідними даними початкового рядка, див. `GET /api/units/(int:id)/`

DELETE /api/screenshots/(int: id) /

Вилучити знімок.

Параметри

- **id** (*int*) – Ідентифікатор знімка вікна

1.12.14 Додатки

Нове в версії 4.4.1.

GET /api/addons/

Повертає список додатків.

Дивись також:

Атрибути об'єкта додатка документовано у `GET /api/addons/(int:id) /`.

GET /api/addons/(int: id) /

Повертає дані щодо додатка.

Параметри

- **id** (*int*) – Ідентифікатор додатка

Об'єкт відповіді JSON

- **name** (*string*) – назва додатка
- **component** (*string*) – адреса пов'язаного об'єкта складника
- **configuration** (*object*) – Необов'язкові налаштування додатка

Дивись також:

Додатки

**POST /api/components/(string: project) /
string: component/addons/**

Створює додаток.

Параметри

- **project_slug** (*string*) – Ідентифікатор проекту
- **component_slug** (*string*) – Ідентифікатор складника

Надіслати запит щодо об'єкта JSON

- **name** (*string*) – назва додатка
- **configuration** (*object*) – Необов'язкові налаштування додатка

PATCH /api/addons/(int: id) /

Редагувати часткові відомості щодо додатка.

Параметри

- **id** (*int*) – Ідентифікатор додатка

Об'єкт відповіді JSON

- **configuration** (*object*) – Необов'язкові налаштування додатка

PUT /api/addons/ (int: id) /

Редагувати повні відомості щодо додатка.

Параметри

- **id** (*int*) – Ідентифікатор додатка

Об'єкт відповіді JSON

- **configuration** (*object*) – Необов'язкові налаштування додатка

DELETE /api/addons/ (int: id) /

Видалити додаток.

Параметри

- **id** (*int*) – Ідентифікатор додатка

1.12.15 Списки складників

Нове в версії 4.0.

GET /api/component-lists/

Повертає список списків складників.

Дивись також:

Атрибути об'єкта списку складників документовано у `GET /api/component-lists/(str:slug)/`.

GET /api/component-lists/ (str: slug) /

Повертає відомості щодо списку складників.

Параметри

- **slug** (*string*) – Ідентифікатор списку складників

Об'єкт відповіді JSON

- **name** (*string*) – назва списку складників
- **slug** (*string*) – ідентифікатор списку складників
- **show_dashboard** (*boolean*) – визначає, чи слід показувати його на панелі приладів
- **components** (*array*) – посилання на пов'язані складники; див. `GET /api/components/(string:project)/(string:component)/`
- **auto_assign** (*array*) – правила автоматичної прив'язки

PUT /api/component-lists/ (str: slug) /

Змінює параметри списку складників.

Параметри

- **slug** (*string*) – Ідентифікатор списку складників

Надіслати запит щодо об'єкта JSON

- **name** (*string*) – назва списку складників
- **slug** (*string*) – ідентифікатор списку складників
- **show_dashboard** (*boolean*) – визначає, чи слід показувати його на панелі приладів

PATCH `/api/component-lists/ (str: slug) /`

Змінює параметри списку складників.

Параметри

- **slug** (*string*) – Ідентифікатор списку складників

Надіслати запит щодо об'єкта JSON

- **name** (*string*) – назва списку складників
- **slug** (*string*) – ідентифікатор списку складників
- **show_dashboard** (*boolean*) – визначає, чи слід показувати його на панелі приладів

DELETE `/api/component-lists/ (str: slug) /`

Вилучає список складників.

Параметри

- **slug** (*string*) – Ідентифікатор списку складників

POST `/api/component-lists/ (str: slug) /components/`

Пов'язати складник зі списком складників.

Параметри

- **slug** (*string*) – Ідентифікатор списку складників

Параметри форми

- **string component_id** – Ідентифікатор складника

DELETE `/api/component-lists/ (str: slug) /components/
str: component_slug`

Відв'язати складник зі списку складників.

Параметри

- **slug** (*string*) – Ідентифікатор списку складників
- **component_slug** (*string*) – Ідентифікатор складника

1.12.16 Глосарій

Змінено в версії 4.5: У нових версіях глосарії зберігаються як звичайні складники, переклади і рядки. Будь ласка, скористайтеся відповідним API.

1.12.17 Завдання

Нове в версії 4.4.

GET `/api/tasks/`

Наразі перелік завдань недоступний.

GET `/api/tasks/ (str: uuid) /`

Повертає відомості про завдання

Параметри

- **uuid** (*string*) – UUID завдання

Об'єкт відповіді JSON

- **completed** (*boolean*) – Чи виконано завдання

- **progress** (*int*) – Перебіг виконання завдання у відсотках
- **result** (*object*) – Відомості про результат або перебіг виконання завдання
- **log** (*string*) – Журнал завдань

1.12.18 Метрика

GET /api/metrics/

Повертає метрику сервера.

Об'єкт відповіді JSON

- **units** (*int*) – Кількість модулів
- **units_translated** (*int*) – Кількість перекладених модулів
- **users** (*int*) – Кількість користувачів
- **changes** (*int*) – Кількість змін
- **projects** (*int*) – Кількість проєктів
- **components** (*int*) – Кількість складників
- **translations** (*int*) – Кількість перекладів
- **languages** (*int*) – Кількість використаних мов
- **checks** (*int*) – Кількість непройдених перевірок якості
- **configuration_errors** (*int*) – Кількість помилок у налаштуваннях
- **suggestions** (*int*) – Кількість пропозицій у черзі
- **celery_queues** (*object*) – Довжини черг Celery, див. *Фонові завдання з використанням Celery*
- **name** (*string*) – Налаштована назва сервера

1.12.19 Обробники сповіщень

Обробники сповіщень надають змогу стороннім програмам сповіщати Weblate про оновлення у репозиторії коду системи керування версіями.

Ви можете скористатися кінцевими точками для проєктів, складників та перекладів у репозиторії для оновлення окремих репозиторіїв. Див. документацію у `POST /api/projects/(string:project)/repository/`.

GET /hooks/update/(string: project) /
string: component/

Застаріло починаючи з версії 2.6: Будь ласка, скористайтеся замість цього `POST /api/components/(string:component)/repository/`, яка працює належним чином із розпізнаванням для обмежених ACL проєктів.

Вмикає оновлення складника (отримання даних з системи керування версіями і сканування змін у перекладах).

GET /hooks/update/(string: project) /

Застаріло починаючи з версії 2.6: Будь ласка, скористайтеся замість цього `POST /api/projects/(string:project)/repository/`, яка працює належним чином із розпізнаванням для обмежених ACL проєктів.

Вмикає оновлення усіх складників проєкту (отримання даних із системи керування версіями і сканування змін у перекладах).

POST /hooks/github/

Особливий скрипт для обробки сповіщень GitHub і автоматичного оновлення відповідних складників.

Примітка: До GitHub включено безпосередню підтримку сповіщень Weblate: увімкніть скрипт перехоплення для служби Weblate у параметрах репозиторію і встановіть адресу, яка відповідає встановленому вами екземпляру Weblate.

Дивись також:

Автоматичне отримання змін з GitHub

Настанови щодо налаштування інтеграції із GitHub

<https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks>

Загальні відомості щодо вебскриптів GitHub

ENABLE_HOOKS

Вмикання скриптів для усього екземпляра Weblate

POST /hooks/gitlab/

Особливий скрипт для обробки сповіщень GitLab і автоматичного оновлення відповідних складників.

Дивись також:

Автоматичне отримання змін з GitLab

Настанови щодо налаштування інтеграції із GitLab

<https://docs.gitlab.com/ee/user/project/integrations/webhooks.html>

Загальні відомості щодо вебскриптів GitLab

ENABLE_HOOKS

Вмикання скриптів для усього екземпляра Weblate

POST /hooks/bitbucket/

Особливий скрипт для обробки сповіщень Bitbucket і автоматичного оновлення відповідних складників.

Дивись також:

Автоматичне отримання змін з Bitbucket

Настанови щодо налаштування інтеграції із Bitbucket

<https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/>

Загальні відомості щодо вебскриптів Bitbucket

ENABLE_HOOKS

Вмикання скриптів для усього екземпляра Weblate

POST /hooks/pagure/

Нове в версії 3.3.

Особливий скрипт для обробки сповіщень Pagure і автоматичного оновлення відповідних складників.

Дивись також:

Автоматичне отримання змін з Pagure

Настанови щодо налаштування інтеграції із Pagure

https://docs.pagure.org/pagure/usage/using_webhooks.html

Загальні відомості щодо вебскриптів Pagure

ENABLE_HOOKS

Вмикання скриптів для усього екземпляра Weblate

POST /hooks/azure/

Нове в версії 3.8.

Особливий скрипт для обробки сповіщень Azure DevOps і автоматичного оновлення відповідних складників.

Примітка: Будь ласка, переконайтеся, що для параметра *Подробиці щодо ресурсу для надсилання* встановлено значення *Усі*, інакше Weblate не зможе встановити відповідність вашого сховища Azure.

Дивись також:

Автоматичне отримання змін зі репозиторіїв Azure

Настанови щодо налаштування інтеграції із Azure

https:

[//learn.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops](https://learn.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops)

Загальні відомості щодо вебскриптів Azure DevOps

ENABLE_HOOKS

Вмикання скриптів для усього екземпляра Weblate

POST /hooks/gitea/

Нове в версії 3.9.

Особливий скрипт для обробки сповіщень вебскриптів Gitea і автоматичного оновлення відповідних складників.

Дивись також:

Автоматичне отримання змін зі репозиторіїв Gitea

Настанови щодо налаштування інтеграції із Gitea

https://docs.gitea.io/en-us/webhooks/

Загальні відомості щодо вебскриптів Gitea

ENABLE_HOOKS

Вмикання скриптів для усього екземпляра Weblate

POST /hooks/gitee/

Нове в версії 3.9.

Особливий скрипт для обробки сповіщень вебскриптів Gitee і автоматичного оновлення відповідних складників.

Дивись також:

Автоматичне отримання змін зі репозиторіїв Gitee

Настанови щодо налаштування інтеграції із Gitee

https://gitee.com/help/categories/40

Загальні відомості щодо вебскриптів Gitee

ENABLE_HOOKS

Вмикання скриптів для усього екземпляра Weblate

1.12.20 Експорт

У Weblate надає різні засоби експортування для уможливлення подальшої обробки даних.

GET `/exports/stats/(string: project) /`
string: `component/`

Параметри запиту

- **format** (*string*) – Формат виведення: `json` або `csv`

Застаріло починаючи з версії 2.6: Будь ласка, скористайтесь замість цього `GET /api/components/(string:project)/(string:component)/statistics/` і `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/`; надає доступ також і до керованих за допомогою ACL проєктів.

Отримує статистичні дані для вказаного складника у вказаному форматі.

Приклад запиту:

```
GET /exports/stats/weblate/main/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Приклад відповіді:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

[
  {
    "code": "cs",
    "failing": 0,
    "failing_percent": 0.0,
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "last_author": "Michal Čihař",
    "last_change": "2012-03-28T15:07:38+00:00",
    "name": "Czech",
    "total": 436,
    "total_words": 15271,
    "translated": 436,
    "translated_percent": 100.0,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/cs/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/main/cs/"
  },
  {
    "code": "nl",
    "failing": 21,
    "failing_percent": 4.8,
    "fuzzy": 11,
    "fuzzy_percent": 2.5,
    "last_author": null,
    "last_change": null,
    "name": "Dutch",
    "total": 436,
    "total_words": 15271,
    "translated": 319,
    "translated_percent": 73.2,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/nl/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/main/nl/"
  }
]
```

(continues on next page)

(continued from previous page)

```

    },
    {
      "code": "el",
      "failing": 11,
      "failing_percent": 2.5,
      "fuzzy": 21,
      "fuzzy_percent": 4.8,
      "last_author": null,
      "last_change": null,
      "name": "Greek",
      "total": 436,
      "total_words": 15271,
      "translated": 312,
      "translated_percent": 71.6,
      "translated_words": 3201,
      "url": "http://hosted.weblate.org/engage/weblate/el/",
      "url_translate": "http://hosted.weblate.org/projects/weblate/main/el/"
    }
  ]

```

1.12.21 RSS-стрічки

Зміни в перекладах експортуються до стрічок RSS.

GET /exports/rss/ (string: project) /
string: component/string: language/

Отримує RSS-стрічку з останніми змінами в перекладі.

GET /exports/rss/ (string: project) /
string: component/

Отримує RSS-стрічку з останніми змінами у складнику.

GET /exports/rss/ (string: project) /

Отримує RSS-стрічку з останніми змінами у проєкті.

GET /exports/rss/language/ (string: language) /

Отримує RSS-стрічку з останніми змінами в мові.

GET /exports/rss/

Отримує RSS-стрічку з останніми змінами в екземплярі Weblate.

Дивись також:

[RSS у Вікіпедії](#)

1.13 Клієнт Weblate

Нове в версії 2.7: Повну підтримку допоміжного засобу `wlc` реалізовано з версії Weblate 2.7. Якщо ви користуєтесь давнішою версією, ви можете зіткнутися з певними несумісностями у програмному інтерфейсі.

1.13.1 Установлення

Клієнт Weblate постачається окремо і включає до себе модуль Python. Щоб користуватися наведеними нижче командами, вам слід встановити `wlc`:

```
pip install wlc
```

1.13.2 Користування Docker

Крім того, клієнт Weblate доступний у форматі образу для Docker.

Образ оприлюднено на Docker Hub: <https://hub.docker.com/r/weblate/wlc>

Установлення:

```
docker pull weblate/wlc
```

Контейнер Docker використовує типові параметри Weblate і встановлює з'єднання із програмним інтерфейсом, який розгорнуто на локальному вузлі. URL-адресу API та API_KEY можна налаштувати за допомогою аргументів, які приймаються Weblate.

У команді запуску контейнера використовують такий синтаксис:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Приклад:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

Якщо вам хочеться передати ваш *Файли налаштувань* до контейнера Docker, найпростішим способом буде додати ваш поточний каталог томом `/home/weblate`:

```
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

1.13.3 Перші кроки

Налаштування `wlc` зберігаються у `~/ .config/weblate` (інші місця див. у *Файли налаштувань*). Будь ласка, створіть його так, щоб усе відповідало вашому середовищу:

```
[weblate]
url = https://hosted.weblate.org/api/

[keys]
https://hosted.weblate.org/api/ = APIKEY
```

Далі, ви можете викликати команди на типовому сервері:

```
wlc ls
wlc commit sandbox/hello-world
```

Дивись також:

Файли налаштувань

1.13.4 Короткий опис

```
wlc [arguments] <command> [options]
```

Команди вказують на те, яку дію має бути виконано.

1.13.5 Опис

Клієнт Weblate складається з бібліотеки мовою Python і засобу командного рядка для керування Weblate віддалено за допомогою *програмного інтерфейсу*. Викликати засіб командного рядка можна за допомогою команди **wlc**, його вбудовано до *wlc*.

Аргументи

Програма приймає вказані нижче аргументи, які визначають формат виведення даних або екземпляр Weblate, яким слід скористатися. Ці аргументи слід вводити перед будь-якою командою.

--format {csv,json,text,html}

Вказати формат виведення даних.

--url URL

Вказати адресу програмного інтерфейсу. Має вищий пріоритет за будь-яке значення, яке буде виявлено у файлі налаштувань, див. *Файли налаштувань*. Адреса має завершуватися `/api/`. Приклад: `https://hosted.weblate.org/api/`.

--key KEY

Вказати ключ до програмного інтерфейсу, яким слід скористатися. Має вищий пріоритет за будь-яке значення, яке буде виявлено у файлі налаштувань, див. *Файли налаштувань*. Знайти ваш ключ можна у профілі на Weblate.

--config PATH

Перевизначає шлях до файла налаштувань, див. *Файли налаштувань*.

--config-section SECTION

Перевизначає розділ файла налаштувань для використання, див. *Файли налаштувань*.

Команди

Можна скористатися такими командами:

version

Виводить поточну версію.

list-languages

Виводить список використаних у Weblate мов.

list-projects

Виводить список проєктів у Weblate.

list-components

Виводить список складників у Weblate.

list-translations

Виводить список перекладів у Weblate.

show

Показує об'єкт Weblate (переклад, складник або проєкт).

ls

Виводить список об'єкта Weblate (переклад, складник або проєкт).

commit

Вносить зміни, які виконано в об'єкті Weblate (перекладі, складнику або проєкті).

pull

Отримує зміни у віддаленому репозиторії до об'єкта Weblate (перекладу, складника або проєкту).

push

Записує зміни у об'єкті Weblate до віддаленого репозиторію (переклад, складник або проєкт).

reset

Нове в версії 0.7: Підтримку передбачено з версії wlc 0.7.

Скидає зміни в об'єкті Weblate так, щоб дані збігалися із даними у віддаленому репозиторії (переклад, складник або проєкт).

cleanup

Нове в версії 0.9: Підтримку передбачено з версії wlc 0.9.

Влучає усі незаписані зміни в об'єкті Weblate так, щоб дані збігалися із даними у віддаленому репозиторії (переклад, складник або проєкт).

repo

Показує стан у репозиторії для вказаного об'єкта Weblate (перекладу, складника або проєкту).

stats

Показує докладну статистику щодо вказаного об'єкта Weblate (перекладу, складника або проєкту).

lock-status

Нове в версії 0.5: Підтримку передбачено з версії wlc 0.5.

Показує стан блокування.

lock

Нове в версії 0.5: Підтримку передбачено з версії wlc 0.5.

Блокує складник від подальшого перекладу на Weblate.

unlock

Нове в версії 0.5: Підтримку передбачено з версії wlc 0.5.

Розблоковує переклад складника Weblate.

changes

Нове в версії 0.7: Підтримку передбачено з версії wlc 0.7 і версії Weblate 2.10.

Показує зміни у вказаному об'єкті.

download

Нове в версії 0.7: Підтримку передбачено з версії wlc 0.7.

Отримує файл перекладу.

--convert

Перетворює формат файла. Якщо формат файла не вказано, перетворення на сервері не відбуватиметься, а файл буде отримано у форматі, у якому він зберігається у репозиторії.

--output

Вказує файл для зберігання виведених даних. Якщо файл не вказано, дані буде виведено до стандартного виведення (stdout).

upload

Нове в версії 0.9: Підтримку передбачено з версії wlc 0.9.

Вивантажує файл перекладу.

--overwrite

Перезаписати наявні переклади під час вивантаження.

--input

Файл, з якого читатимуться дані. Якщо не вказано, дані буде прочитано зі stdin.

--method

Спосіб вивантаження, яким слід скористатися. Див. [Способи імпортування](#).

--fuzzy

Обробка неточних (позначених для редагування) перекладів (*empty*, *process*, *approve*)

--author-name

Ім'я автора, щоб перевизначити ім'я із запису поточного користувача системи

--author-email

Адреса електронної пошти автора, щоб перевизначити дані із запису користувача системи

Підказка: Докладніші відомості щодо виклику окремих команд можна отримати за допомогою параметра `--help`. Приклад: `wlc ls --help`.

1.13.6 Файли налаштувань**.weblate, .weblate.ini, weblate.ini**

Змінено в версії 1.6: Можна використовувати файли із суфіксами назв *.ini*.

Файл налаштувань окремого проекту

C:\Users\NAME\AppData\weblate.ini

Нове в версії 1.6.

Файл налаштувань користувача у Windows.

~/.config/weblate

Файл налаштувань користувача

/etc/xdg/weblate

Загальносистемний файл налаштувань

У програмі реалізовано специфікацію XDG, тому ви можете скоригувати розташування файлів налаштувань за допомогою змінних середовища `XD_CONFIG_HOME` і `XD_CONFIG_DIRS`. У Windows пріоритетним місцем зберігання файла налаштувань є каталог APPDATA.

Вказані нижче параметри може бути налаштовано за допомогою розділу `[weblate]` (ви можете змінити розділ за допомогою параметра `--config-section`):

key

Ключ доступу до програмного інтерфейсу Weblate.

url

Адреса сервера програмного інтерфейсу, типово, `http://127.0.0.1:8000/api/`.

translation

Шлях до типового перекладу — складника або проекту.

Файл налаштувань є файлом формату INI. Приклад:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

У розділі [keys] можна зберігати додаткові ключі до програмного інтерфейсу:

```
[keys]
https://hosted.weblate.org/api/ = APIKEY
```

Це надасть вам змогу зберігати ключі у ваших особистих параметрах, використовуючи налаштування з .weblate у репозиторії системи керування версіями так, щоб wlc було відомо, з яким сервером слід працювати.

1.13.7 Приклади

Вивести версію поточної програми:

```
$ wlc version
version: 0.1
```

Список усіх проєктів:

```
$ wlc list-projects
name: Hello
slug: hello
url: http://example.com/api/projects/hello/
web: https://weblate.org/
web_url: http://example.com/projects/hello/
```

Вивантажити файл перекладу:

```
$ wlc upload project/component/language --input /tmp/hello.po
```

Ви також можете позначити проєкт, з яким має працювати wlc:

```
$ cat .weblate
[weblate]
url = https://hosted.weblate.org/api/
translation = weblate/application

$ wlc show
branch: main
file_format: po
source_language: en
filemask: weblate/locale/*/LC_MESSAGES/django.po
git_export: https://hosted.weblate.org/git/weblate/application/
license: GPL-3.0+
license_url: https://spdx.org/licenses/GPL-3.0+
name: Application
new_base: weblate/locale/django.pot
project: weblate
repo: git://github.com/WeblateOrg/weblate.git
slug: application
template:
url: https://hosted.weblate.org/api/components/weblate/application/
vcs: git
web_url: https://hosted.weblate.org/projects/weblate/application/
```

З цими налаштуваннями доволі просто надсилати зміни з черги у поточному проєкті:

```
$ wlc commit
```

1.14 Програмний інтерфейс Python Weblate

1.14.1 Установлення

Програмний інтерфейс Python постачається окремо. Вам слід встановити *Клієнт Weblate* (wlc), щоб отримати доступ до нього.

```
pip install wlc
```

1.14.2 wlc

WeblateException

exception `wlc.WeblateException`

Основний клас для всіх винятків.

Weblate

class `wlc.Weblate` (*key=""*, *url=None*, *config=None*)

Параметри

- **key** (*str*) – Ключ користувача
- **url** (*str*) – Адреса програмного інтерфейсу сервера. Якщо не вказано, буде використано типу.
- **config** (`wlc.config.WeblateConfig`) – Об'єкт налаштувань, перевизначає усі інші параметри.

Клас доступу до програмного інтерфейсу, визначає ключ до програмного інтерфейсу і, необов'язково, адресу програмного інтерфейсу.

get (*path*)

Параметри

path (*str*) – Шлях запити

Тип повернення

object

Виконує один виклик GET програмного інтерфейсу.

post (*path*, ***kwargs*)

Параметри

path (*str*) – Шлях запити

Тип повернення

object

Виконує один виклик GET програмного інтерфейсу.

1.14.3 wlc.config

WeblateConfig

```
class wlc.config.WeblateConfig (section='wlc')
```

Параметри

section (*str*) – Розділ налаштувань, яким слід скористатися

Обробник файла налаштувань, у якому використано специфікацію XDG.

load (*path=None*)

Параметри

path (*str*) – Шлях, з якого слід завантажувати налаштування.

Завантажує налаштування з файла. Якщо файл не вказано, завантаження відбуватиметься з файла налаштувань *wlc* (*~/.config/wlc*), який розташовано у вашому каталозі налаштувань XDG (*/etc/xdg/wlc*).

1.14.4 wlc.main

```
wlc.main.main (settings=None, stdout=None, args=None)
```

Параметри

- **settings** (*list*) – Параметри для перевизначення у форматі списку кортежів
- **stdout** (*object*) – Файловий об'єкт *stdout* для виведення даних. Типовим є *sys.stdout*
- **args** (*list*) – Аргументи командного рядка для обробки. Типовими є *sys.args*

Головна точка входу для інтерфейсу командного рядка.

```
@wlc.main.register_command (command)
```

Декоратор для реєстрації класу *Command* у основному обробнику, який використовується *main()*.

Command

```
class wlc.main.Command (args, config, stdout=None)
```

Основний клас для виклику команд.

2.1 Настанови з налаштування

2.1.1 Установлення Weblate

Установлення за допомогою Docker

Якщо Weblate розгорнуто у середовищі Docker, ви зможете отримати ваш працездатний і запущений особистий екземпляр Weblate за декілька секунд. Усі залежності Weblate вже включено. Типовим засобом роботи із базою даних у таких екземплярах є PostgreSQL.

Вимоги щодо обладнання

Weblate має працювати на будь-якому сучасному обладнанні без проблем. Нижче наведено мінімальні налаштування, які потрібні для запуску Weblate на одному вузлі (Weblate, бази даних і вебсервера):

- 3 ГБ оперативної пам'яті
- 2 ядра процесора
- 1 ГБ вільного місця на диску

Чим більше пам'яті — тим краще. Пам'ять використовується для кешування на усіх рівнях (у файловій системі, базі даних та Weblate).

Збільшення кількості одночасних користувачів системи збільшує кількість потрібних для роботи ядер центрального процесора. Для сотень складників перекладу рекомендовано використовувати принаймні 4 ГБ оперативної пам'яті.

Типовий об'єм сховища даних близько 300 МБ на 1 мільйон збережених слів. Об'єм даних, потрібний для зберігання клонованих репозиторіїв коду, може бути різним, але Weblate намагається його скоротити створенням поверхневих клонів.

Примітка: Реальні вимоги до встановленого вами Weblate значно залежать від розміру перекладів, які ним керуються.

Установлення

У наведених нижче прикладах припускається, що у вас є працездатне середовище Docker із встановленим `docker-compose`. Будь ласка, ознайомтеся із документацією до Docker, щоб отримати додаткові настанови.

1. Клонування репозиторію `weblate-docker`:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Створіть файл `docker-compose.override.yml` з вашими параметрами. Із повним списком змінних середовища можна ознайомитися у розділі [Змінні середовища Docker](#).

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
    environment:
      WEBLATE_EMAIL_HOST: smtp.example.com
      WEBLATE_EMAIL_HOST_USER: user
      WEBLATE_EMAIL_HOST_PASSWORD: pass
      WEBLATE_SERVER_EMAIL: weblate@example.com
      WEBLATE_DEFAULT_FROM_EMAIL: weblate@example.com
      WEBLATE_SITE_DOMAIN: weblate.example.com
      WEBLATE_ADMIN_PASSWORD: password for the admin user
      WEBLATE_ADMIN_EMAIL: weblate.admin@example.com
```

Примітка: Якщо `WEBLATE_ADMIN_PASSWORD` не встановлено, для адміністративного користувача буде створено випадковий пароль, який буде показано під час першого запуску.

У наданому прикладі Weblate очікуватиме на дані на порту 80. Змініть прив'язку портів у файлі `docker-compose.override.yml`, щоб скоригувати цю типову поведінку.

3. Запуск контейнерів Weblate:

```
docker-compose up
```

Насолоджуйтеся розгорнутим Weblate. Доступ до нього можна отримати за допомогою порту 80 контейнера `weblate`.

Змінено в версії 2.15-2: Конфігурацію було нещодавно змінено: раніше був окремий контейнер вебсервера. Починаючи з версії 2.15-2, вебсервер вбудовано до контейнера Weblate.

Змінено в версії 3.7.1-6: Станом на липень 2019 року (починаючи з мітки 3.7.1-6), контейнери не запускаються від імені користувача `root`. Це призвело до зміни порту зв'язку з 80 на 8080.

Дивись також:

[Виклик команд керування](#)

Вибір теґу концентратора Docker

На концентраторі Docker ви можете користуватися вказаними нижче теґами. Із повним списком можна ознайомитися тут: <https://hub.docker.com/r/weblate/weblate/tags/>

Назва теґу	Опис	Варіант використання
latest	Стабільний випуск Weblate, відповідає найновішому випуску	Оновлення-накладки у промисловому середовищі
<VERSION>-	Стабільний випуск Weblate	Надійний розгорнутий екземпляр у промисловому середовищі
edge	Стабільний випуск Weblate із змінами з гілки розробки у контейнері Docker (наприклад, із оновленими залежностями)	Оновлення-накладки у проміжному тестовому середовищі
edge-<DATE>-	Стабільний випуск Weblate із змінами з гілки розробки у контейнері Docker (наприклад, із оновленими залежностями)	Надійний розгорнутий екземпляр у тестовому середовищі
bleeding	Версії Weblate у розробці з Git	Оновлення-накладки для тестування новітніх можливостей Weblate
bleeding-<DATE>-	Версії Weblate у розробці з Git	Надійний розгорнутий екземпляр для тестування новітніх можливостей Weblate

Тестування усіх образів виконується перед оприлюдненням у нашій системі неперервної інтеграції, тому навіть версія *bleeding* має бути безпечною для користування.

Контейнер Docker із підтримкою HTTPS

Будь ласка, ознайомтеся із розділом [Установлення](#), щоб отримати загальні настанови щодо розгортання. У цьому розділі наведено лише відмінності від типової процедури.

Використання власних сертифікатів SSL

Нове в версії 3.8-3.

Якщо у вас є власний сертифікат SSL, яким ви хочете скористатися, просто розмістіть його файли на томі даних Weblate (див. [Томи контейнера Docker](#)):

- `ssl/fullchain.pem` містить сертифікат, включно із будь-якими потрібними сертифікатами служби сертифікації
- `ssl/privkey.pem` містить закритий ключ

Власником обох цих файлів має бути той самий користувач, що і користувач, від імені якого запускається контейнер Docker. Маскою прав доступу до цих файлів має бути `600` (читання і запис лише для користувача-власника).

Крім того, контейнер Weblate прийматиме з'єднання на порту SSL 4443. Вам варто включити переспрямування портів для HTTPS у перевизначення композицій Docker:

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
      - 443:4443
```

Якщо на сервері вже обробляються дані інших сайтів, ймовірно, порти 80 і 443 використовуються оберненим проксі, наприклад NGINX. Щоб передати з'єднання HTTPS від NGINX до контейнера Docker, ви можете скористатися такими налаштуваннями:

```
server {
    listen 443;
    listen [::]:443;

    server_name <SITE_URL>;
    ssl_certificate /etc/letsencrypt/live/<SITE>/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/<SITE>/privkey.pem;

    location / {
        proxy_set_header HOST $host;
        proxy_set_header X-Forwarded-Proto https;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Host $server_name;
        proxy_pass https://127.0.0.1:<EXPOSED_DOCKER_PORT>;
    }
}
```

Замініть <SITE_URL>, <SITE> і <EXPOSED_DOCKER_PORT> реальними значеннями з вашого середовища.

Автоматична сертифікація SSL за допомогою Let's Encrypt

Якщо ви хочете скористатися автоматично створеними сертифікатами SSL від [Let's Encrypt](#) на загальнодоступній встановленій версії, вам слід додати обернений проксі HTTPS. Для цього буде використано додатковий контейнер Docker, [https-portal](#). Використання цього контейнера виконується за допомогою файла `docker-compose-https.yml`. Далі, вам слід створити файл `docker-compose-https.override.yml` із вашими параметрами:

```
version: '3'
services:
  weblate:
    environment:
      WEBLATE_EMAIL_HOST: smtp.example.com
      WEBLATE_EMAIL_HOST_USER: user
      WEBLATE_EMAIL_HOST_PASSWORD: pass
      WEBLATE_SITE_DOMAIN: weblate.example.com
      WEBLATE_ADMIN_PASSWORD: password for admin user
  https-portal:
    environment:
      DOMAINS: 'weblate.example.com -> http://weblate:8080'
```

Під час кожного виклику **docker-compose** вам слід передавати програмі обидва файли, а потім виконати:

```
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml
↵build
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml up
```

Оновлення контейнера Docker

Зазвичай, варто оновлювати лише контейнер Weblate, а версію контейнера PostgreSQL лишати незмінною, оскільки оновлення PostgreSQL є доволі болючою процедурою і, у більшості випадків, не дає значних переваг.

Змінено в версії 4.10-1: З моменту випуску Weblate 4.10-1 у контейнері Docker використано Django 4.0. Це потребує встановлення PostgreSQL 10 або новішої версії. Будь ласка, оновіть пакунки до оновлення Weblate. Докладніший опис можна знайти у розділах *Оновлення з 4.9 до 4.10* та *Оновлення контейнера PostgreSQL*.

Ви можете досягти цього фіксацією наявного docker-compose із простим отриманням найсвіжіших образів і наступним перезапуском:

```
# Fetch latest versions of the images
docker-compose pull
# Stop and destroy the containers
docker-compose down
# Spawn new containers in the background
docker-compose up -d
# Follow the logs during upgrade
docker-compose logs -f
```

Базу даних Weblate має бути автоматично перенесено на нову версію при першому запуску. Потреби у додаткових діях вручну виникнути не повинно.

Примітка: Підтримки оновлення до основних версій у Weblate не передбачено. Якщо ви працюєте із версією 3.x і хочете оновитися до 4.x, спочатку оновіться до найсвіжішої версії 4.0.x-у (на час написання цього підручника такою версією була 4.0.4-5), а далі продовжуйте оновлення до новіших версій.

Ймовірно, вам також варто оновити репозиторій docker-compose, хоча, у більшості випадків, потреби у цьому немає. Див. *Оновлення контейнера PostgreSQL* щодо оновлення сервера PostgreSQL.

Оновлення контейнера PostgreSQL

У контейнерах PostgreSQL не передбачено підтримки автоматичного оновлення до нових версій. Вам доведеться виконати оновлення вручну. Нижче показано кроки одного із варіантів оновлення.

Дивись також:

<https://github.com/docker-library/postgres/issues/37>

1. Зупинення контейнера Weblate:

```
docker-compose stop weblate cache
```

2. Резервне копіювання бази даних:

```
docker-compose exec database pg_dumpall --clean --username weblate > backup.sql
```

3. Зупинення контейнера бази даних:

```
docker-compose stop database
```

4. Вилучення тому PostgreSQL:

```
docker-compose rm -v database
docker volume remove weblate-docker_postgres-data
```

5. Скоригуйте ваш docker-compose.yml так, щоб у ньому було використано нову версію PostgreSQL.
6. Запуск контейнера бази даних:

```
docker-compose up -d database
```

7. Відновлення бази даних з резервної копії:

```
cat backup.sql | docker-compose exec -T database psql --username weblate --  
→ dbname postgres
```

8. (Необов'язкове) Оновити пароль для користувача Weblate. У цьому може виникнути потреба при перенесенні на PostgreSQL 14 або 15, оскільки спосіб зберігання паролів було змінено:

```
docker-compose exec -T database psql --username weblate --dbname postgres -c  
→ "ALTER USER weblate WITH PASSWORD 'weblate'"
```

9. Запуск решти контейнерів:

```
docker-compose up -d
```

Вхід від імені адміністратора

Після налаштування контейнера ви можете увійти до системи від імені користувача *admin* із використанням пароля, який надається у змінній *WEBLATE_ADMIN_PASSWORD*, або випадкового пароля, який створено під час першого запуску, якщо пароль не було встановлено за допомогою змінної.

Щоб скинути пароль *admin*, перезапустіть контейнер зі змінною *WEBLATE_ADMIN_PASSWORD*, значенням якої є новий пароль.

Дивись також:

WEBLATE_ADMIN_PASSWORD, *WEBLATE_ADMIN_NAME*, *WEBLATE_ADMIN_EMAIL*

Кількість процесів і споживання пам'яті

Кількість процесів обробки у uWSGI і Celery визначається автоматично на основі кількості процесорів. Це добре працює для більшості «хмарних» віртуальних машин, оскільки у них, зазвичай, встановлено декілька процесорів і доволі багато оперативної пам'яті.

Якщо на вашому комп'ютері багато процесорів і виникають проблеми нестачі пам'яті, спробуйте скоротити кількість обробників:

```
environment:  
  WEBLATE_WORKERS: 2
```

Ви також можете скоригувати окремі категорії обробників:

```
environment:  
  WEB_WORKERS: 4  
  CELERY_MAIN_OPTIONS: --concurrency 2  
  CELERY_NOTIFY_OPTIONS: --concurrency 1  
  CELERY_TRANSLATE_OPTIONS: --concurrency 1
```

Дивись також:

WEBLATE_WORKERS, *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*,
CELERY_MEMORY_OPTIONS, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*,
CELERY_BEAT_OPTIONS, *WEB_WORKERS*

Горизонтальне масштабування

Нове в версії 4.6.

Ви можете запускати кілька контейнерів Weblate для горизонтального масштабування служби. Том `/app/data` має бути спільним для всіх контейнерів. Радимо використовувати для нього кластерну файлову систему, зокрема GlusterFS. Том `/app/cache` має бути власним для кожного з контейнерів.

У кожного контейнера Weblate є визначена за допомогою змінної середовища `:envvar: WEBLATE_SERVICE` роль. Будь ласка, точно виконуйте настанови з документації, оскільки деякі служби у кластері мають запускатися лише один раз, крім того, має значення порядок запуску служб.

Прикладом налаштувань у репозиторії `docker-compose` є файл `docker-compose-split.yml`.

Змінні середовища Docker

Багато з *параметрів налаштування* може бути встановлено у контейнері Docker за допомогою змінних середовища, які описано нижче.

Якщо вам потрібно визначити параметр, який не буде відкрито за допомогою змінних середовища Docker, див. *Налаштування поза змінними середовища*.

Загальні параметри

WEBLATE_DEBUG

Налаштовує режим діагностики Django за допомогою *DEBUG*.

Приклад:

```
environment:
  WEBLATE_DEBUG: 1
```

Дивись також:

Вимкнути режим діагностики

WEBLATE_LOGLEVEL

Налаштовує докладність ведення журналу.

WEBLATE_LOGLEVEL_DATABASE

Налаштовує докладність ведення журналу для запитів до бази даних.

WEBLATE_SITE_TITLE

Налаштовує заголовок сайта, який буде показано у верхньому колонтитулі усіх сторінок.

WEBLATE_SITE_DOMAIN

Налаштовує домен сайта. Цей параметр слід вказати обов'язково.

Дивись також:

Установіть належний домен сайта, `SITE_DOMAIN`

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Налаштовує ім'я та адресу електронної пошти адміністратора сайта. Використовується одразу для встановлення *ADMINS* і створення користувача *admin* (див. *WEBLATE_ADMIN_PASSWORD*, щоб дізнатися про це більше).

Приклад:

```
environment:
  WEBLATE_ADMIN_NAME: Weblate admin
  WEBLATE_ADMIN_EMAIL: noreply@example.com
```

Дивись також:

Вхід від імені адміністратора, Належне налаштування записів адміністраторів, ADMINS

WEBLATE_ADMIN_PASSWORD

Установлює пароль для користувача *admin*.

- Якщо не встановлено і користувача *admin* не існує, його буде створено із випадковим паролем, який буде показано при першому запуску контейнера.
- Якщо не встановлено, а запис користувача *admin* існує, ніяких дій не виконуватиметься.
- Якщо встановлено, запис користувача *admin* буде скориговано під час кожного запуску контейнера відповідно до *WEBLATE_ADMIN_PASSWORD*, *WEBLATE_ADMIN_NAME* і *WEBLATE_ADMIN_EMAIL*.

Попередження: Зберігання пароля у файлі налаштувань може знизити рівень захищеності вашої системи. Цією змінною варто користуватися лише для початкового налаштування (або для надання змоги Weblate створити випадковий пароль при початковому запуску) або для відновлення доступу до сервера після втрати пароля.

Дивись також:

Вхід від імені адміністратора, WEBLATE_ADMIN_PASSWORD, WEBLATE_ADMIN_PASSWORD_FILE, WEBLATE_ADMIN_NAME, WEBLATE_ADMIN_EMAIL

WEBLATE_ADMIN_PASSWORD_FILE

Установлює шлях до файла, що містить пароль для користувача *admin*.

Дивись також:

WEBLATE_ADMIN_PASSWORD

WEBLATE_SERVER_EMAIL

Адреса електронної пошти, з якої надіслано повідомлення про помилки.

Дивись також:

SERVER_EMAIL, Налаштування надсилання електронної пошти

WEBLATE_DEFAULT_FROM_EMAIL

Налаштовує адресу для вихідних повідомлень електронної пошти.

Дивись також:

DEFAULT_FROM_EMAIL, Налаштування надсилання електронної пошти

WEBLATE_CONTACT_FORM

Налаштовує поведінку форми зв'язку, див. *CONTACT_FORM*.

WEBLATE_ALLOWED_HOSTS

Налаштовує дозволіні назви вузлів HTTP за допомогою *ALLOWED_HOSTS*.

Типове значення — *, що дозволяє використання будь-яких адрес.

Приклад:

```
environment:
  WEBLATE_ALLOWED_HOSTS: weblate.example.com, example.com
```


Дивись також:

[ALLOWED_HOSTS](#), *Налаштовування дозволених вузлів*, *Установіть належний домен сайта*

WEBLATE_REGISTRATION_OPEN

Налаштовує, чи відкрита реєстрація перемиканням [REGISTRATION_OPEN](#).

Приклад:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Налаштувати, які способи розпізнавання може бути використано для створення облікового запису за допомогою [REGISTRATION_ALLOW_BACKENDS](#).

Приклад:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
  WEBLATE_REGISTRATION_ALLOW_BACKENDS: azuread-oauth2,azuread-tenant-
  ↪oauth2
```

WEBLATE_REGISTRATION_REBIND

Нове в версії 4.16.

Налаштовує [REGISTRATION_REBIND](#).

WEBLATE_TIME_ZONE

Налаштовує використаний часовий пояс у Weblate, див. [TIME_ZONE](#).

Примітка: Щоб змінити часовий пояс самого контейнера Docker, скористайтесь змінною середовища TZ.

Приклад:

```
environment:
  WEBLATE_TIME_ZONE: Europe/Prague
```

WEBLATE_ENABLE_HTTPS

Вказує Weblate, що слід працювати за оберненим проксі HTTPS. У результаті Weblate використовує HTTPS у посиланнях електронної пошти та програмного інтерфейсу та встановлює прапорці захищеності у куках.

Підказка: Будь ласка, ознайомтеся із документацією до [ENABLE_HTTPS](#), щоб дізнатися більше про можливі проблеми.

Примітка: Це не призводить до того, що контейнер Weblate починає приймати з'єднання HTTPS. Прийняття таких з'єднань слід налаштувати окремо, див. приклади у розділі *Контейнер Docker із підтримкою HTTPS*.

Приклад:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Дивись також:

[ENABLE_HTTPS](#) *Установіть належний домен сайта*, [WEBLATE_SECURE_PROXY_SSL_HEADER](#)

WEBLATE_INTERLEDGER_PAYMENT_POINTERS

Нове в версії 4.12.1.

Надає Weblate змогу встановлювати поле `meta[name=monetization]` у заголовку документа. Якщо вказано декілька, значення буде вибрано випадковим чином.

Дивись також:

`INTERLEDGER_PAYMENT_POINTERS`

WEBLATE_IP_PROXY_HEADER

Дозволяє Weblate отримувати IP-адресу з будь-якого вказаного заголовка HTTP. Скористайтесь цим, якщо використовуєте обернений проксі перед контейнером Weblate.

Вмикає `IP_BEHIND_REVERSE_PROXY` і встановлює `IP_PROXY_HEADER`.

Примітка: Формат має відповідати очікуванням Django. Django [перетворює](#) прості назви заголовків HTTP ось так:

- перетворює усі символи до верхнього регістру
- замінює усі дефіси символами підкреслення
- дописує префікс HTTP_

Отже X-Forwarded-For буде пов'язано із HTTP_X_FORWARDED_FOR.

Приклад:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

Кортеж, який складається з заголовка HTTP і комбінації значень. Вказує на те, що запит є захищеним. Потрібен, якщо Weblate працює за оберненим проксі, який виконує обривання SSL без передавання стандартних заголовків HTTPS.

Приклад:

```
environment:
  WEBLATE_SECURE_PROXY_SSL_HEADER: HTTP_X_FORWARDED_PROTO,https
```

Дивись також:

`SECURE_PROXY_SSL_HEADER`

WEBLATE_REQUIRE_LOGIN

Вмикає `REQUIRE_LOGIN` для примусового розпізнавання в усьому Weblate.

Приклад:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

Додає винятки URL-адрес для розпізнавання, потрібного для всього встановлення Weblate, за допомогою `LOGIN_REQUIRED_URLS_EXCEPTIONS`.

Ви можете або замінити усі параметри, або змінити типове значення за допомогою змінних ADD і REMOVE.

WEBLATE_GOOGLE_ANALYTICS_ID

Налаштовує ідентифікатор для Google Analytics зміною `GOOGLE_ANALYTICS_ID`.

WEBLATE_GITHUB_USERNAME**WEBLATE_GITHUB_TOKEN****WEBLATE_GITHUB_HOST**

Налаштовує GitHub для інтеграції запитів щодо злиття Gitea зміною `GITHUB_CREDENTIALS`.

Дивись також:

Запити щодо злиття GitHub

WEBLATE_GITLAB_USERNAME**WEBLATE_GITLAB_TOKEN****WEBLATE_GITLAB_HOST**

Налаштовує GitLab для інтеграції запитів щодо об'єднання Pagure зміною `GITLAB_CREDENTIALS`.

Дивись також:

Запити щодо об'єднання GitLab

WEBLATE_GITEA_USERNAME**WEBLATE_GITEA_TOKEN****WEBLATE_GITEA_HOST**

Налаштовує Pagure для інтеграції запитів щодо злиття Gitea зміною `GITEA_CREDENTIALS`.

Дивись також:

Запити щодо злиття Gitea

WEBLATE_PAGURE_USERNAME**WEBLATE_PAGURE_TOKEN****WEBLATE_PAGURE_HOST**

Налаштовує Pagure для інтеграції запитів щодо об'єднання Pagure зміною `PAGURE_CREDENTIALS`.

Дивись також:

Запити щодо об'єднання Pagure

WEBLATE_BITBUCKETSERVER_USERNAME**WEBLATE_BITBUCKETSERVER_TOKEN****WEBLATE_BITBUCKETSERVER_HOST**

Налаштовує сервер Bitbucket для інтеграції запитів щодо злиття Gitea зміною `BITBUCKETSERVER_CREDENTIALS`.

Дивись також:

Запит щодо злиття для сервера Bitbucket

WEBLATE_DEFAULT_PULL_MESSAGE

Налаштовує типовий заголовок та повідомлення для запитів щодо об'єднання, які створено за допомогою програмного інтерфейсу за допомогою зміни `DEFAULT_PULL_MESSAGE`

Дивись також:

`DEFAULT_PULL_MESSAGE`

WEBLATE_SIMPLIFY_LANGUAGES

Налаштовує правила спрощення мов, див. [SIMPLIFY_LANGUAGES](#).

WEBLATE_DEFAULT_ACCESS_CONTROL

Налаштовує типове *Керування доступом* для нових проєктів, див. [DEFAULT_ACCESS_CONTROL](#).

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Налаштовує типове значення *Обмежений доступ* для нових складників, див. [DEFAULT_RESTRICTED_COMPONENT](#).

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Налаштовує типове значення *Дозволити поширення перекладу* для нових складників, див. [DEFAULT_TRANSLATION_PROPAGATION](#).

WEBLATE_DEFAULT_COMMITER_EMAIL

Налаштовує [DEFAULT_COMMITER_EMAIL](#).

WEBLATE_DEFAULT_COMMITER_NAME

Налаштовує [DEFAULT_COMMITER_NAME](#).

WEBLATE_DEFAULT_SHARED_TM

Налаштовує [DEFAULT_SHARED_TM](#).

WEBLATE_AKISMET_API_KEY

Налаштовує ключ до програмного інтерфейсу Akismet, див. [AKISMET_API_KEY](#).

WEBLATE_GPG_IDENTITY

Налаштовує підписування внесків за допомогою GPG, див. [WEBLATE_GPG_IDENTITY](#).

Дивись також:

Підписування внесків Git за допомогою GnuPG

WEBLATE_URL_PREFIX

Налаштовує префікс адреси, на якій запущено Weblate, див. [URL_PREFIX](#).

WEBLATE_SILENCED_SYSTEM_CHECKS

Налаштовує перевірки, підсумки яких ви не хочете бачити, див. [SILENCED_SYSTEM_CHECKS](#).

WEBLATE_CSP_SCRIPT_SRC

WEBLATE_CSP_IMG_SRC

WEBLATE_CSP_CONNECT_SRC

WEBLATE_CSP_STYLE_SRC

WEBLATE_CSP_FONT_SRC

Надає змогу налаштовувати заголовок HTTP Content-Security-Policy.

Дивись також:

Правила щодо безпеки даних, [CSP_SCRIPT_SRC](#), [CSP_IMG_SRC](#), [CSP_CONNECT_SRC](#), [CSP_STYLE_SRC](#), [CSP_FONT_SRC](#)

WEBLATE_LICENSE_FILTER

Налаштовує [LICENSE_FILTER](#).

WEBLATE_LICENSE_REQUIRED

Налаштовує [LICENSE_REQUIRED](#)

WEBLATE_WEBSITE_REQUIRED

Налаштовує [WEBSITE_REQUIRED](#)

WEBLATE_HIDE_VERSION

Налаштовує *HIDE_VERSION*.

WEBLATE_BASIC_LANGUAGES

Налаштовує *BASIC_LANGUAGES*.

WEBLATE_DEFAULT_AUTO_WATCH

Налаштовує *DEFAULT_AUTO_WATCH*.

WEBLATE_RATELIMIT_ATTEMPTS**WEBLATE_RATELIMIT_LOCKOUT****WEBLATE_RATELIMIT_WINDOW**

Нове в версії 4.6.

Налаштовує обмеження частоти.

Підказка: Ви можете встановити налаштування для будь-яких діапазонів обмежень частоти. Для цього додайте префікс *WEBLATE_* до будь-якого з параметрів, які описано у *Обмеження частоти*.

Дивись також:

Обмеження частоти, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*, *RATELIMIT_LOCKOUT*

WEBLATE_API_RATELIMIT_ANON**WEBLATE_API_RATELIMIT_USER**

Нове в версії 4.11.

Визначає обмеження на частоту використання програмного інтерфейсу. За типових налаштувань можна виконувати не більше 100 запитів на день для анонімних користувачів і не більше 5000 запитів на годину для користувачів, які пройшли розпізнавання у системі.

Дивись також:

Обмеження частоти у програмному інтерфейсі

WEBLATE_ENABLE_HOOKS

Нове в версії 4.13.

Налаштовує *ENABLE_HOOKS*.

WEBLATE_ENABLE_AVATARS

Нове в версії 4.6.1.

Налаштовує *ENABLE_AVATARS*.

WEBLATE_AVATAR_URL_PREFIX

Нове в версії 4.15.

Налаштовує *AVATAR_URL_PREFIX*.

WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Нове в версії 4.9.

Налаштовує *LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH*.

WEBLATE_SSH_EXTRA_ARGS

Нове в версії 4.9.

Налаштовує *SSH_EXTRA_ARGS*.

WEBLATE_BORG_EXTRA_ARGS

Нове в версії 4.9.

Налаштовує *BORG_EXTRA_ARGS*.

WEBLATE_ENABLE_SHARING

Нове в версії 4.14.1.

Налаштовує *ENABLE_SHARING*.

WEBLATE_EXTRA_HTML_HEAD

Нове в версії 4.15.

Налаштовує *EXTRA_HTML_HEAD*.

WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE

Нове в версії 4.15.

Налаштовує *PRIVATE_COMMIT_EMAIL_TEMPLATE*.

WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN

Нове в версії 4.15.

Налаштовує *PRIVATE_COMMIT_EMAIL_OPT_IN*.

WEBLATE_CORS_ALLOWED_ORIGINS

Нове в версії 4.16.

Уможливити запит CORS вказаного походження.

Приклад:

```
environment:  
  WEBLATE_CORS_ALLOWED_ORIGINS: https://example.com,https://weblate.org
```

CLIENT_MAX_BODY_SIZE

Нове в версії 4.16.3.

Configures maximal body size accepted by the built-in web server.

```
environment:  
  CLIENT_MAX_BODY_SIZE: 200m
```

Підказка: This variable intentionally lacks `WEBLATE_` prefix as it is shared with third-party container used in *Автоматична сертифікація SSL за допомогою Let's Encrypt*.

Параметри автоматичних пропозицій

Змінено в версії 4.13: У новій версії автоматичні пропозиції можна налаштувати за допомогою інтерфейсу користувача, див. *Налаштовування автоматичних пропозицій*.

Наявні змінні середовища буде імпортовано під час перенесення даних на Weblate 4.13, але внесення до них змін більше ні на що не впливатиме.

Параметри розпізнавання

LDAP

WEBLATE_AUTH_LDAP_SERVER_URI

WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE

WEBLATE_AUTH_LDAP_USER_ATTR_MAP

WEBLATE_AUTH_LDAP_BIND_DN

WEBLATE_AUTH_LDAP_BIND_PASSWORD

WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE

Шлях до файла, у якому міститься пароль для розпізнавання до сервера прив'язки LDAP.

Дивись також:

WEBLATE_AUTH_LDAP_BIND_PASSWORD

WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS

WEBLATE_AUTH_LDAP_USER_SEARCH

WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER

WEBLATE_AUTH_LDAP_USER_SEARCH_UNION

WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER

Налаштування розпізнавання за допомогою LDAP.

Приклад безпосередньої прив'язки:

```
environment:
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE: uid=%(user)s,ou=People,dc=example,dc=net
  # map weblate 'full_name' to ldap 'name' and weblate 'email' attribute to
  ↪ 'mail' ldap attribute.
  # another example that can be used with OpenLDAP: 'full_name:cn,email:mail'
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
```

Приклад пошуку і прив'язки:

```
environment:
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH: CN=Users,DC=example,DC=com
```

Приклад пошуку і прив'язки об'єднання:

```
environment:
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH_UNION: ou=users,dc=example,
  ↪ dc=com|ou=otherusers,dc=example,dc=com
```

Приклад із пошуком і прив'язкою до Active Directory:

```
environment:
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS: 0
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH: CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER: (sAMAccountName=%(user)s)
```

Дивись також:

Розпізнавання за допомогою LDAP

GitHub

```
WEBLATE_SOCIAL_AUTH_GITHUB_KEY
WEBLATE_SOCIAL_AUTH_GITHUB_SECRET
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID
```

Вмикає *Розпізнавання за GitHub*.

Bitbucket

```
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET
WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY
WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET
```

Вмикає *Розпізнавання за Bitbucket*.

Facebook

```
WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY
WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET
```

Вмикає *Facebook OAuth 2*.

Google

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS

Вмикає *Google OAuth 2*.

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY

WEBLATE_SOCIAL_AUTH_GITLAB_SECRET

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL

Вмикає *GitLab OAuth 2*.

Gitea

WEBLATE_SOCIAL_AUTH_GITEA_API_URL

WEBLATE_SOCIAL_AUTH_GITEA_KEY

WEBLATE_SOCIAL_AUTH_GITEA_SECRET

Вмикає розпізнавання за допомогою Gitea.

Azure Active Directory

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET

Вмикає розпізнавання за даними Active Directory в Azure, див. *Microsoft Azure Active Directory*.

Azure Active Directory із підтримкою Tenant

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID

Вмикає розпізнавання за даними Active Directory в Azure із підтримкою Tenant, див. *Microsoft Azure Active Directory*.

Keycloak

WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY

WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET

WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY

WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM

WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL

WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL

WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE

WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE

Вмикає розпізнавання за допомогою Keycloak, див. [документацію](#).

Постачальники Linux

Ви можете увімкнути розпізнавання за допомогою служб розпізнавання постачальників Linux, встановивши для вказаних нижче змінних довільне значення.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_OPENINFRA

WEBLATE_SOCIAL_AUTH_UBUNTU

Ідентифікатор

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Вмикає розпізнавання за допомогою Slack, див. [Ідентифікатор](#).

OpenID Connect

Нове в версії 4.13-1.

WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT

WEBLATE_SOCIAL_AUTH_OIDC_KEY

WEBLATE_SOCIAL_AUTH_OIDC_SECRET

WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY

Налаштовує типову інтеграцію з OpenID Connect.

Дивись також:

[OIDC \(OpenID Connect\)](#)

SAML

При першому запуску контейнера автоматично створюються самопідписані ключі SAML. Якщо ви хочете скористатися власними ключами, розмістіть сертифікат і закритий ключ у `/app/data/ssl/saml.crt` і `/app/data/ssl/saml.key`.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

WEBLATE_SAML_IDP_IMAGE

WEBLATE_SAML_IDP_TITLE

Параметри надавача профілів SAML, див. [Розпізнавання за SAML](#).

Інші параметри розпізнавання

WEBLATE_NO_EMAIL_AUTH

Вимикає розпізнавання за допомогою електронної пошти, якщо встановлено будь-яке значення. Див. [Вимикання розпізнавання за паролем](#).

Налаштування бази даних PostgreSQL

База даних створюється `docker-compose.yml`, отже ці параметри стосуються одразу контейнерів Weblate і PostgreSQL.

Дивись також:

[Налаштування бази даних для Weblate](#)

POSTGRES_PASSWORD

Пароль до PostgreSQL.

POSTGRES_PASSWORD_FILE

Шлях до файлу, який містить пароль до PostgreSQL. Можна скористатися як альтернативою до `POSTGRES_PASSWORD`.

POSTGRES_USER

Ім'я користувача PostgreSQL.

POSTGRES_DATABASE

Назва бази даних PostgreSQL.

POSTGRES_HOST

Назва вузла сервера PostgreSQL або його IP-адреса. Типове значення — `database`.

POSTGRES_PORT

Порт сервера PostgreSQL. Типового порту немає (використовується типове значення порту).

POSTGRES_SSL_MODE

Налаштовує спосіб обробки SSL у PostgreSQL при з'єднанні із сервером. Можливі варіанти можна знайти у розділі [Опис режимів SSL](#)

POSTGRES_ALTER_ROLE

Налаштовує назву ролі для зміни під час перенесення даних, див. [Налаштування Weblate на використання PostgreSQL](#).

POSTGRES_CONN_MAX_AGE

Нове в версії 4.8.1.

Тривалість життя з'єднання із базою даних у форматі цілого числа секунд. Скористайтесь значенням 0, щоб закривати з'єднання із базою даних наприкінці кожного запиту (це типова поведінка).

Вмикання відліку тривалості з'єднання типово спричиняє збільшення кількості відкритих з'єднань із базою даних. Будь ласка, скоригуйте налаштування бази даних, перш ніж вмикати цю можливість.

Приклад налаштувань:

```
environment:
  POSTGRES_CONN_MAX_AGE: 3600
```

Дивись також:

[CONN_MAX_AGE](#), [Persistent connections](#)

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

Нове в версії 4.9.1.

Вимкнути вказівники на боці сервера у базі даних. Це необхідно у деяких конфігураціях **pgbouncer**.

Приклад налаштувань:

```
environment:
  POSTGRES_DISABLE_SERVER_SIDE_CURSORS: 1
```

Дивись також:

[DISABLE_SERVER_SIDE_CURSORS](#), [Transaction pooling and server-side cursors](#)

Параметри резервного копіювання бази даних

Дивись також:

Дампи даних для резервних копій

WEBLATE_DATABASE_BACKUP

Налаштовує щоденне створення дампу бази даних за допомогою [DATABASE_BACKUP](#). Типовим значенням є `plain`.

Налаштування сервера кешування

Наполегливо рекомендуємо користуватися Redis. Вам слід надати екземпляр Redis, якщо ви запускаєте Weblate у Docker.

Дивись також:

Увімкніть кешування

REDIS_HOST

Назва вузла сервера Redis або IP-адреса. Типовою є `cache`.

REDIS_PORT

Порт сервера Redis. Типовим є 6379.

REDIS_DB

Номер бази даних Redis. Типовим є 1.

REDIS_PASSWORD

Пароль до сервера Redis. Типово, не використовується.

REDIS_PASSWORD_FILE

Шлях до файла, у якому міститься пароль для розпізнавання на сервері Redis.

Дивись також:

`REDIS_PASSWORD`

REDIS_TLS

Вмикає використання SSL для з'єднання Redis.

REDIS_VERIFY_SSL

Може бути використано для вимикання перевірки сертифікатів SSL для з'єднання Redis.

Налаштування сервера електронної пошти

Для забезпечення роботи з надсилання електронної пошти вам слід вказати поштовий сервер.

Приклад налаштувань TLS:

```
environment:
  WEBLATE_EMAIL_HOST: smtp.example.com
  WEBLATE_EMAIL_HOST_USER: user
  WEBLATE_EMAIL_HOST_PASSWORD: pass
```

Приклад налаштувань SSL:

```
environment:
  WEBLATE_EMAIL_HOST: smtp.example.com
  WEBLATE_EMAIL_PORT: 465
  WEBLATE_EMAIL_HOST_USER: user
  WEBLATE_EMAIL_HOST_PASSWORD: pass
  WEBLATE_EMAIL_USE_TLS: 0
  WEBLATE_EMAIL_USE_SSL: 1
```

Дивись також:

Налаштовування вихідної електронної пошти

WEBLATE_EMAIL_HOST

Назва вузла або IP-адреса поштового сервера.

Дивись також:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_SSL`, `WEBLATE_EMAIL_USE_TLS`,
`EMAIL_HOST`

WEBLATE_EMAIL_PORT

Порт поштового сервера. Типовим є 25.

Дивись також:

`EMAIL_PORT`

WEBLATE_EMAIL_HOST_USER

Користувач для розпізнавання для доступу до електронної пошти.

Дивись також:

`EMAIL_HOST_USER`

WEBLATE_EMAIL_HOST_PASSWORD

Пароль для розпізнавання для доступу до електронної пошти.

Дивись також:

`EMAIL_HOST_PASSWORD`

WEBLATE_EMAIL_HOST_PASSWORD_FILE

Шлях до файла, у якому міститься пароль для розпізнавання для доступу до електронної пошти.

Дивись також:

[WEBLATE_EMAIL_HOST_PASSWORD](#)

WEBLATE_EMAIL_USE_SSL

Визначає, чи слід використовувати неявне з'єднання TLS (захищене) при обміні даними із сервером SMTP. У більшій частині документації з електронної пошти цей тип з'єднання TLS називають SSL. Зазвичай, ним користуються на порту 465. Якщо у вас виникають проблеми, див. явне визначення TLS за допомогою [WEBLATE_EMAIL_USE_TLS](#).

Змінено в версії 4.11: Підтримка SSL/TLS вмикається автоматично на основі значення [WEBLATE_EMAIL_PORT](#).

Дивись також:

[WEBLATE_EMAIL_PORT](#), [WEBLATE_EMAIL_USE_TLS](#), [EMAIL_USE_SSL](#)

WEBLATE_EMAIL_USE_TLS

Визначає, чи слід використовувати з'єднання TLS (захищене) при обміні даними із сервером SMTP. Використовується для явних з'єднань TLS, зазвичай, на портах 587 і 25. Якщо у вас виникають проблеми із зависанням з'єднання, див. неявний параметр TLS [WEBLATE_EMAIL_USE_SSL](#).

Змінено в версії 4.11: Підтримка SSL/TLS вмикається автоматично на основі значення [WEBLATE_EMAIL_PORT](#).

Дивись також:

[WEBLATE_EMAIL_PORT](#), [WEBLATE_EMAIL_USE_SSL](#), [EMAIL_USE_TLS](#)

WEBLATE_EMAIL_BACKEND

Налаштовує модуль обробки Django на використання для надсилання повідомлень електронної пошти.

Дивись також:

[Налаштовування надсилання електронної пошти](#), [EMAIL_BACKEND](#)

WEBLATE_AUTO_UPDATE

Налаштовує, чи має Weblate оновлювати сховища і те, як це слід робити.

Дивись також:

[AUTO_UPDATE](#)

Примітка: Це булевий параметр (скористайтесь значенням "true" або "false").

Інтеграція сайта

WEBLATE_GET_HELP_URL

Налаштовує [GET_HELP_URL](#).

WEBLATE_STATUS_URL

Налаштовує [STATUS_URL](#).

WEBLATE_LEGAL_URL

Налаштовує [LEGAL_URL](#).

WEBLATE_PRIVACY_URL

Налаштовує [PRIVACY_URL](#).

Звітування про помилки

Рекомендуємо збирати помилки щодо встановленого екземпляра систематично, див. *Збирання звітів щодо помилок*.

Щоб увімкнути підтримку Rollbar, встановіть таке:

ROLLBAR_KEY

Ваш ключ доступу до сервера надсилання Rollbar.

ROLLBAR_ENVIRONMENT

Ваше середовище Rollbar, типове значення — `production`.

Щоб увімкнути підтримку Sentry, встановіть таке:

SENTRY_DSN

Ваш DSN Sentry.

SENTRY_ENVIRONMENT

Ваше середовище Sentry (необов'язково).

CDN локалізації

WEBLATE_LOCALIZE_CDN_URL

WEBLATE_LOCALIZE_CDN_PATH

Нове в версії 4.2.1.

Налаштування для *CDN локалізації JavaScript*.

`WEBLATE_LOCALIZE_CDN_PATH` є шляхом у контейнері. Репозиторій даних має перебувати на постійному томі, а не у тимчасовому репозиторії зберігання даних.

Одним із можливих варіантів є зберігання даних у каталозі даних Weblate:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

Примітка: Ви несете відповідальність за налаштування обслуговування файлів, згенерованих Weblate, файли зберігаються лише в налаштованому місці.

Дивись також:

weblate-cdn, `LOCALIZE_CDN_URL`, `LOCALIZE_CDN_PATH`

Зміна увімкнених застосунків, перевірок, додатків та автоматичних виправлень

Нове в версії 3.8-5.

Будовані налаштування увімкнених перевірок, додатків або автоматичних виправлень можна скоригувати за допомогою таких змінних:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Приклад:

```
environment:
  WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
  ↪ SameBookendingWhitespace
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon, customize.addons.OtherAddon
```

Дивись також:

CHECK_LIST, AUTOFIX_LIST, WEBLATE_ADDONS, INSTALLED_APPS

Параметри контейнерів

WEBLATE_WORKERS

Нове в версії 4.6.1.

Базова кількість процесів-обробників, які працюють у контейнері. Якщо не вказано, визначається автоматично при запуску контейнера на основі кількості доступних ядер процесора.

Використовується для визначення *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*, *CELERY_MEMORY_OPTIONS*, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*, *CELERY_BEAT_OPTIONS*, and *WEB_WORKERS*. Ви можете скористатися цими параметрами для коригування параметрів.

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

За допомогою цих змінних ви можете скоригувати параметри обробника Celery. Цим можна скористатися для коригування паралельної обробки (`--concurrency 16`) або використання іншої реалізації буфера (`--pool=gevent`).

Типово, кількість конкурентних обробників визначається на основі *WEBLATE_WORKERS*.

Приклад:

```
environment:
  CELERY_MAIN_OPTIONS: --concurrency 16
```

Дивись також:

Параметри обробника Celery, *Фонові завдання з використанням Celery*

WEB_WORKERS

Налаштувати кількість обробників uWSGI, які має бути виконано.

Типовим значенням є `WEBLATE_WORKERS`.

Приклад:

```
environment:
  WEB_WORKERS: 32
```

WEBLATE_SERVICE

Визначає, які служби слід запускати всередині контейнера. Скористайтесь цим для масштабування <docker-scaling.

Визначено такі перевірки:

celery-beat

Планувальник завдань Celery, має бути запущено лише один екземпляр. Цей контейнер також відповідає за перенесення структури бази даних, його має бути запущено перед усіма іншими.

celery-backup

Обробник Celery для резервного копіювання, має бути запущено лише один екземпляр.

celery-celery

Загальний обробник Celery.

celery-memory

Обробник пам'яті перекладів Celery.

celery-notify

Обробник сповіщень Celery.

celery-translate

Обробник автоматичних перекладів Celery.

web

Вебсервер.

Томи контейнера Docker

Існує два томи даних (дані і кеш), які експортуються контейнером Weblate. Інші контейнери служб (PostgreSQL або Redis) мають свої томи даних, але про них у цій документації не йтиметься.

Том даних використовується для зберігання сталих даних Weblate, зокрема клонованих репозиторіїв, або для налаштування встановленого Weblate.

Розташування тому Docker у основній системі залежить від ваших налаштувань Docker, але, зазвичай, він зберігається у `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/`. (шлях складається з назви вашого каталогу docker-compose, назви контейнера та назви тому). У контейнері його змонтовано до `/app/data`.

Том кешу монтується як `/app/cache` і використовується для зберігання статичних файлів і `CACHE_DIR`. Його вміст відтворюється при запуску контейнера, а тому може бути змонтовано з використанням тимчасової файлової системи, зокрема `tmpfs`.

Під час створення томів власноруч, каталоги повинні належати UID 1000, як користувачеві, що використовується всередині контейнера.

Дивись також:

[Документація з томів Docker](#)

Налаштовування поза змінними середовища

Доступ до більшості *параметрів налаштувань*, які пов'язано із встановленим Weblate, має здійснюватися за допомогою *змінних середовища Docker*.

Якщо вами буде виявлено параметр, доступу до якого немає зі змінної середовища, і ви вважаєте, що такий доступ має бути реалізовано, *попросіть про відкриття доступу у майбутніх версіях Weblate*.

Якщо вам потрібно змінити значення параметра, доступу до якого немає зі змінних середовища Docker, ви можете це зробити або за допомогою *тому даних*, або за допомогою *розширення образу Docker*.

Дивись також:

Налаштовування Weblate

Перевизначення параметрів з тому даних

Ви можете створити файл з адресою `/app/data/settings-override.py`, тобто у корені *тому даних*, з метою розширення або перевизначення параметрів, значення яких встановлено за допомогою змінних середовища.

Перевизначення параметрів за допомогою розширення образу Docker

Щоб перевизначити параметри на рівні образу Docker, а не з тому даних, виконайте такі дії:

1. *Створіть нетиповий пакунок Python*.
2. Додайте до вашого пакунка модуль, який імпортує усі параметри з `weblate.settings_docker`.

Наприклад, у структурі пакунка-прикладу, яку визначено у *Створення модуля Python*, ви можете створити файл `weblate_customization/weblate_customization/settings.py` із таким початковим кодом:

```
from weblate.settings_docker import *
```

3. Створіть нетиповий Dockerfile, який успадковуватиме офіційний образ Docker Weblate, а потім встановіть ваш пакунок і вкажіть для змінної середовища `DJANGO_SETTINGS_MODULE` ваш модуль параметрів:

```
FROM weblate/weblate

USER root

COPY weblate_customization /usr/src/weblate_customization
RUN pip install --no-cache-dir /usr/src/weblate_customization
ENV DJANGO_SETTINGS_MODULE=weblate_customization.settings

USER 1000
```

4. Замість використання офіційного образу Docker Weblate, зберіть нетиповий образ з цього файла Dockerfile.

Немає простого способу зробити це за допомогою `docker-compose.override.yml`. Можна було б додати `build: .` до вузла `weblate` у цьому файлі, але тоді б ваш нетиповий образ було б позначено як `weblate/weblate` у вашій системі, що призвело б до проблем.

Тому, замість використання `docker-compose.yml` безпосередньо з офіційного сховища, без змін, і його розширення за допомогою `docker-compose.override.yml`, вам варто створити копію офіційного файла `docker-compose.yml` і внести до неї зміни, замінивши `image: weblate/weblate` на `build: ..`

Див. [Compose file build reference](#), щоб дізнатися більше про збирання образів з початкового коду за допомогою `docker-compose`.

5. Розширте ваш модуль нетипових параметрів, визначивши або перевизначивши параметри.

Ви можете визначити параметри до або після інструкції з імпортування, наведеної вище, щоб визначити, які параметри мають пріоритет. Параметри, які визначено до інструкції з імпортування, може бути перевизначено змінними середовища та перевизначеннями параметрів у томі даних. Параметри, які визначено після інструкції з імпортування, не може бути перевизначено.

Ви можете піти і далі. Наприклад, ви можете відтворити деякі з речей, які [робить weblate](#). `docker_settings`, зокрема, надання доступу до параметрів через змінні середовища або уможливлення перевизначення параметрів з файлів Python у томі даних.

Заміна логотипа та інших статичних файлів

Нове в версії 3.8-5.

Статичні файли, які постачаються із Weblate, можна перевизначити розташуванням змінених варіантів у `/app/data/python/customize/static` (див. [Томи контейнера Docker](#)). Наприклад, шляхом створення `/app/data/python/customize/static/favicon.ico` можна замінити `favicon`.

Підказка: Файли копіюються до відповідного місця під час запуску контейнера, отже після внесення змін до вмісту тому потрібен перезапуск Weblate.

Цей підхід можна також використати для перевизначення шаблонів Weblate. Наприклад, документи [Правові відомості](#) можна розташувати у `/app/data/python/customize/templates/legal/documents`.

Крім того, ви можете включити власний модуль (див. [Налаштовування Weblate](#)) і додати його як окремий том до контейнера Docker. Приклад:

```
weblate:
  volumes:
    - weblate-data:/app/data
    - ./weblate_customization/weblate_customization:/app/data/python/weblate_
      ↪ customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

Налаштування сервера PostgreSQL

У контейнері для PostgreSQL використано типові налаштування PostgreSQL, він не зможе ефективно використати ваші ядра процесора або пам'ять. Для підвищення швидкодії рекомендуємо додатково налаштувати контейнер.

Налаштування можна скоригувати, як це описано у розділі *Database Configuration* на https://hub.docker.com/_/postgres. Налаштування, які відповідають вашому середовищу можна створити за допомогою <https://pgtune.leopard.in.ua/>.

Нутрощі контейнера

Контейнер використовує для запуску окремих служб **supervisor**. У випадку *Горизонтальне масштабування* ця програма запускає єдину службу у контейнері.

Для перевірки стану служб скористайтеся таким:

```
docker-compose exec --user weblate weblate supervisorctl status
```

Для кожної черги Celery передбачено окремі служби (щоб дізнатися більше, див. *Фонові завдання з використанням Celery*). Ви можете зупинити обробку певних завдань, зупинивши відповідний обробник:

```
docker-compose exec --user weblate weblate supervisorctl stop celery-translate
```

Установлення на Debian і Ubuntu

Вимоги щодо обладнання

Weblate має працювати на будь-якому сучасному обладнанні без проблем. Нижче наведено мінімальні налаштування, які потрібні для запуску Weblate на одному вузлі (Weblate, бази даних і вебсервера):

- 3 ГБ оперативної пам'яті
- 2 ядра процесора
- 1 ГБ вільного місця на диску

Чим більше пам'яті — тим краще. Пам'ять використовується для кешування на усіх рівнях (у файловій системі, базі даних та Weblate).

Збільшення кількості одночасних користувачів системи збільшує кількість потрібних для роботи ядер центрального процесора. Для сотень складників перекладу рекомендовано використовувати принаймні 4 ГБ оперативної пам'яті.

Типовий об'єм сховища даних близько 300 МБ на 1 мільйон збережених слів. Об'єм даних, потрібний для зберігання клонованих репозиторіїв коду, може бути різним, але Weblate намагається його скоротити створенням поверхневих клонів.

Примітка: Реальні вимоги до встановленого вами Weblate значно залежать від розміру перекладів, які ним керуються.

Установлення

Вимоги до системи

Установіть залежності, потрібні для збирання модулів Python (див. *Вимоги для програмного забезпечення*):

```
apt install -y \
  libxml2-dev libxslt-dev libfreetype6-dev libjpeg-dev libz-dev libyaml-dev \
  libffi-dev libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev \
  libacl1-dev libssl-dev libpq-dev libjpeg-dev build-essential \
  python3-gdbm python3-dev python3-pip python3-virtualenv virtualenv git
```

Установіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (див. *Необов'язкові залежності*):

```
apt install -y \
  tesseract-ocr libtesseract-dev libleptonica-dev \
  libldap2-dev libldap-common libsasl2-dev \
  libxmlsec1-dev
```

Додатково встановіть програмне забезпечення для роботи промислового сервера, див. [Запуск сервера](#), [Налаштування бази даних для Weblate](#), [Фонові завдання з використанням Celery](#). Залежно від розміру встановлених складників, вам може знадобитися запуск цих складників на окремих серверах.

Настанови із локального встановлення:

```
# Web server option 1: NGINX and uWSGI
apt install -y nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with `mod_wsgi`
apt install -y apache2 libapache2-mod-wsgi-py3

# Caching backend: Redis
apt install -y redis-server

# Database server: PostgreSQL
apt install -y postgresql postgresql-contrib

# SMTP server
apt install -y exim4
```

Модулі Python

Підказка: Ми використовуємо `virtualenv` для встановлення Weblate в окремому середовищі з вашої системи. Якщо ви не при звичаєні до такого встановлення, ознайомтеся із [User Guide](#) до `virtualenv`.

1. Створіть `virtualenv` для Weblate:

```
virtualenv ~/weblate-env
```

2. Активуйте `virtualenv` для Weblate:

```
. ~/weblate-env/bin/activate
```

3. Установіть Weblate разом із усіма необов'язковими залежностями:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Будь ласка, ознайомтеся із розділом щодо [Необов'язкові залежності](#), щоб дізнатися про коригування необов'язкових залежностей.

Примітка: У деяких дистрибутивах Linux спроба запуску Weblate завершується повідомленням про помилку, яку пов'язано із `libffi`:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

Причиною є несумісність двійкових пакунків, які поширюють за допомогою `PyPI`, із дистрибутивом. Щоб усунути причину помилки, вам слід повторно зібрати пакунок у вашій системі:

```
pip install --force-reinstall --no-binary :all: cffi
```

Налаштовування Weblate

Примітка: Далі ми припускаємо, що віртуальне середовище, яке використовується Weblate, активоване (шляхом виконання `./~/weblate-env/bin/activate`). Якщо ні, вкажіть повний шлях до команди **weblate** як `~/weblate-env/bin/weblate`.

1. Скопіюйте файл `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` до `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Скоригуйте значення у новому файлі `settings.py` відповідно до ваших уподобань. Вам доведеться надати принаймні реєстраційні дані для доступу до бази даних та закритий ключ Django, а також вам варто внести додаткові зміни для вашої промислової конфігурації, див. [Коригування налаштувань](#).
3. Створіть базу даних і її структуру для Weblate (у прикладі вказано параметри для PostgreSQL, ознайомтеся з розділом [check](#) [Налаштування бази даних для Weblate](#), щоб дізнатися більше про готову для робочого середовища конфігурацію):

```
weblate migrate
```

4. Створіть обліковий запис адміністративного користувача і скопіюйте його пароль до буфера обміну, а також збережіть його для користування в подальшому:

```
weblate createadmin
```

5. Зберіть статичні файли для свого вебсервера (див. [Запуск сервера](#) і [Обслуговування статичних файлів](#)):

```
weblate collectstatic
```

6. Стисніть файли JavaScript і CSS (необов'язково, див. [Стиснення клієнтських даних](#)):

```
weblate compress
```

7. Запустіть обробники Celery. У цьому немає потреби для розробки, але ми наполегливо радимо це у всіх інших варіантах. [Фонові завдання з використанням Celery](#), містить більше інформації:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Запустіть сервер для розробки (докладніше про налаштування робочого середовища: [Запуск сервера](#)):

```
weblate runserver
```

Після встановлення

Вітаємо, тепер ваш сервер Weblate запущено, і ви можете розпочати ним користуватися.

- Ви можете отримувати доступ до Weblate за адресою `http://localhost:8000/`.
- Увійдіть до системи за допомогою реєстраційних даних, які було отримано під час встановлення, або зареєструйте нових користувачів.
- Тепер ви можете віддавати команди Weblate за допомогою **weblate**, коли віртуальне середовище Weblate є активним, див. [Команди керування](#).

- Припинити роботу тестового сервера можна за допомогою натискання комбінації клавіш Ctrl+C.
- Ознайомтеся із переліком можливих проблем встановлення або за URL-адресою `/manage/performance/` (див. *Інтерфейс керування*), або за допомогою команди **weblate check --deploy**, див. *Промислові налаштування*.

Додавання перекладу

1. Відкрийте адміністративний інтерфейс (`http://localhost:8000/create/project/`) і створіть проєкт, який ви хочете перекладати. Див. *Налаштування проєкту*, щоб дізнатися більше.

Усе, що вам слід тут вказати, це назва проєкту і адреса його сайту.

2. Створіть складник, який є справжнім об'єктом для перекладу — він вказує на репозиторій системи керування версіями і визначає, які файли слід перекладати. Див. *Налаштовування складників*, щоб дізнатися більше.

Важливими полями тут є такі: *Назва складника*, *Репозиторій з джерелами* та *Маска файлів* для пошуку придатних до перекладу файлів. У Weblate передбачено підтримку широкого діапазону форматів, зокрема *GNU Gettext*, *Рядкові ресурси Android*, *Рядки Apple iOS*, *Властивості Java*, *Формат Stringsdict* та *Формат Fluent*. Див. *Підтримувані формати файлів*, щоб дізнатися більше.

3. Щойно буде виконано вказані вище дії (це може бути доволі тривалим процесом, залежно від розмірів вашого репозиторію системи керування версіями та кількості повідомлень, які слід перекласти), ви можете розпочати переклад.

Установлення у SUSE та openSUSE

Вимоги щодо обладнання

Weblate має працювати на будь-якому сучасному обладнанні без проблем. Нижче наведено мінімальні налаштування, які потрібні для запуску Weblate на одному вузлі (Weblate, бази даних і вебсервера):

- 3 ГБ оперативної пам'яті
- 2 ядра процесора
- 1 ГБ вільного місця на диску

Чим більше пам'яті — тим краще. Пам'ять використовується для кешування на усіх рівнях (у файловій системі, базі даних та Weblate).

Збільшення кількості одночасних користувачів системи збільшує кількість потрібних для роботи ядер центрального процесора. Для сотень складників перекладу рекомендовано використовувати принаймні 4 ГБ оперативної пам'яті.

Типовий об'єм сховища даних близько 300 МБ на 1 мільйон збережених слів. Об'єм даних, потрібний для зберігання клонованих репозиторіїв коду, може бути різним, але Weblate намагається його скоротити створенням поверхневих клонів.

Примітка: Реальні вимоги до встановленого вами Weblate значно залежать від розміру перекладів, які ним керуються.

Установлення

Вимоги до системи

Установіть залежності, потрібні для збирання модулів Python (див. *Вимоги для програмного забезпечення*):

```
zypper install \
  libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
  libyaml-devel libffi-devel cairo-devel pango-devel \
  gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
  python3-devel git
```

Установіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (див. *Необов'язкові залежності*):

```
zypper install tesseract-ocr tesseract-devel leptonica-devel
zypper install libldap2-devel libsasl2-devel
zypper install libxmlsec1-devel
```

Додатково встановіть програмне забезпечення для роботи промислового сервера, див. *Запуск сервера, Налаштування бази даних для Weblate, Фонові завдання з використанням Celery*. Залежно від розміру встановлених складників, вам може знадобитися запуск цих складників на окремих серверах.

Настанови із локального встановлення:

```
# Web server option 1: NGINX and uWSGI
zypper install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
zypper install apache2 apache2-mod_wsgi

# Caching backend: Redis
zypper install redis-server

# Database server: PostgreSQL
zypper install postgresql postgresql-contrib

# SMTP server
zypper install postfix
```

Модулі Python

Підказка: Ми використовуємо `virtualenv` для встановлення Weblate в окремому середовищі з вашої системи. Якщо ви не при звичаєні до такого встановлення, ознайомтеся із [User Guide](#) до `virtualenv`.

1. Створіть `virtualenv` для Weblate:

```
virtualenv ~/weblate-env
```

2. Активуйте `virtualenv` для Weblate:

```
. ~/weblate-env/bin/activate
```

3. Установіть Weblate разом із усіма обов'язковими залежностями:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```


Будь ласка, ознайомтеся із розділом щодо *Необов'язкові залежності*, щоб дізнатися про коригування необов'язкових залежностей.

Примітка: У деяких дистрибутивах Linux спроба запуску Weblate завершується повідомленням про помилку, яку пов'язано із libffi:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

Причиною є несумісність двійкових пакунків, які поширюють за допомогою PyPI, із дистрибутивом. Щоб усунути причину помилки, вам слід повторно зібрати пакунок у вашій системі:

```
pip install --force-reinstall --no-binary :all: cffi
```

Налаштовування Weblate

Примітка: Далі ми припускаємо, що віртуальне середовище, яке використовується Weblate, активоване (шляхом виконання `./~/weblate-env/bin/activate`). Якщо ні, вкажіть повний шлях до команди **weblate** як `~/weblate-env/bin/weblate`.

1. Скопіюйте файл `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` до `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Скоригуйте значення у новому файлі `settings.py` відповідно до ваших уподобань. Вам доведеться надати принаймні реєстраційні дані для доступу до бази даних та закритий ключ Django, а також вам варто внести додаткові зміни для вашої промислової конфігурації, див. *Коригування налаштувань*.
3. Створіть базу даних і її структуру для Weblate (у прикладі вказано параметри для PostgreSQL, ознайомтеся з розділом *check* *Налаштування бази даних для Weblate*, щоб дізнатися більше про готову для робочого середовища конфігурацію):

```
weblate migrate
```

4. Створіть обліковий запис адміністративного користувача і скопіюйте його пароль до буфера обміну, а також збережіть його для користування в подальшому:

```
weblate createadmin
```

5. Зберіть статичні файли для свого вебсервера (див. *Запуск сервера* і *Обслуговування статичних файлів*):

```
weblate collectstatic
```

6. Стисніть файли JavaScript і CSS (необов'язково, див. *Стиснення клієнтських даних*):

```
weblate compress
```

7. Запустіть обробники Celery. У цьому немає потреби для розробки, але ми наполегливо радимо це у всіх інших варіантах. *Фонові завдання з використанням Celery*, містить більше інформації:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Запустіть сервер для розробки (докладніше про налаштування робочого середовища: *Запуск сервера*):

```
weblate runserver
```

Після встановлення

Вітаємо, тепер ваш сервер Weblate запущено, і ви можете розпочати ним користуватися.

- Ви можете отримувати доступ до Weblate за адресою `http://localhost:8000/`.
- Увійдіть до системи за допомогою реєстраційних даних, які було отримано під час встановлення, або зареєструйте нових користувачів.
- Тепер ви можете віддавати команди Weblate за допомогою **weblate**, коли віртуальне середовище Weblate є активним, див. *Команди керування*.
- Припинити роботу тестового сервера можна за допомогою натискання комбінації клавіш Ctrl+C.
- Ознайомтеся із переліком можливих проблем встановлення або за URL-адресою `/manage/performance/` (див. *Інтерфейс керування*), або за допомогою команди **weblate check --deploy**, див. *Промислові налаштування*.

Додавання перекладу

1. Відкрийте адміністративний інтерфейс (`http://localhost:8000/create/project/`) і створіть проєкт, який ви хочете перекладати. Див. *Налаштування проєкту*, щоб дізнатися більше.

Усе, що вам слід тут вказати, це назва проєкту і адреса його сайту.

2. Створіть складник, який є справжнім об'єктом для перекладу — він вказує на репозиторій системи керування версіями і визначає, які файли слід перекладати. Див. *Налаштування складників*, щоб дізнатися більше.

Важливими полями тут є такі: *Назва складника*, *Репозиторій з джерелами* та *Маска файлів* для пошуку придатних до перекладу файлів. У Weblate передбачено підтримку широкого діапазону форматів, зокрема *GNU Gettext*, *Рядкові ресурси Android*, *Рядки Apple iOS*, *Властивості Java*, *Формат Stringsdict* та *Формат Fluent*. Див. *Підтримувані формати файлів*, щоб дізнатися більше.

3. Щойно буде виконано вказані вище дії (це може бути доволі тривалим процесом, залежно від розмірів вашого репозиторію системи керування версіями та кількості повідомлень, які слід перекласти), ви можете розпочати переклад.

Установлення у RedHat, Fedora і CentOS

Вимоги щодо обладнання

Weblate має працювати на будь-якому сучасному обладнанні без проблем. Нижче наведено мінімальні налаштування, які потрібні для запуску Weblate на одному вузлі (Weblate, бази даних і вебсервера):

- 3 ГБ оперативної пам'яті
- 2 ядра процесора
- 1 ГБ вільного місця на диску

Чим більше пам'яті — тим краще. Пам'ять використовується для кешування на усіх рівнях (у файловій системі, базі даних та Weblate).

Збільшення кількості одночасних користувачів системи збільшує кількість потрібних для роботи ядер центрального процесора. Для сотень складників перекладу рекомендовано використовувати принаймні 4 ГБ оперативної пам'яті.

Типовий об'єм сховища даних близько 300 МБ на 1 мільйон збережених слів. Об'єм даних, потрібний для зберігання клонованих репозиторіїв коду, може бути різним, але Weblate намагається його скоротити створенням поверхневих клонів.

Примітка: Реальні вимоги до встановленого вами Weblate значно залежать від розміру перекладів, які ним керуються.

Установлення

Вимоги до системи

Установіть залежності, потрібні для збирання модулів Python (див. *Вимоги для програмного забезпечення*):

```
dnf install \
    libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
    libyaml-devel libffi-devel cairo-devel pango-devel \
    gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
    python3-devel git
```

Установіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (див. *Необов'язкові залежності*):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
dnf install libldap2-devel libsasl2-devel
dnf install libxmlsec1-devel
```

Додатково встановіть програмне забезпечення для роботи промислового сервера, див. *Запуск сервера, Налаштування бази даних для Weblate, Фонові завдання з використанням Celery*. Залежно від розміру встановлених складників, вам може знадобитися запуск цих складників на окремих серверах.

Настанови із локального встановлення:

```
# Web server option 1: NGINX and uWSGI
dnf install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
dnf install apache2 apache2-mod_wsgi

# Caching backend: Redis
dnf install redis

# Database server: PostgreSQL
dnf install postgresql postgresql-contrib

# SMTP server
dnf install postfix
```

Модулі Python

Підказка: Ми використовуємо `virtualenv` для встановлення Weblate в окремому середовищі з вашої системи. Якщо ви не при звичаєні до такого встановлення, ознайомтеся із [User Guide](#) до `virtualenv`.

1. Створіть `virtualenv` для Weblate:

```
virtualenv ~/weblate-env
```

2. Активуйте `virtualenv` для Weblate:

```
. ~/weblate-env/bin/activate
```

3. Установіть Weblate разом із усіма необов'язковими залежностями:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Будь ласка, ознайомтеся із розділом щодо *Необов'язкові залежності*, щоб дізнатися про коригування необов'язкових залежностей.

Примітка: У деяких дистрибутивах Linux спроба запуску Weblate завершується повідомленням про помилку, яку пов'язано із `libffi`:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi_
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

Причиною є несумісність двійкових пакунків, які поширюють за допомогою PyPI, із дистрибутивом. Щоб усунути причину помилки, вам слід повторно зібрати пакунок у вашій системі:

```
pip install --force-reinstall --no-binary :all: cffi
```

Налаштовування Weblate

Примітка: Далі ми припускаємо, що віртуальне середовище, яке використовується Weblate, активоване (шляхом виконання `. ~/weblate-env/bin/activate`). Якщо ні, вкажіть повний шлях до команди **weblate** як `~/weblate-env/bin/weblate`.

1. Скопіюйте файл `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` до `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Скоригуйте значення у новому файлі `settings.py` відповідно до ваших уподобань. Вам доведеться надати принаймні реєстраційні дані для доступу до бази даних та закритий ключ Django, а також вам варто внести додаткові зміни для вашої промислової конфігурації, див. *Коригування налаштувань*.
3. Створіть базу даних і її структуру для Weblate (у прикладі вказано параметри для PostgreSQL, ознайомтеся з розділом *check* *Налаштування бази даних для Weblate*, щоб дізнатися більше про готову для робочого середовища конфігурацію):

```
weblate migrate
```

- Створіть обліковий запис адміністративного користувача і скопіюйте його пароль до буфера обміну, а також збережіть його для користування в подальшому:

```
weblate createadmin
```

- Зберіть статичні файли для свого вебсервера (див. *Запуск сервера і Обслуговування статичних файлів*):

```
weblate collectstatic
```

- Стисніть файли JavaScript і CSS (необов'язково, див. *Стиснення клієнтських даних*):

```
weblate compress
```

- Запустіть обробники Celery. У цьому немає потреби для розробки, але ми наполегливо радимо це у всіх інших варіантах. *Фонові завдання з використанням Celery*, містить більше інформації:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

- Запустіть сервер для розробки (докладніше про налаштування робочого середовища: *Запуск сервера*):

```
weblate runserver
```

Після встановлення

Вітаємо, тепер ваш сервер Weblate запущено, і ви можете розпочати ним користуватися.

- Ви можете отримувати доступ до Weblate за адресою `http://localhost:8000/`.
- Увійдіть до системи за допомогою реєстраційних даних, які було отримано під час встановлення, або зареєструйте нових користувачів.
- Тепер ви можете віддавати команди Weblate за допомогою **weblate**, коли віртуальне середовище Weblate є активним, див. *Команди керування*.
- Припинити роботу тестового сервера можна за допомогою натискання комбінації клавіш Ctrl+C.
- Ознайомтеся із переліком можливих проблем встановлення або за URL-адресою `/manage/performance/` (див. *Інтерфейс керування*), або за допомогою команди **weblate check --deploy**, див. *Промислові налаштування*.

Додавання перекладу

- Відкрийте адміністративний інтерфейс (`http://localhost:8000/create/project/`) і створіть проєкт, який ви хочете перекладати. Див. *Налаштування проєкту*, щоб дізнатися більше.

Усе, що вам слід тут вказати, це назва проєкту і адреса його сайту.

- Створіть складник, який є справжнім об'єктом для перекладу — він вказує на репозиторій системи керування версіями і визначає, які файли слід перекладати. Див. *Налаштування складників*, щоб дізнатися більше.

Важливими полями тут є такі: *Назва складника*, *Репозиторій з джерелами* та *Маска файлів* для пошуку придатних до перекладу файлів. У Weblate передбачено підтримку широкого діапазону форматів, зокрема *GNU Gettext*, *Рядкові ресурси Android*, *Рядки Apple iOS*, *Властивості Java*, *Формат Stringsdict* та *Формат Fluent*. Див. *Підтримувані формати файлів*, щоб дізнатися більше.

- Щойно буде виконано вказані вище дії (це може бути доволі тривалим процесом, залежно від розмірів вашого репозиторію системи керування версіями та кількості повідомлень, які слід перекласти), ви можете розпочати переклад.

Установлення на macOS

Вимоги щодо обладнання

Weblate має працювати на будь-якому сучасному обладнанні без проблем. Нижче наведено мінімальні налаштування, які потрібні для запуску Weblate на одному вузлі (Weblate, бази даних і вебсервера):

- 3 ГБ оперативної пам'яті
- 2 ядра процесора
- 1 ГБ вільного місця на диску

Чим більше пам'яті — тим краще. Пам'ять використовується для кешування на усіх рівнях (у файловій системі, базі даних та Weblate).

Збільшення кількості одночасних користувачів системи збільшує кількість потрібних для роботи ядер центрального процесора. Для сотень складників перекладу рекомендовано використовувати принаймні 4 ГБ оперативної пам'яті.

Типовий об'єм сховища даних близько 300 МБ на 1 мільйон збережених слів. Об'єм даних, потрібний для зберігання клонованих репозиторіїв коду, може бути різним, але Weblate намагається його скоротити створенням поверхневих клонів.

Примітка: Реальні вимоги до встановленого вами Weblate значно залежать від розміру перекладів, які ним керуються.

Установлення

Вимоги до системи

Установіть залежності, потрібні для збирання модулів Python (див. *Вимоги для програмного забезпечення*):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
pip install virtualenv
```

Переконайтеся, що рір зможе знайти версії libffi і openssl, яка надається homebrew — це потрібно буде на кроці збирання встановленої версії.

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig:/usr/local/opt/
↳openssl@3/lib/pkgconfig"
```

Установіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (див. *Необов'язкові залежності*):

```
brew install tesseract
```

Додатково встановіть програмне забезпечення для роботи промислового сервера, див. *Запуск сервера, Налаштування бази даних для Weblate, Фонові завдання з використанням Celery*. Залежно від розміру встановлених складників, вам може знадобитися запуск цих складників на окремих серверах.

Настанови із локального встановлення:

```
# Web server option 1: NGINX and uWSGI
brew install nginx uwsgi

# Web server option 2: Apache with `mod_wsgi`
brew install httpd
```

(continues on next page)

(continued from previous page)

```
# Caching backend: Redis
brew install redis

# Database server: PostgreSQL
brew install postgresql
```

Модулі Python

Підказка: Ми використовуємо `virtualenv` для встановлення Weblate в окремому середовищі з вашої системи. Якщо ви не при звичаєні до такого встановлення, ознайомтеся із [User Guide](#) до `virtualenv`.

1. Створіть `virtualenv` для Weblate:

```
virtualenv ~/weblate-env
```

2. Активуйте `virtualenv` для Weblate:

```
. ~/weblate-env/bin/activate
```

3. Установіть Weblate разом із усіма необов'язковими залежностями:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Будь ласка, ознайомтеся із розділом щодо *Необов'язкові залежності*, щоб дізнатися про коригування необов'язкових залежностей.

Примітка: У деяких дистрибутивах Linux спроба запуску Weblate завершується повідомленням про помилку, яку пов'язано із `libffi`:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

Причиною є несумісність двійкових пакунків, які поширюють за допомогою PyPI, із дистрибутивом. Щоб усунути причину помилки, вам слід повторно зібрати пакунок у вашій системі:

```
pip install --force-reinstall --no-binary :all: cffi
```

Налаштовування Weblate

Примітка: Далі ми припускаємо, що віртуальне середовище, яке використовується Weblate, активоване (шляхом виконання `. ~/weblate-env/bin/activate`). Якщо ні, вкажіть повний шлях до команди **weblate** як `~/weblate-env/bin/weblate`.

1. Скопіюйте файл `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` до `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Скоригуйте значення у новому файлі `settings.py` відповідно до ваших уподобань. Вам доведеться надати принаймні реєстраційні дані для доступу до бази даних та закритий ключ Django, а також вам варто внести додаткові зміни для вашої промислової конфігурації, див. *Коригування налаштувань*.

- Створіть базу даних і її структуру для Weblate (у прикладі вказано параметри для PostgreSQL, ознайомтеся з розділом [check налаштування бази даних для Weblate](#), щоб дізнатися більше про готову для робочого середовища конфігурацію):

```
weblate migrate
```

- Створіть обліковий запис адміністративного користувача і скопіюйте його пароль до буфера обміну, а також збережіть його для користування в подальшому:

```
weblate createadmin
```

- Зберіть статичні файли для свого вебсервера (див. [Запуск сервера](#) і [Обслуговування статичних файлів](#)):

```
weblate collectstatic
```

- Стисніть файли JavaScript і CSS (необов'язково, див. [Стиснення клієнтських даних](#)):

```
weblate compress
```

- Запустіть обробники Celery. У цьому немає потреби для розробки, але ми наполегливо радимо це у всіх інших варіантах. [Фонові завдання з використанням Celery](#), містить більше інформації:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

- Запустіть сервер для розробки (докладніше про налаштування робочого середовища: [Запуск сервера](#)):

```
weblate runserver
```

Після встановлення

Вітаємо, тепер ваш сервер Weblate запущено, і ви можете розпочати ним користуватися.

- Ви можете отримувати доступ до Weblate за адресою `http://localhost:8000/`.
- Увійдіть до системи за допомогою реєстраційних даних, які було отримано під час встановлення, або зареєструйте нових користувачів.
- Тепер ви можете віддавати команди Weblate за допомогою **weblate**, коли віртуальне середовище Weblate є активним, див. [Команди керування](#).
- Припинити роботу тестового сервера можна за допомогою натискання комбінації клавіш Ctrl+C.
- Ознайомтеся із переліком можливих проблем встановлення або за URL-адресою `/manage/performance/` (див. [Інтерфейс керування](#)), або за допомогою команди **weblate check --deploy**, див. [Промислові налаштування](#).

Додавання перекладу

- Відкрийте адміністративний інтерфейс (`http://localhost:8000/create/project/`) і створіть проєкт, який ви хочете перекладати. Див. [Налаштування проєкту](#), щоб дізнатися більше.

Усе, що вам слід тут вказати, це назва проєкту і адреса його сайту.

- Створіть складник, який є справжнім об'єктом для перекладу — він вказує на репозиторій системи керування версіями і визначає, які файли слід перекладати. Див. [Налаштовування складників](#), щоб дізнатися більше.

Важливими полями тут є такі: [Назва складника](#), [Репозиторій з джерелами](#) та [Маска файлів](#) для пошуку придатних до перекладу файлів. У Weblate передбачено підтримку широкого діапазону форматів, зокрема [GNU Gettext](#), [Рядкові ресурси Android](#), [Рядки Apple iOS](#), [Властивості Java](#), [Формат Stringsdict](#) та [Формат Fluent](#). Див. [Підтримувані формати файлів](#), щоб дізнатися більше.

3. Щойно буде виконано вказані вище дії (це може бути доволі тривалим процесом, залежно від розмірів вашого репозиторію системи керування версіями та кількості повідомлень, які слід перекласти), ви можете розпочати переклад.

Установлення з початкового коду

1. Будь ласка, для встановлення Weblate спочатку виконайте настанови зі встановлення для вашої системи:

- *Установлення на Debian і Ubuntu*
- *Установлення у SUSE та openSUSE*
- *Установлення у RedHat, Fedora і CentOS*

2. Отримайте найсвіжіший код Weblate за допомогою Git (або отримайте архів з кодом і розпакуйте його):

```
git clone https://github.com/WeblateOrg/weblate.git weblate-src
```

Крім того, ви можете скористатися архівами випущених версій. Ви можете отримати їх з нашого сайту <<https://weblate.org/>>. Ці архіви є криптографічно підписаними. Див. *Перевірка підписів випуску*.

3. Установіть поточний код Weblate до virtualenv:

```
. ~/weblate-env/bin/activate
pip install -e weblate-src
```

4. Скопіюйте weblate/settings_example.py до weblate/settings.py.
5. Скорируйте значення у новому файлі settings.py відповідно до ваших уподобань. Вам доведеться надати принаймні реєстраційні дані для доступу до бази даних та закритий ключ Django, а також вам варто внести додаткові зміни для вашої промислової конфігурації, див. *Коригування налаштувань*.
6. Створіть базу даних, яка використовуватиметься Weblate, див. *Налаштування бази даних для Weblate*.
7. Побудуйте таблиці Django, статичні файли та початкові дані (див. *Заповнення бази даних і Обслуговування статичних файлів*):

```
weblate migrate
weblate collectstatic
weblate compress
```

Примітка: Цей крок слід повторювати кожного разу, коли ви оновлюватимете репозиторій.

Установлення на OpenShift

Якщо Weblate за допомогою шаблону OpenShift, ви зможете отримати ваш працездатний і запущений особистий екземпляр Weblate за декілька секунд. Усі залежності Weblate вже включено. Типовим засобом роботи із базою даних у таких екземплярах є PostgreSQL, також буде використано сталі запити щодо томів.

Шаблон можна знайти тут: <<https://github.com/WeblateOrg/openshift/>>.

Установлення

У наведених нижче прикладах припускається, що у вас є працездатне середовище OpenShift v3.x із встановленим клієнтом `oc`. Будь ласка, ознайомтеся із документацією до OpenShift, щоб отримати додаткові настанови.

Файл `template.yml` є придатним для роботи із усіма складниками в OpenShift. Також передбачено варіант `template-external-postgresql.yml`, який не запускає сервер PostgreSQL і надає змогу налаштувати зовнішній сервер PostgreSQL.

Вебконсоль

Скопіюйте вміст файлу `template.yml` та імпортуйте його до вашого проєкту, а потім скористайтеся кнопкою `Create` у вебконсолі OpenShift для створення вашого застосунку. У вебконсолі програма попросить вас надати значення для всіх параметрів, які використано у шаблоні.

Командний рядок

Щоб вивантажити шаблон Weblate до бібліотеки шаблонів вашого поточного проєкту, вкажіть файл `template.yml` у такій команді:

```
$ oc create -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/  
↪template.yml \  
-n <PROJECT>
```

Після цього шаблон стане доступним для вибору за допомогою вебконсолі або інтерфейсу командного рядка.

Параметри

Параметри, значення яких ви можете перевизначити, зазначено у розділі параметрів шаблону. Ви можете вивести цей список у командному рядку за допомогою вказаної нижче команди, у якій слід зазначити файл, який буде використано:

```
$ oc process --parameters -f https://raw.githubusercontent.com/WeblateOrg/  
↪openshift/main/template.yml  
  
# If the template is already uploaded  
$ oc process --parameters -n <PROJECT> weblate
```

Попередня обробка

Ви також можете скористатися інтерфейсом командного рядка для обробки шаблонів та скористатися налаштуваннями, які буде створено, для негайного створення об'єктів.

```
$ oc process -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/  
↪template.yml \  
-p APPLICATION_NAME=weblate \  
-p WEBLATE_VERSION=4.3.1-1 \  
-p WEBLATE_SITE_DOMAIN=weblate.app-openshift.example.com \  
-p POSTGRESQL_IMAGE=docker-registry.default.svc:5000/openshift/postgresql:9.6 \  
-p REDIS_IMAGE=docker-registry.default.svc:5000/openshift/redis:3.2 \  
| oc create -f
```

Екземпляр Weblate стане доступним після успішного перенесення та розгортання у домені, вказаному за допомогою параметра `WEBLATE_SITE_DOMAIN`.

Після налаштування контейнера ви можете увійти до системи від імені користувача *admin* із використанням пароля, який надається у змінній `WEBLATE_ADMIN_PASSWORD`, або випадкового пароля, який створено під час першого запуску, якщо пароль не було встановлено за допомогою змінної.

Щоб скинути пароль *admin*, перезапустіть контейнер зі змінною `WEBLATE_ADMIN_PASSWORD`, значенням якої є новий пароль у відповідному `Secret`.

Очищення

```
$ oc delete all -l app=<APPLICATION_NAME>
$ oc delete configmap -l app= <APPLICATION_NAME>
$ oc delete secret -l app=<APPLICATION_NAME>
# ATTENTION! The following command is only optional and will permanently delete
↳ all of your data.
$ oc delete pvc -l app=<APPLICATION_NAME>

$ oc delete all -l app=weblate \
  && oc delete secret -l app=weblate \
  && oc delete configmap -l app=weblate \
  && oc delete pvc -l app=weblate
```

Налаштування

Під час обробки шаблону буде створено відповідну `ConfigMap`, якою можна скористатися для налаштування образу `Weblate`. `ConfigMap` безпосередньо монтується як змінні середовища і спричиняють створення нової розгорнутої системи після кожного внесення до неї змін. Докладніший опис параметрів налаштувань наведено у розділі *Змінні середовища Docker*, де також можна знайти повний список змінних середовища.

Установлення на Kubernetes

Примітка: Для написання цієї частини підручника нам потрібні автори, які мають досвід роботи із Kubernetes, для докладного опису процедури налаштування.

Якщо `Weblate` за допомогою карти `Helm` для `Kubernetes`, ви зможете отримати ваш працездатний і запущений особистий екземпляр `Weblate` за декілька секунд. Усі залежності `Weblate` вже включено. Типовим засобом роботи із базою даних у таких екземплярах є `PostgreSQL`, також буде використано сталі запити щодо томів.

Карту можна знайти у <https://github.com/WeblateOrg/helm/>, переглянути її можна на <https://artifacthub.io/packages/helm/weblate/weblate>.

Установлення

```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

Налаштування

Із повним списком змінних середовища можна ознайомитися у розділі *Змінні середовища Docker*.

Залежно від налаштувань та вашого досвіду, виберіть для себе відповідний спосіб встановлення:

- `install/docker`, рекомендовано для промислових налаштувань.
- Установлення у віртуальному середовищі, рекомендовано для промислових конфігурацій:
 - *Установлення на Debian і Ubuntu*
 - *Установлення у SUSE та openSUSE*
 - *Установлення у RedHat, Fedora і CentOS*
 - *Установлення на macOS*
- `install/source`, рекомендовано для розробки.
- *Установлення на OpenShift*
- *Установлення на Kubernetes*

2.1.2 Вимоги для програмного забезпечення

Операційна система

Відомо, що Weblate працює у Linux, FreeBSD і macOS. Ймовірно, Weblate працює на більшості інших Unix-подібних системах.

Підтримки Weblate у Windows не передбачено. Втім, Weblate може там працювати — будемо раді вашим латкам, які реалізуватимуть підтримку.

Інші служби

Weblate використовує для власної роботи інші служби. Вам потрібні принаймні такі запущені служби:

- Сервер бази даних PostgreSQL, див. *Налаштування бази даних для Weblate*.
- Сервер Redis для кешування і черги завдань, див. *Фонові завдання з використанням Celery*.
- Сервер SMTP для вихідної пошти, див. *Налаштовування вихідної електронної пошти*.

Залежності Python

Weblate написано мовою **Python**. Передбачено підтримку Python 3.6 та новіших версій. Ви можете встановити залежності за допомогою `pip` або пакунків вашого дистрибутива. Повний список залежностей можна знайти у `requirements.txt`.

Основні залежності:

Django

<https://www.djangoproject.com/>

Celery

<https://docs.celeryq.dev/>

Translate Toolkit

<https://toolkit.translatehouse.org/>

translation-finder

<https://github.com/WeblateOrg/translation-finder>

Python Social Auth

<https://python-social-auth.readthedocs.io/>

Бібліотеки REST Django

<https://www.django-rest-framework.org/>

Необов'язкові залежності

Вказані нижче модулі є необхідними для користування деякими з можливостей Weblate. Список усіх цих модулів наведено у файлі `requirements-optional.txt`.

Mercurial (необов'язкова, призначено для підтримки сховищ *Mercurial*)

<https://www.mercurial-scm.org/>

phply (необов'язковий, для *Рядки PHP*)

<https://github.com/viraptor/phply>

tesseract (необов'язковий для розпізнавання тексту у *знімків вікон*)

<https://github.com/sirfz/tesseract>

python-akismet (необов'язковий, для *захисту від спаму*)

<https://github.com/Nekmo/python-akismet>

ruamel.yaml (необов'язковий, для *Файли YAML*)

<https://pypi.org/project/ruamel.yaml/>

Zeep (необов'язковий, для *ms-microsoft-terminology*)

<https://docs.python-zeep.org/>

aeidon (необов'язковий, для *Файли субтитрів*)

<https://pypi.org/project/aeidon/>

fluent.syntax (необов'язкове для *Формат Fluent*)

<https://projectfluent.org/>

Підказка: При встановленні за допомогою `pip` ви можете безпосередньо вказати бажані можливості:

```
pip install "Weblate[PHP,Fluent]"
```

Ви також можете встановити Weblate з усіма додатковими можливостями:

```
pip install "Weblate[all]"
```

Або встановити Weblate без будь-яких необов'язкових можливостей:

```
pip install Weblate
```

Залежності модуля баз даних

У Weblate передбачено підтримку PostgreSQL, MySQL і MariaDB, див. *Налаштування бази даних для Weblate* і документацію з модулів, щоб дізнатися більше.

Інші вимоги до системи

У системі має бути встановлено вказані нижче залежності:

Git

<https://git-scm.com/>

Pango, Cairo та пов'язані файли заголовків, а також дані інтроспекції GObject

<https://cairographics.org/>, <https://pango.gnome.org/>, see *Pango i Cairo*

git-review (необов'язкова, призначено для підтримки Gerrit)

<https://pypi.org/project/git-review/>

git-svn (необов'язкова, призначено для підтримки Subversion)

<https://git-scm.com/docs/git-svn>

tesseract і його дані (необов'язкова, для оптичного розпізнавання тексту на знімках вікон)

<https://github.com/tesseract-ocr/tesseract>

licensee (необов'язкова, для виявлення умов ліцензування при створенні складників)

<https://github.com/licensee/licensee>

Залежності для збирання

Для збирання *Залежності Python* може виникнути потреба у встановленні відповідних залежностей. Список залежить від способу встановлення, тому ознайомтеся із документацією до окремих пакунків. Можливо, встановлення не знадобиться, якщо ви користуєтеся попередньо зібраним *Wheels* під час встановлення за допомогою *pip* або пакунками з вашого дистрибутива.

Pango i Cairo

Змінено в версії 3.7.

Weblate використовує Pango i Cairo для обробки растрових віджетів (див. *promotion*) і перевірки обробки перекладених рядків в інтерфейсі (див. *Керування шрифтами*). Для належного встановлення прив'язок Python для цих складників вам слід встановити спочатку загальносистемні бібліотеки — вам потрібні i Cairo, i Pango, які, очевидно, залежать від GLib. Усі ці пакунки слід встановити разом із пакунками для розробки та даними інтроспекції GObject.

2.1.3 Перевірка підписів випуску

Випуск Weblate має криптографічний підпис розробника, відповідального за випуск. У поточній версії цим розробником є Міхал Чихарж (Michal Čihař). Відбиток його ключа PGP:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

а додаткові відомості щодо ідентифікації можна знайти тут: <<https://keybase.io/nijel>>.

Вам слід перевірити, чи відповідає підпис отриманому вами архіву. Перевірка дає змогу впевнитися, що ви використовуєте саме той код, який було випущено розробниками. Вам також слід перевірити дату підписування, щоб переконатися, що отримано найсвіжішу версію.

Разом із кожним архівом розповсюджуються файли `.asc`, які містять підпис PGP до архіву. Якщо ви запишете файл архіву разом із підписом до однієї теки, ви зможете перевірити підпис:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Can't check signature: public key not found
```

Як можна бачити, GPG скаржиться на те, що відкритий ключ є невідомим програмі. На цьому етапі ви можете виконати один з таких кроків:

- Скористайтеся *wkd* для отримання ключа:

```
$ gpg --auto-key-locate wkd --locate-keys michal@cihar.com
pub  rsa4096 2009-06-17 [SC]
    63CB1DF1EF12CF2AC0EE5A329C27B31342B7511D
uid          [ultimate] Michal Čihař <michal@cihar.com>
uid          [ultimate] Michal Čihař <nijel@debian.org>
uid          [ultimate] [jpeg image of size 8848]
uid          [ultimate] Michal Čihař (Braiiins) <michal.cihar@braiins.cz>
sub  rsa4096 2009-06-17 [E]
sub  rsa4096 2015-09-09 [S]
```

- Отримайте збірку ключів з сервера Міхала, потім імпортуйте її за допомогою такої команди:

```
$ gpg --import wmxth3chu9jfxdxywj1skpmhsj311mzm
```

- Отримайте та імпортуйте ключ з одного із серверів ключів:

```
$ gpg --keyserver hkp://pgp.mit.edu --recv-keys 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: key 9C27B31342B7511D: "Michal Čihař <michal@cihar.com>" imported
gpg: Total number processed: 1
gpg:          unchanged: 1
```

Це дещо поліпшить ситуацію — на цьому кроці ви можете перевірити, чи є підпис від вказаного ключа коректним, але ви все ще не можете довіряти імені, яке використано у ключі:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:          using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:          aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:          aka "[jpeg image of size 8848]" [ultimate]
gpg:          aka "Michal Čihař (Braiiins) <michal.cihar@braiins.cz>" [ultimate]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

Проблема полягає у тому, що випустити ключ для цього імені може будь-хто. Вам слід переконатися, що ключ справді належить згаданій особі. У підручнику з конфіденційності GNU цю тему висвітлено у розділі [Validating other keys on your public keyring](#). Найнадійнішим способом встановлення автентичності ключа є особиста зустріч із розробником і обмін відбитками ключів. Втім, ви також можете покластися на мережу довіри. У такій мережі ви можете довіритися ключ опосередковано — через підписи інших осіб, які зустрічалися із розробником особисто.

Щойно ключ стане довіреним, має бути показано попередження:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:          using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:          aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:          aka "[jpeg image of size 8848]" [ultimate]
gpg:          aka "Michal Čihař (Braiiins) <michal.cihar@braiins.cz>" [ultimate]
```

Якщо підпис є некоректним (до архіву було внесено зміни), ви отримаєте ясне повідомлення щодо помилки, незалежно від того, є ключ довіреним чи ні:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:          using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: BAD signature from "Michal Čihař <michal@cihar.com>" [ultimate]
```

2.1.4 Права доступу у файловій системі

Процес Weblate повинен мати можливість читати дані з каталогу, де зберігає дані, і записувати дані до нього — `DATA_DIR`. Усі файли у цьому каталозі мають належати користувачеві, від імені якого запущено усі процеси Weblate (типово, WSGI і Celery, див. *Запуск сервера* і *Фонові завдання з використанням Celery*) і бути придатними до запису від імені цього користувача.

За типових налаштувань, ці файли розташовуються у тій самій ієрархії каталогів, що і початкові коди Weblate. Втім, ви можете надати перевагу пересуванню цих файлів до вдалішого місця, наприклад `/var/lib/weblate`.

Weblate намагається створити каталоги автоматично, але це неможливо зробити, якщо для цього немає прав доступу.

Вам також слід бути обережним із запуском *Команди керування*, оскільки програму слід запускати від імені того самого користувача, що і користувач, від імені якого запущено Weblate, інакше права доступу до деяких файлів можуть виявитися не тими, які потрібні для працездатності системи.

У контейнері Docker усі файли в томі `/app/data` мають належати користувачеві `weblate` всередині контейнера (UID 1000).

Дивись також:

Обслуговування статичних файлів

2.1.5 Налаштування бази даних для Weblate

Рекомендуємо користуватися Weblate у поєднанні із сервером баз даних PostgreSQL.

Дивись також:

Використання потужного рушія бази даних, Databases, Перенесення даних з інших баз даних до PostgreSQL

PostgreSQL

PostgreSQL, зазвичай є найкращим вибором для заснованих на Django сайтів. Це еталонна база даних, яка використовується для реалізації шару бази даних Django.

Примітка: Weblate використовує розширення обробки триграм, яке можна встановити у деяких випадках. Пошукайте `postgresql-contrib` або пакунок із подібною назвою.

Дивись також:

PostgreSQL notes

Створення бази даних у PostgreSQL

Зазвичай, варто зробити так, щоб Weblate працював із окремою базою даних і окремим обліковим записом користувача:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser --superuser --pwprompt weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Підказка: Якщо ви не хочете робити користувача Weblate надкористувачем у PostgreSQL, ви можете пропустити це. У такому випадку вам доведеться виконати деякі з кроків з перенесення вручну від імені надкористувача PostgreSQL у схемі, яку використовуватиме Weblate:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
CREATE EXTENSION IF NOT EXISTS btree_gin WITH SCHEMA weblate;
```

Налаштовування Weblate на використання PostgreSQL

Фрагмент `settings.py` для PostgreSQL:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    }
}
```

Під час перенесення бази даних виконується `ALTER ROLE` щодо ролі у базі даних, яку використовує Weblate. У більшості випадків назва ролі збігається із іменем користувача. У складніших випадках назва ролі відрізняється від імені користувача, і ви побачите повідомлення про помилку щодо ролі, якої не існує, під час перенесення бази даних (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). Таке, як відомо, трапляється із базою даних Azure для PostgreSQL, але проблеми не обмежено лише цим середовищем. Будь ласка, встановіть значення `ALTER_ROLE`, щоб змінити назву ролі, яку Weblate має поміняти під час перенесення бази даних.

MySQL і MariaDB

Підказка: Деякі можливості Weblate працюватимуть краще з використанням *PostgreSQL*. Це стосується пошуку і пам'яті перекладів — можливостей, у яких використовується повнотекстова обробка у базі даних — реалізація у PostgreSQL є найкращою.

Weblate можна також користуватися із MySQL або MariaDB. Будь ласка, ознайомтеся із розділами [MySQL notes](#) і [MariaDB notes](#), щоб дізнатися більше про можливі проблеми із використанням Django разом із цими серверами баз даних. Через обмеження, які накладаються використанням інших баз даних, рекомендуємо використовувати для нововстановлених екземплярів *PostgreSQL*.

Weblate вимагає MySQL хоча б версії 5.7.8, чи MariaDB принаймні 10.2.7 версії.

Рекомендовано такі налаштування для Weblate:

- Скористайтеся набором символів `utf8mb4`, щоб уможливити обробку вищих блоків Unicode (зокрема емоджі).
- Налаштуйте сервер за допомогою `innodb_large_prefix` так, щоб уможливити довші індекси у текстових полях.
- Установіть рівень ізоляції `READ COMMITTED`.
- Режим SQL має бути встановлено у значення `STRICT_TRANS_TABLES`.

MySQL 8.x, MariaDB 10.5.x або новіші версії мають придатні типові налаштування, отже потреби у додатковому коригуванні налаштувань сервера не повинно виникнути — усе, що потрібно, можна налаштувати на клієнтському боці.

Далі наведено приклад `/etc/my.cnf.d/server.cnf` для сервера із 8 гігабайтами оперативної пам'яті. Вказані параметри мають бути достатніми для більшості випадків встановлення. У MySQL і MariaDB передбачено параметри підвищення швидкодії вашого сервера, які ми вважаємо необов'язковими, якщо ви не плануєте одночасну роботу у системі багатьох користувачів. Докладніший опис можна знайти у документації, яка надається розробниками відповідного програмного забезпечення.

Надзвичайно важливим для зменшення кількості проблем при встановленні є встановлення належного значення параметра `innodb_file_per_table` і перезапуск MySQL/MariaDB до встановлення Weblate.

```
[mysqld]
character-set-server = utf8mb4
character-set-client = utf8mb4
collation-server = utf8mb4_unicode_ci

datadir=/var/lib/mysql

log-error=/var/log/mariadb/mariadb.log

innodb_large_prefix=1
innodb_file_format=Barracuda
innodb_file_per_table=1
innodb_buffer_pool_size=2G
sql_mode=STRICT_TRANS_TABLES
```

Підказка: Якщо ви бачите повідомлення про помилку `#1071 - Specified key was too long; max key length is 767 bytes`, будь ласка, оновіть ваші налаштування, включивши до них параметри `innodb`, вказані вище, і перезапустіть встановлення.

Підказка: Якщо ви отримуєте повідомлення про помилку `#2006 - MySQL server has gone away`, проблему може усунути встановлення належного значення `CONN_MAX_AGE`.

Налаштовування Weblate на використання MySQL/MariaDB

Фрагмент `settings.py` для MySQL і MariaDB:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "127.0.0.1",
        # Set to empty string for default
        "PORT": "3306",
        # In case you wish to use additional
        # connection options
        "OPTIONS": {},
    }
}
```

Вам слід створити обліковий запис користувача `weblate` у MySQL або MariaDB до початку встановлення. Для цього скористайтеся наведеними далі командами:

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

2.1.6 Інші налаштування

Налаштовування вихідної електронної пошти

Weblate надсилає повідомлення електронної пошти з різних нагод — для активації облікових записів та сповіщення щодо різних подій, які налаштовано користувачами. Для надсилання пошти програмі потрібен доступ до сервера SMTP.

Сервер електронної пошти можна налаштувати за допомогою таких параметрів: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_TLS`, `EMAIL_USE_SSL`, `EMAIL_HOST_USER` і `EMAIL_PORT`. Назви параметрів є доволі прозорими з точки зору їхнього призначення, але ви можете знайти і докладніший опис у документації до Django.

Підказка: Якщо ви отримуєте повідомлення про помилку щодо непідтримуваного способу розпізнавання (наприклад, `SMTP AUTH extension not supported by server`), найімовірнішою причиною є використання незахищеного з'єднання і відмова сервера у розпізнаванні у незахищений спосіб. У таких випадках варто спробувати увімкнути `EMAIL_USE_TLS`.

Дивись також:

Не надходять повідомлення електронної пошти від Weblate, Налаштовування вихідної пошти у контейнері Docker

Робота за реверсивним проксі-сервером

Працездатність можливостей Weblate залежить від можливості отримання клієнтської IP-адреси. Серед цих можливостей *Обмеження частоти*, *Захист від спаму* та *Часопис перевірок*.

За типових налаштувань Weblate обробляє IP-адресу з REMOTE_ADDR, яка встановлюється обробником WSGI.

Якщо ви працюєте з реверсивним проксі, це поле, найімовірніше, міститиме його адресу. Вам слід налаштувати Weblate на довіру до додаткових заголовків HTTP і визначити IP-адресу з цих заголовків. Цю можливість не може бути типово увімкнено, оскільки її вмикання дозволить підміну IP-адреси для встановлених екземплярів, які не використовують оберненого проксі. Для більшості звичайних конфігурацій може бути достатньо вмикання `IP_BEHIND_REVERSE_PROXY`, але вам, ймовірно, доведеться скоригувати значення параметрів `IP_PROXY_HEADER` і `IP_PROXY_OFFSET`.

Ще однією річчю, про яку слід подбати, є заголовок `Host`. Він має відповідати тому, що налаштовано у `SITE_DOMAIN`. Можливо, знадобиться додаткове налаштування у вашому оберненому проксі (наприклад, скористатися `ProxyPreserveHost On` для Apache або `proxy_set_header Host $host;` для nginx).

Дивись також:

Захист від спаму, *Обмеження частоти*, *Часопис перевірок*, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`, `SECURE_PROXY_SSL_HEADER`

HTTP-проксі

Weblate виконує команди системи керування версіями, а ці команди приймають налаштування проксі-сервера із середовища. Рекомендованим підходом є визначення параметрів проксі-сервера у `settings.py`:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Дивись також:

Змінні середовища для проксі-серверів

2.1.7 Коригування налаштувань

Дивись також:

Зразок налаштувань

Скопіюйте `weblate/settings_example.py` до `weblate/settings.py` і скоригуйте його відповідно до ваших потреб. Ймовірно, вам потрібно скоригувати такі параметри: `ADMINS`

Список адміністраторів сайта, які отримуватимуть сповіщення при помилках, зокрема сповіщення щодо невдалого об'єднання або помилок Django.

Дивись також:

`ADMINS`, *Належне налаштування записів адміністраторів*

`ALLOWED_HOSTS`

Вам слід встановити для цього параметра значення списку вузлів, які має обслуговувати ваш сайт. Приклад:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Крім того, ви можете включити символ-замінник:

```
ALLOWED_HOSTS = [ "*" ]
```

Дивись також:

[ALLOWED_HOSTS](#), [WEBLATE_ALLOWED_HOSTS](#), [Налаштовування дозволених вузлів](#)

SESSION_ENGINE

Налаштування способу збереження ваших сеансів. Якщо ви збережете типовий рушій обробника бази даних, вам слід запланувати **weblate clearsessions** для видалення застарілих даних сеансу з бази даних.

Якщо ви використовуєте Redis як кеш (див. [Увімкніть кешування](#)) рекомендуємо скористатися ним і для сеансів:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Дивись також:

[Configuring the session engine](#), [SESSION_ENGINE](#)

DATABASES

Можливість з'єднання із сервером бази даних. Будь ласка, ознайомтеся із документацією до Django, щоб дізнатися більше.

Дивись також:

[Налаштування бази даних для Weblate](#), [DATABASES](#), [Databases](#)

DEBUG

Вимкніть цю можливість для будь-яких промислових серверів. Якщо діагностичний режим увімкнено, Django показуватиме користувачам дані зворотного трасування при помилці. Якщо ви вимкнете її, помилки буде надіслано електронною поштою до ADMINS (див. вище).

Режим діагностики також уповільнює роботу Weblate, оскільки у такому режимі Django на внутрішньому рівні зберігає набагато більше даних.

Дивись також:

[DEBUG](#), [Вимкнути режим діагностики](#)

DEFAULT_FROM_EMAIL

Адреса електронної пошти відправника для вихідної електронної пошти, наприклад, повідомлень електронної пошти щодо реєстрації.

Дивись також:

[DEFAULT_FROM_EMAIL](#)

SECRET_KEY

Ключ, який використовується Django для підписування даних у куках. Див. [Секретний ключ Django](#), щоб дізнатися більше.

Дивись також:

[SECRET_KEY](#)

SERVER_EMAIL

Адреса електронної пошти, яку буде використано для надсилання повідомлень електронної пошти адміністратору, наприклад, для сповіщень щодо помилок під час об'єднання змін.

Дивись також:

[SERVER_EMAIL](#)

2.1.8 Заповнення бази даних

Коли ваші налаштування буде готово, ви можете запустити `weblate migrate` для створення структури бази даних. Після цього ви зможете створювати проекти перекладу за допомогою інтерфейсу адміністратора.

Якщо ви хочете запустити встановлення у неінтерактивному режимі, ви можете скористатися командою `weblate migrate --noinput`, а потім створити користувача-адміністратора за допомогою команди `createadmin`.

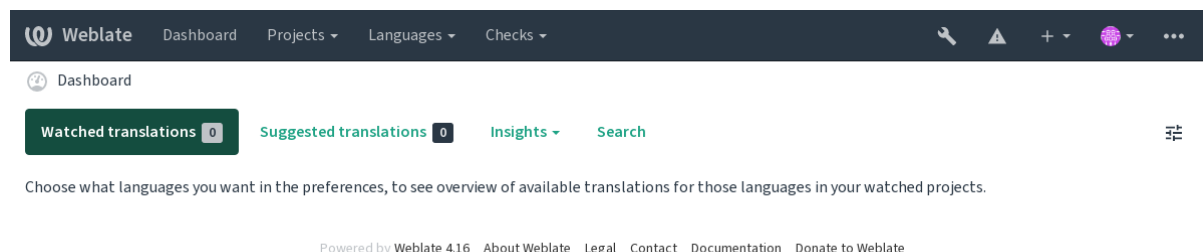
Щойно потрібні дії буде виконано, вам слід також ознайомитися зі *Звітом щодо швидкодії* в інтерфейсі адміністратора, там ви зможете знайти підказки щодо потенційних недоліків у налаштуваннях на вашому сайті.

Дивись також:

Налаштування, Список привілеїв і вбудовані ролі

2.1.9 Промислові налаштування

Для налаштування промислового сервера вам слід виконати коригування, які описано у наступних розділах. Для найкритичніших параметрів буде показано попередження, вказівкою на яке є знак оклику на верхній панелі, який буде показано, якщо ви увійдете на сервер як супер користувач:



Також рекомендуємо вам звернути увагу на підсумки перевірок, які буде започатковано Django (хоча вам і не треба виправляти усі причини показаних попереджень):

```
weblate check --deploy
```

Ви також можете скористатися списком дій з розділу *Інтерфейс керування*.

Дивись також:

Deployment checklist

Вимкнути режим діагностики

Вимикання режиму діагностики Django (*DEBUG*):

```
DEBUG = False
```

З увімкненим режимом діагностики Django зберігає усі виконані запити та показує користувачеві зворотні трасування помилок. Такі надмірні дані є небажаними у промисловій конфігурації.

Дивись також:

Коригування налаштувань

Належне налаштування записів адміністраторів

Установіть належні адреси адміністраторів за допомогою параметра `ADMINS` для визначення того, хто отримуватиме повідомлення електронної пошти у випадку, якщо щось на сервері піде не так. Приклад:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Дивись також:

Коригування налаштувань

Установіть належний домен сайта

Скоригуйте назву сайта і домен у інтерфейсі адміністратора, інакше посилання у RSS або у повідомленнях електронної пошти щодо реєстрації не працюватимуть. Налаштувати ці параметри можна за допомогою параметра `SITE_DOMAIN`, який має містити назву домену сайта.

Змінено в версії 4.2: До випуску 4.2 замість цього використовувалася бібліотека sites Django. Будь ласка, див. The “sites” framework.

Дивись також:

Налаштовування дозволених вузлів, Належне налаштування HTTPS SITE_DOMAIN, WEBLATE_SITE_DOMAIN, ENABLE_HTTPS

Належне налаштування HTTPS

Наполегливо рекомендуємо вам запускати з Weblate з використанням шифрованого протоколу HTTPS. Після його вмикання вам слід встановити `ENABLE_HTTPS` у параметрах сервера:

```
ENABLE_HTTPS = True
```

Підказка: Вам також, ймовірно, варто налаштувати HSTS. Див. [SSL/HTTPS](#), щоб дізнатися більше.

Дивись також:

ENABLE_HTTPS, Налаштовування дозволених вузлів, Установіть належний домен сайта

Установіть SECURE_HSTS_SECONDS належним чином

Якщо обслуговування вашого сайта виконується за допомогою SSL, вам варто встановити значення для `SECURE_HSTS_SECONDS` у `:file`settings.py``, щоб увімкнути строгий захист передавання даних (HTTP Strict Transport Security). Типово, його встановлено у значення 0, як це показано нижче.

```
SECURE_HSTS_SECONDS = 0
```

Якщо встановити ненульове ціле значення, `django.middleware.security.SecurityMiddleware` встановлює заголовок `HTTP Strict Transport Security` в усіх відповідях, які ще не містять такого заголовка.

Попередження: Установлення помилкового значення може незворотно (на деякий проміжок часу) зашкодити працездатності вашого сайта. Через це, варто спочатку ознайомитися зі документацією до [HTTP Strict Transport Security](#).

Використання потужного рушія бази даних

- Будь ласка, скористайтесь PostgreSQL для промислових середовищ. Див. *Налаштування бази даних для Weblate*, щоб дізнатися більше.
- Застосовуйте суміжне розташування для запуску сервера бази даних, інакше швидкодія або надійність мережі можуть зіпсувати роботу Weblate.
- Перевірте швидкодію сервера бази даних або налаштуйте його конфігурацію, наприклад, застосувавши PG Tune.

Дивись також:

Налаштування бази даних для Weblate, Перенесення даних з інших баз даних до PostgreSQL, Коригування налаштувань, Databases

Увімкніть кешування

Якщо можна, скористайтесь Redis з Django коригуванням значення змінної налаштувань CACHES. Приклад:

```
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/0",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # 'LOCATION': 'unix:///var/run/redis/redis.sock?db=0',
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    },
}
```

Підказка: Якщо ви хочете змінити параметри Redis для кешування, вам, ймовірно, варто також змінити їх і для Celery, див. *Фонові завдання з використанням Celery*.

Дивись також:

Кешування аватару, Django's cache framework

Кешування аватару

На додачу до кешування даних Django, Weblate виконує кешування аватарів. Рекомендуємо для виконання цього завдання скористатися окремим файловим кешем:

```
CACHES = {
    "default": {
        # Default caching backend setup, see above
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "unix:///var/run/redis/redis.sock?db=0",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
    },
}
```

(continues on next page)

(continued from previous page)

```

    "TIMEOUT": 604800,
    "OPTIONS": {
        "MAX_ENTRIES": 1000,
    },
},
}

```

Дивись також:

`ENABLE_AVATARS`, `AVATAR_URL_PREFIX`, *Аватари*, *Увімкніть кешування*, Django's cache framework

Налаштовування надсилання електронної пошти

Weblate має надсилати повідомлення електронної пошти у декількох випадках. Ці повідомлення повинні мати правильну адресу відправника. Будь ласка, налаштуйте `SERVER_EMAIL` і `DEFAULT_FROM_EMAIL` так, щоб значення змінних відповідали вашому середовищу. Приклад:

```

SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"

```

Примітка: Щоб вимкнути надсилання повідомлень електронної пошти з Weblate, встановіть `EMAIL_BACKEND` у значення `django.core.mail.backends.dummy.EmailBackend`.

У результаті буде вимкнено надсилання *усіх* повідомлень електронної пошти, зокрема повідомлень щодо реєстрації та повідомлень щодо скидання паролів.

Дивись також:

Коригування налаштувань, *Налаштовування вихідної електронної пошти*, `EMAIL_BACKEND`, `DEFAULT_FROM_EMAIL`, `SERVER_EMAIL`

Налаштовування дозволених вузлів

Django потрібно, щоб у змінній `ALLOWED_HOSTS` містився список назв доменів, які може обслуговувати ваш сайт. Якщо ця змінна міститиме порожнє значення, будь-які запити буде заблоковано.

Якщо для змінної вказано значення, яке не відповідає вашому серверу HTTP, ви отримаєте повідомлення про помилки, подібні до `Invalid HTTP_HOST header: '1.1.1.1'`. Тоді вам потрібно додати `'1.1.1.1'` до `ALLOWED_HOSTS`.

Підказка: У контейнері Docker доступ до відповідної можливості здійснюється за допомогою змінної `WEBLATE_ALLOWED_HOSTS`.

Дивись також:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Установіть належний домен сайта*

Секретний ключ Django

Значення параметра `SECRET_KEY` використовується Django для підписування кук. Вам слід створити ваше власне значення і не користуватися значенням з прикладу налаштувань.

Створити новий ключ можна за допомогою програми `weblate/examples/generate-secret-key`, яка постачається разом із Weblate.

Дивись також:

`SECRET_KEY`

Запуск завдань щодо супроводу

Для забезпечення оптимальної швидкодії варто запускати деякі завдання з супроводу працездатності сервера у фоновому режимі. У нових версіях це автоматично виконується за допомогою *Фонові завдання з використанням Celery* і стосується таких завдань:

- Перевірки коректності налаштувань (щогодинно).
- Надсилання змін з черги (щогодини), див. «*Лінії*» *внески* і `commit_pending`.
- Оновлення нагадувань щодо складників (щоденно).
- Оновлення віддалених гілок (щодня), див. `AUTO_UPDATE`.
- Резервного копіювання пам'яті перекладів до JSON (щодня), див. `dump_memory`.
- Завдань із супроводу обробки всього тексту й бази даних (щоденних і щотижневих завдань), див. `cleanuptrans`.

Змінено в версії 3.2: Починаючи з версії 3.2, типовим способом виконання цих завдань є використання Celery, а Weblate постачається уже налаштованим відповідним чином, див. *Фонові завдання з використанням Celery*.

Локалі і кодування системи

Для локалі системи слід вибрати якусь із локалей UTF-8. У більшості дистрибутивів Linux такий вибір є типовим. Якщо у вашій системі вибрано якусь іншу локаль, будь ласка, змініть локаль на один з варіантів UTF-8.

Зробити це можна, наприклад, редагуванням `/etc/default/locale` зі встановленням `LANG="C.UTF-8"`.

У деяких випадках окремі служби мають окремі налаштування для локалей. Це залежить від дистрибутива та вебсерверів, тому для вирішення цього завдання ознайомтеся із документацією до пакунків вашого сервера.

У Apache на Ubuntu використовують `/etc/apache2/envvars`:

```
export LANG='en_US.UTF-8'
export LC_ALL='en_US.UTF-8'
```

У Apache на CentOS використовують `/etc/sysconfig/httpd` (або `/opt/rh/httpd24/root/etc/sysconfig/httpd`):

```
LANG='en_US.UTF-8'
```

Використання нетипових уповноважень за сертифікатом

Weblate не виконує перевірку сертифікатів SSL під час обробки запитів HTTP. Якщо ви використовуєте нетипову службу сертифікації, яка не є довіреною у типових пакетах, вам слід додати її сертифікат як довірений.

Бажаним підходом для цього є додавання на загальносистемному рівні. Щоб дізнатися більше про способи додавання, будь ласка, ознайомтеся із документацією до дистрибутива (наприклад, у Debian досягти потрібного результату можна записом сертифіката служби сертифікації до `/usr/local/share/ca-certificates/` із наступним виконанням команди `update-ca-certificates`).

Щойно сертифікат буде додано, загальносистемні інструменти зареєструють довіру до сертифіката. Це стосується і Git.

У кодї мовою Python вам доведеться налаштувати запити на використання загальносистемного набору служб сертифікації, замість набору, який постачається разом із інтерпретатором. Для цього слід вставити до `settings.py` такий фрагмент коду (шлях є специфічним для Debian):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Стиснення клієнтських даних

Weblate постачається із набором файлів JavaScript і CSS. З міркувань забезпечення оптимальної швидкодії варто стиснути їх до надсилання клієнту. У типовій конфігурації результат досягається пакуванням «на льоту» із певною зайвою витратою ресурсів. На серверах із значним навантаженням рекомендуємо увімкнути режим попереднього стиснення. Зробити це слід у налаштуваннях, а стиснення слід започатковувати після кожного оновлення Weblate.

Перемикання у налаштуваннях є простим — увімкніть `django.conf.settings.COMPRESS_OFFLINE` і налаштуйте `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` (останній вже включено у приклад налаштувань):

```
COMPRESS_OFFLINE = True
```

При кожному розгортанні вам слід стискати файли відповідно до поточної версії:

```
weblate compress
```

Підказка: У офіційному образі для Docker цю можливість вже увімкнено.

Дивись також:

[Common Deployment Scenarios](#), *Обслуговування статичних файлів*

2.1.10 Запуск сервера

Підказка: Якщо у вас немає досвіду з послугами, описаними далі, ви можете спробувати *Установлення за допомогою Docker*.

Для роботи Weblate слід мати запущеними декілька служб. Рекомендований набір складається з таких служб:

- Сервер бази даних (див. *Налаштування бази даних для Weblate*)
- Сервер кешу (див. *Увімкніть кешування*)
- Сервер-обгортка для статичних файлів і переривання SSL (див. *Обслуговування статичних файлів*)

- Сервер WSGI для динамічних даних (див. [Зразок налаштувань для NGINX і uWSGI](#))
- Celery для виконання фонових завдань (див. [Фонові завдання з використанням Celery](#))

Примітка: Між службами є деякі залежності. Наприклад, служби кешування та бази даних має бути запущено до запуску процесів Celery або uwsgi.

У більшості випадків усі служби працюють на одному (віртуальному) сервері, але якщо ваш екземпляр є високонавантаженим, ви можете розділити служби. Єдиним обмеженням щодо цього є те, що сервери Celery і Wsgi повинні мати доступу до [DATA_DIR](#).

Примітка: Процес WSGI має бути запущено від імені того ж користувача, що й Celery, інакше файли в [DATA_DIR](#) буде збережено від імені різних власників, що призведе до проблем під час роботи сервера.

Див. також [Права доступу у файлової системі](#) і [Фонові завдання з використанням Celery](#).

Запуск вебсервера

Запуск Weblate не відрізняється від запуску будь-якої іншої програми на основі Django. Django, зазвичай, виконується як uWSGI або fsgi (див. приклади для різних вебсерверів нижче).

Для тестування ви можете скористатися вбудованим до Django сервером:

```
weblate runserver
```

Попередження: НЕ КОРИСТУЙТЕСЯ ЦИМ СЕРВЕРОМ У ПРОМИСЛОВОМУ СЕРЕДОВИЩІ. Він не проходив перевірки захисту або швидкодії. Також ознайомтеся із документацією до Django щодо `runserver`.

Підказка: Вбудований сервер Django обслуговує статичні файли лише з увімкненим [DEBUG](#), оскільки його призначено лише для розробників. Для промислових потреб, будь ласка, скористайтеся конфігураціями wsgi з [Зразок налаштувань для NGINX і uWSGI](#), [Зразок налаштувань для Apache](#), [Зразок налаштувань для Apache і Unicorn](#) та [Обслуговування статичних файлів](#).

Обслуговування статичних файлів

Змінено в версії 2.4: До версії 2.4 Weblate не міг належним чином користуватися бібліотекою обробки статичних файлів Django, і налаштування було складнішими.

Django потребує збирання статичних файлів у одному каталозі. Щоб забезпечити потрібні умови, скористайтеся командою `weblate collectstatic --noinput`. У результаті її виконання статичні файли буде скопійовано до каталогу, який визначається параметром `STATIC_ROOT` (типовим є каталог `static` у DATA_DIR`).

Рекомендуємо обслуговувати статичні файли безпосередньо з вашого сервера. Вам слід використовувати це для таких шляхів:

/static/

Обслуговує статичні файли для Weblate і адміністративного інтерфейсу (з визначеного змінною `STATIC_ROOT`).

/media/

Використовується для вивантажених користувачами мультимедійних даних (наприклад, знімків вікон).

/favicon.ico

Має бути перезаписано для перезапису правила обслуговування `/static/favicon.ico`.

Дивись також:

Зразок налаштувань для NGINX і uWSGI, Зразок налаштувань для Apache, Зразок налаштувань для Apache і Unicorn, Стиснення клієнтських даних, How to deploy Django, How to deploy static files

Правила щодо безпеки даних

За типових налаштувань Weblate увімкнено проміжне програмне забезпечення `weblate.middleware.SecurityMiddleware`, яке встановлює пов'язані із захистом заголовки HTTP, зокрема `Content-Security-Policy` або `X-XSS-Protection`. Ці заголовки типово налаштовано на роботу із Weblate та його налаштуваннями, але це може потребувати підлаштування до вашого середовища.

Дивись також:

`CSP_SCRIPT_SRC`, `CSP_IMG_SRC`, `CSP_CONNECT_SRC`, `CSP_STYLE_SRC`, `CSP_FONT_SRC`

Зразок налаштувань для NGINX і uWSGI

Для запуску промислового сервера скористайтеся обгорткою `wsgi`, яку буде встановлено з Weblate (у віртуальному середовищі відповідний файл встановлено як `~/weblate-env/lib/python3.9/site-packages/weblate/wsgi.py`). Не забудьте встановити шлях пошуку бібліотек у Python відповідно до вашого віртуального середовища (наприклад, за допомогою інструкції `virtualenv = /home/user/weblate-env у uWSGI`).

Наведені нижче налаштування призначено для запуску Weblate як uWSGI із сервером NGINX.

Налаштування для NGINX (також доступні як `weblate/examples/weblate.nginx.conf`):

```
#
# nginx configuration for Weblate
#
# You will want to change:
#
# - server_name
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
server {
    listen 80;
    server_name weblate;
    # Not used
    root /var/www/html;

    location ~ ^/favicon.ico$ {
        # DATA_DIR/static/favicon.ico
        alias /home/weblate/data/static/favicon.ico;
        expires 30d;
    }

    location /static/ {
        # DATA_DIR/static/
        alias /home/weblate/data/static/;
        expires 30d;
    }

    location /media/ {
```

(continues on next page)

(continued from previous page)

```

    # DATA_DIR/media/
    alias /home/weblate/data/media/;
    expires 30d;
}

location / {
    include uwsgi_params;
    # Needed for long running operations in admin interface
    uwsgi_read_timeout 3600;
    # Adjust based to uwsgi configuration:
    uwsgi_pass unix:///run/uwsgi/app/weblate/socket;
    # uwsgi_pass 127.0.0.1:8080;
}
}

```

Налаштування для uWSGI (також доступне як `weblate/examples/weblate.uwsgi.ini`):

```

#
# uWSGI configuration for Weblate
#
# You will want to change:
#
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
[uwsgi]
plugins      = python3
master       = true
protocol     = uwsgi
socket       = 127.0.0.1:8080
wsgi-file    = /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/wsgi.
↳ py

# Add path to Weblate checkout if you did not install
# Weblate by pip
# python-path = /path/to/weblate

# In case you're using virtualenv uncomment this:
virtualenv = /home/weblate/weblate-env

# Needed for OAuth/OpenID
buffer-size  = 8192

# Reload when consuming too much of memory
reload-on-rss = 250

# Increase number of workers for heavily loaded sites
workers      = 8

# Enable threads for Sentry error submission
enable-threads = true

# Child processes do not need file descriptors
close-on-exec = true

# Avoid default 0000 umask
umask = 0022

# Run as weblate user

```

(continues on next page)

(continued from previous page)

```

uid = weblate
gid = weblate

# Enable harakiri mode (kill requests after some time)
# harakiri = 3600
# harakiri-verbose = true

# Enable uWSGI stats server
# stats = :1717
# stats-http = true

# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true

```

Дивись також:

How to use Django with uWSGI

Зразок налаштувань для Apache

Рекомендовано скористатися попереднім відгалуженням MPM при використанні WSGI разом із Weblate.

У наведених нижче налаштуваннях Weblate запущено як WSGI, вам знадобиться увімкнений `mod_wsgi` (налаштування доступні як `weblate/examples/apache.conf`):

```

#
# VirtualHost for Weblate
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate_
↪ request-timeout=600

```

(continues on next page)

(continued from previous page)

```

WSGIProcessGroup weblate
WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias / /home/weblate/weblate-env/lib/python3.9/site-packages/
↳weblate/wsgi.py process-group=weblate
WSGIPassAuthorization On

<Directory /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/>
  <Files wsgi.py>
    Require all granted
  </Files>
</Directory>

</VirtualHost>

```

Примітка: Для роботи Weblate потрібен Python 3, тому вам слід переконатися, що ви працюєте із версією modwsgi для Python 3. Зазвичай, ця версія встановлюється з окремого пакунка, наприклад `libapache2-mod-wsgi-py3`.

Дивись також:

Локалі і кодування системи, [How to use Django with Apache and mod_wsgi](#)

Зразок налаштувань для Apache і Gunicorn

Наведені нижче налаштування призначено для запуску Weblate у Gunicorn із Apache 2.4 (налаштування доступні як `weblate/examples/apache.gunicorn.conf`):

```

#
# VirtualHost for Weblate using gunicorn on localhost:8000
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
<VirtualHost *:443>
  ServerAdmin admin@weblate.example.org
  ServerName weblate.example.org

  # DATA_DIR/static/favicon.ico
  Alias /favicon.ico /home/weblate/data/static/favicon.ico

  # DATA_DIR/static/
  Alias /static/ /home/weblate/data/static/
  <Directory /home/weblate/data/static/>
    Require all granted
  </Directory>

  # DATA_DIR/media/
  Alias /media/ /home/weblate/data/media/
  <Directory /home/weblate/data/media/>
    Require all granted
  </Directory>

```

(continues on next page)

(continued from previous page)

```

SSLEngine on
SSLCertificateFile /etc/apache2/ssl/https_cert.cert
SSLCertificateKeyFile /etc/apache2/ssl/https_key.pem
SSLProxyEngine On

ProxyPass /favicon.ico !
ProxyPass /static/ !
ProxyPass /media/ !

ProxyPass / http://localhost:8000/
ProxyPassReverse / http://localhost:8000/
ProxyPreserveHost On
</VirtualHost>

```

Дивись також:

How to use Django with Gunicorn

Запуск Weblate у певному каталозі

Нове в версії 1.3.

Рекомендовано скористатися попереднім відгалуженням MPM при використанні WSGI разом із Weblate.

Зразок налаштувань Apache для обслуговування Weblate у /weblate. Знову ж таки, використано mod_wsgi (ці налаштування можна знайти у файлі weblate/examples/apache-path.conf):

```

#
# VirtualHost for Weblate, running under /weblate path
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /weblate/favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /weblate/static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /weblate/media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate_
    ↳ request-timeout=600
    WSGIProcessGroup weblate

```

(continues on next page)

(continued from previous page)

```

WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.9/site-packages/
↪weblate/wsgi.py process-group=weblate
WSGIPassAuthorization On

<Directory /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/>
  <Files wsgi.py>
    Require all granted
  </Files>
</Directory>

</VirtualHost>

```

Крім того, вам слід скоригувати `weblate/settings.py`:

```
URL_PREFIX = "/weblate"
```

2.1.11 Фонові завдання з використанням Celery

Нове в версії 3.2.

Weblate використовує Celery для виконання регулярних завдань та завдань із резервного копіювання. У вас має працювати служба Celery, яка запускати ці завдання. Наприклад, ця служба відповідає за обробку таких дій (цей список не є повним):

- Отримання вебскриптів від зовнішніх служб (див. *Обробники сповіщень*).
- Виконання завдань регулярного супроводу, зокрема резервного копіювання, чищення, щоденного запуску додатків або оновлень (див. *Резервне копіювання і пересування Weblate*, *BACKGROUND_TASKS*, *Додатки*).
- Запуск *Автоматичний переклад*.
- Надсилання сповіщення-резюме.
- Розвантаження процесу wsgi від витратних обчислювальних дій.
- Надсилання змін з черги (див. *«Лінійні» внески*).

Типова конфігурація з використанням модуля обробки Redis виглядає так:

```

CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

```

Дивись також:

Налаштовування засобу розподілу навантаження Redis у Celery

Крім того, вам слід запустити обробник Celery для обробки завдань і запустити заплановані завдання. Зробити це можна безпосередньо з командного рядка (це здебільшого є корисним для діагностики або розробки):

```

./weblate/examples/celery start
./weblate/examples/celery stop

```

Примітка: Процес Celery має бути запущено від імені того ж користувача, що й процес WSGI, інакше файли в *DATA_DIR* буде збережено від імені різних власників, що призведе до проблем під час роботи сервера.

Див. також *Права доступу у файлової системі* і *Запуск сервера*.

Виконання завдань Celery у wsgi з використанням «жадібного» режиму

Примітка: Це матиме значний негативний вплив на швидкодію вебінтерфейсу і зашкодить роботі можливостей, залежно від частоти запуску (наприклад, для надсилання змін з черги, сповіщень дайджесту або резервного копіювання).

Для розробки варто скористатися інтенсивнішими налаштуваннями, за яких обробка усіх завдань відбувається на місці:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

Запуск Celery як служби системи

Найімовірніше, вам потрібен буде запуск Celery як фонової служби. Опис відповідних налаштувань можна знайти у розділі [Daemonization](#). Для найпоширенішої конфігурації Linux із використанням systemd ви можете скористатися файлами прикладів, які постачаються у теці `examples`. Список цих файлів наведено нижче.

Модуль `systemd`, який слід зберегти як `/etc/systemd/system/celery-weblate.service`:

```
[Unit]
Description=Celery Service (Weblate)
After=network.target

[Service]
Type=forking
User=weblate
Group=weblate
EnvironmentFile=/etc/default/celery-weblate
WorkingDirectory=/home/weblate
RuntimeDirectory=celery
RuntimeDirectoryPreserve=restart
LogsDirectory=celery
ExecStart=/bin/sh -c '${CELERY_BIN} multi start ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'
ExecStop=/bin/sh -c '${CELERY_BIN} multi stopwait ${CELERYD_NODES} \
  --pidfile=${CELERYD_PID_FILE}'
ExecReload=/bin/sh -c '${CELERY_BIN} multi restart ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'

[Install]
WantedBy=multi-user.target
```

Налаштування середовища, які слід зберегти як `/etc/default/celery-weblate`:

```
# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
CELERY_BIN="/home/weblate/weblate-env/bin/celery"

# App instance to use
# comment out this line if you don't use an app
CELERY_APP="weblate.utils"
```

(continues on next page)

(continued from previous page)

```
# Extra command-line arguments to the worker,
# increase concurrency if you get weblate.E019
CELERYD_OPTS="--beat:celery --queues:celery=celery --prefetch-multiplier:celery=4 \
--queues:notify=notify --prefetch-multiplier:notify=10 \
--queues:memory=memory --prefetch-multiplier:memory=10 \
--queues:translate=translate --prefetch-multiplier:translate=4 \
--concurrency:backup=1 --queues:backup=backup --prefetch-multiplier:backup=2"

# Logging configuration
# - %n will be replaced with the first part of the nodename.
# - %I will be replaced with the current child process index
# and is important when using the prefork pool to avoid race conditions.
CELERYD_PID_FILE="/run/celery/weblate-%n.pid"
CELERYD_LOG_FILE="/var/log/celery/weblate-%n%I.log"
CELERYD_LOG_LEVEL="INFO"
```

Додаткові налаштування для оновлення журналу Celery за допомогою **logrotate**, які слід зберегти у `/etc/logrotate.d/celery`:

```
/var/log/celery/*.log {
    weekly
    missingok
    rotate 12
    compress
    notifempty
}
```

Регулярні завдання з використанням тактів Celery

Weblate постачається із вбудованою конфігурацією для запланованих завдань. Втім, ви можете визначити додаткові завдання у файлі `settings.py`. Приклад наведено у розділі «*Лінії*» *внески*.

Виконання завдань має забезпечувати фонові служба регулярних подій Celery. Якщо ця служба не працює належним чином, її могло бути не запущено або її базу даних могло бути пошкоджено. Якщо виникають проблеми, ознайомтеся із журналом запуску Celery, щоб визначити причину.

Спостереження за станом Celery

Знайти поточну довжину черг завдань Celery у *Інтерфейс керування*. Ви також можете скористатися `celery_queues` у командному рядку. У випадку значного зростання довжини черги ви також побачите повідомлення про помилку налаштувань у адміністративному інтерфейсі.

Попередження: Повідомлення про помилки Celery, типово, записуються лише до журналу Celery і є невидимими для користувача. Якщо ви хочете бачити огляд помилок подібного типу, рекомендуємо вам налаштувати *Збирання звітів щодо помилок*.

Дивись також:

Спостереження за Weblate, Як перевірити, чи правильно налаштовано мій Weblate?, Configuration and defaults, Workers Guide, Daemonization, Monitoring and Management Guide, `celery_queues`

2.1.12 Спостереження за Weblate

У Weblate передбачено адресу `/healthz/` для простих перевірок працездатності, наприклад, за допомогою Kubernetes. У контейнера Docker є вбудована перевірка працездатності з використанням цієї адреси.

Для спостереження за метрикою Weblate ви можете скористатися кінцевою точкою програмного інтерфейсу `GET /api/metrics/`.

Дивись також:

Як перевірити, чи правильно налаштовано мій Weblate?, *Спостереження за станом Celery*, додаток Weblate для Munin

2.1.13 Збирання звітів щодо помилок

Weblate, як і будь-яке інше програмне забезпечення, може містити помилки. Для збирання корисних даних щодо критичних помилок ми рекомендуємо користуватися сторонніми службами, які можуть збирати такі дані. Це особливо корисно у випадку помилок, які пов'язано із завданнями Celery. Якщо не користуватися сторонніми засобами, від таких помилок лишається лише запис у журналі — ви не побачите сповіщення щодо них. У Weblate передбачено підтримку таких служб:

Sentry

У Weblate є вбудована підтримка Sentry. Щоб скористатися нею, достатньо встановити `SENTRY_DSN` у файлі `settings.py`:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

У Weblate передбачено вбудовану підтримку Rollbar. Щоб скористатися нею, достатньо виконати настанови для сповіщувача Rollbar для Python.

Якщо коротко, вам слід скоригувати `settings.py`:

```
# Add rollbar as last middleware:
MIDDLEWARE = [
    # ... other middleware classes ...
    "rollbar.contrib.django.middleware.RollbarNotifierMiddleware",
]

# Configure client access
ROLLBAR = {
    "access_token": "POST_SERVER_ITEM_ACCESS_TOKEN",
    "client_token": "POST_CLIENT_ITEM_ACCESS_TOKEN",
    "environment": "development" if DEBUG else "production",
    "branch": "main",
    "root": "/absolute/path/to/code/root",
}
```

Усе інше інтегровано автоматично, ви тепер збираєте помилки на боці сервера і на боці клієнта.

2.1.14 Перенесення Weblate на інший сервер

Перенесення Weblate на інший сервер має бути доволі простою процедурою. Втім, Weblate зберігає дані у декількох місцях, і ці дані слід перенести належним чином. Найкращим підходом є зупинення Weblate для перенесення даних.

Перенесення бази даних

Залежно від модуля обробки вашої бази даних, ви можете вибрати один із декількох варіантів перенесення бази даних. Найприроднішим з цих варіантів є використання власних інструментів бази даних, оскільки, зазвичай, вони є найефективнішими (приклад `mysqldump` або `pg_dump`). Ви також можете скористатися реплікацією, якщо у базі даних передбачено її підтримку.

Дивись також:

Перенесення даних між базами даних описано у розділі *Перенесення даних з інших баз даних до PostgreSQL*.

Перенесення репозиторіїв системи керування версіями

Репозиторії коду систем керування версіями, які зберігаються у `DATA_DIR`, слід також перенести. Ви можете просто скопіювати їх або скористатися командою `rsync` для виконання завдання з перенесення у ефективніший спосіб.

Інші нотатки

Не забудьте перенести інші служби, які Weblate може використовувати, зокрема Redis, завдання Cron і не-типові модулі розпізнавання.

2.2 Розгортання Weblate

Weblate можна без проблем встановити у вашій «хмарі». Будь ласка, ознайомтеся із докладними настановами для вашої платформи:

- *Установлення за допомогою Docker*
- *Установлення на OpenShift*
- *Установлення на Kubernetes*

2.2.1 Сторонні розгортання Weblate

Примітка: Вказані нижче розгорнуті версії Weblate не є розробленими або супроводжуваними командою Weblate. Частини налаштувань цих версій можуть відрізнятися від описаних у цій документації.

Стос Weblate Bitnami

У Bitnami передбачено стек Weblate для багатьох платформ у <<https://bitnami.com/stack/weblate>>.

Дивись також:

Weblate, який запаковано Bitnami

Пакунок Weblate Cloudron

Cloudron — платформа для самостійного розміщення мережних застосунків. Актуальність версії Weblate, яку встановлено з Cloudron, підтримуватиметься автоматично. Супровід пакунка здійснює команда Cloudron за допомогою власного репозиторію пакунків Weblate.



Weblate у YunoHost

Проект із власним сервером YunoHost надає пакунок із Weblate. Після встановлення YunoHost ви можете встановити Weblate як будь-яку іншу програму. У ваше розпорядження буде додано повністю працездатний стос програм із резервним копіюванням і відновленням даних, але файл параметрів може потребувати додаткового налаштування для забезпечення специфічних для вас потреб.

Ви можете скористатися вашим адміністративним інтерфейсом або цієї кнопкою (вона приведе вас до вашого сервера):



Також можна скористатися інтерфейсом командного рядка:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Оновлення Weblate

2.3.1 Оновлення образу Docker

До офіційного образу Docker (див. [install/docker](#)) інтегровано усі кроки оновлення. Кроків, які слід виконати вручну, окрім отримання коду найсвіжішої версії, немає.

Дивись також:

Оновлення контейнера Docker

2.3.2 Загальні настанови щодо оновлення

Перед оновленням, будь ласка, ознайомтеся із поточними *вимогами*, оскільки їх могло бути змінено. Щойно усі потрібні пакунки буде встановлено або оновлено, будь ласка, скоригуйте ваш `settings.py` так, щоб він відповідав змінам у налаштуваннях (зверніться до `settings_example.py`, щоб дізнатися більше про правильні значення).

Завжди ознайомлюйтеся з *настановами щодо специфіки версії* до оновлення. Якщо ви пропустили кілька попередніх версій, виконайте настанови для всіх версій, які ви пропускаєте під час оновлення. Іноді краще оновитися до проміжної версії, щоб забезпечити безпроблемне перенесення даних. Оновлення з пропусканням кількох випусків мають працювати, але їхня перевірка на коректність виконується набагато менш ретельно, ніж перевірка оновлень без пропуску версій.

Примітка: До оновлення рекомендуємо вам створити повну резервну копію бази даних, щоб ви могли повернутися до попереднього стану бази даних, якщо не вдасться виконати оновлення, див. *Резервне копіювання і пересування Weblate*.

1. Зупиніть процеси `wsgi` і `Celery`. Оновлення може призвести до несумісних змін у базі даних, тому завжди безпечніше, якщо під час оновлення застарілі процеси не працюватимуть.
2. Оновіть код Weblate.

Для встановлень за допомогою `pip` цього можна досягти такою командою:

```
pip install -U "Weblate[all]==version"
```

Або, якщо ви просто хочете отримати найсвіжішу випущену версію:

```
pip install -U "Weblate[all]"
```

Якщо ви не хочете встановлювати усі обов'язкові залежності, зробіть так:

```
pip install -U Weblate
```

Якщо ви працюєте із копією з `Git`, вам слід отримати новий початковий код і оновити встановлену у вас версію:

```
cd weblate-src
git pull
# Update Weblate inside your virtualenv
. ~/weblate-env/bin/pip install -e .
# Install dependencies directly when not using virtualenv
pip install --upgrade -r requirements.txt
# Install optional dependencies directly when not using virtualenv
pip install --upgrade -r requirements-optional.txt
```

3. У нового випуску Weblate можуть бути нові *необов'язкові залежності*. Будь ласка, перевірте, чи забезпечують вони роботу потрібних вам можливостей.
4. Оновіть файл налаштувань. Зверніться до файла `settings_example.py` або *специфічних для випуску настанов*, щоб ознайомитися із потрібними для цього діями.
5. Оновіть структуру бази даних:

```
weblate migrate --noinput
```

6. Зберіть оновлені статичні файли (див. *Запуск сервера* і *Обслуговування статичних файлів*):

```
weblate collectstatic --noinput --clear
```

7. Стисніть файли JavaScript і CSS (необов'язково, див. *Стиснення клієнтських даних*):


```
weblate compress
```

8. Якщо ви працюєте із версією з Git, вам слід також повторно створювати файли локалей кожного разу після оновлення. Зробити це можна за допомогою такої команди:

```
weblate compilemessages
```

9. Перевірте, чи є працездатною ваша конфігурація (див. також [Промислові налаштування](#)):

```
weblate check --deploy
```

10. Перезапустіть засіб обробки Celery (див. [Фонові завдання з використанням Celery](#)).

2.3.3 Специфічні для версій настанови

Оновлення з версії 2.x

Якщо ви оновлюєте систему з версії 2.x, завжди спочатку оновіться до версії 3.0.1, а потім продовжіть оновлення у межах послідовності версій 3.x. Оновлення, у яких пропущено цей крок, вважаються непідтримуваними — вони не працюватимуть як слід.

Дивись також:

[Документація із оновлення з 2.20 до 3.0 у документації до Weblate 3.0](#)

Оновлення з версії 3.x

Якщо ви оновлюєте систему з версії 3.x, завжди спочатку оновіться до версії 4.0.4 або 4.1.1, а потім продовжіть оновлення у межах послідовності версій 4.x. Оновлення, у яких пропущено цей крок, вважаються непідтримуваними — вони не працюватимуть як слід.

Дивись також:

[Документація із оновлення з 3.11 до 4.0 у документації до Weblate 4.0](#)

Оновлення з 4.0 до 4.1

Будь ласка, виконайте настанови із розділу [Загальні настанови щодо оновлення](#) для виконання оновлення.

Помітні зміни у налаштуваннях або залежностях:

- До `settings_example.py` внесено декілька змін. Найпомітнішими є зміна проміжного програмного забезпечення. Будь ласка, скоригуйте параметри відповідним чином.
- Передбачено нові формати файлів. Можливо, вам варто включити їх до списку зміненого [WEBLATE_FORMATS](#).
- Передбачено нові перевірки якості. Можливо, вам варто включити їх до списку зміненого [CHECK_LIST](#).
- Змінено параметр `DEFAULT_THROTTLE_CLASSES` так, щоб уможливити звітування щодо обмеження частоти у програмному інтерфейсі.
- Маємо деякі нові і оновлені вимоги щодо програмного забезпечення.
- Змінено [INSTALLED_APPS](#).
- Параметр `MT_DEEPL_API_VERSION` було вилучено у версії 4.7. Замість нього для рушія комп'ютерного перекладу [DeepL](#) використано новий параметр `MT_DEEPL_API_URL`. Можливо, вам доведеться скоригувати `MT_DEEPL_API_URL` відповідно до параметрів вашої передплати.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.1 до 4.2

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

Помітні зміни у налаштуваннях або залежностях:

- Підтримки оновлення з випусків 3.x більше не передбачено. Будь ласка, спочатку виконайте оновлення до версії 4.0 або 4.1.
- Маємо деякі нові і оновлені вимоги щодо програмного забезпечення.
- До `settings_example.py` внесено декілька змін. Найпомітнішими є нове проміжне програмне забезпечення та зміна порядку виконання застосунків.
- Ключі для заснованих на JSON форматів більше не включають початкової крапки. Рядки скориговано під час перенесення бази даних, але зовнішні складники можуть потребувати коригування, якщо ви покладаєтеся на ключі для експортування або використання програмного інтерфейсу.
- Налаштування Celery змінено так, що у них більше не використовується черга `memory`. Будь ласка, скоригуйте ваші скрипти запуску та значення параметра `CELERY_TASK_ROUTES`.
- Домен Weblate тепер налаштовується у параметрах, див. `SITE_DOMAIN` (або `WEBLATE_SITE_DOMAIN`). Вам слід буде налаштувати його до запуску Weblate.
- Поля імені користувача і адреси електронної пошти у базі даних користувачів тепер мають містити неповторювані текстові дані (регістр символів не береться до уваги). У попередніх версіях це обмеження не було застосовано примусово для PostgreSQL.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.2 до 4.3

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

Помітні зміни у налаштуваннях або залежностях:

- Передбачено певні зміни у перевірках якості. Можливо, вам варто включити їх до списку зміненого `CHECK_LIST`.
- Атрибут початкової мови було перенесено з проєкту до складника, який доступний у програмному інтерфейсі. Вам слід оновити *Клієнт Weblate*, якщо ви ним користуєтесь.
- Перенесення бази даних з версії 4.3 може бути тривалим. Тривалість залежить від кількості рядків, переклад яких виконується (очікуваний час для перенесення 100000 початкових рядків складає близько однієї години).
- Змінено `INSTALLED_APPS`.
- Нове налаштування `SESSION_COOKIE_AGE_AUTHENTICATED`, яке доповнює `SESSION_COOKIE_AGE`.
- Якщо ви користувалися **hub** або **lab** для інтеграції із GitHub або GitLab, вам слід переналаштувати параметри, див. `GITHUB_CREDENTIALS` і `GITLAB_CREDENTIALS`.

Змінено в версії 4.3.1:

- Налаштування Celery змінено так, щоб додати чергу `memory`. Будь ласка, скоригуйте ваші скрипти запуску та значення параметра `CELERY_TASK_ROUTES`.

Змінено в версії 4.3.2:

- Метод `post_update` у додатках тепер приймає додатковий параметр `skip_push`.

Дивись також:*Загальні настанови щодо оновлення***Оновлення з 4.3 до 4.4**

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

Помітні зміни у налаштуваннях або залежностях:

- Змінено `INSTALLED_APPS`, тут слід додавати `weblate.configuration`.
- Тепер потрібен Django 3.1.
- Якщо ви використовуєте MySQL або MariaDB, підвищено вимоги щодо мінімальних версій, див. *MySQL і MariaDB*.

Змінено в версії 4.4.1:

- *Одномовний gettext* у новій версії використовує одразу `msgid msgctxt`, якщо такі поля є. Це змінює ідентифікацію рядків перекладу у відповідних файлах розривання посилань на розширені дані Weblate, такі як знімки екрана або стани розгляду. Будь ласка, переконайтеся, що зміни з черги для таких файлів записано до репозиторію, перш ніж виконувати оновлення. Рекомендуємо примусово завантажити відповідний складник за допомогою *loadpo*.
- Збільшено мінімальну обов'язкову версію `translate-toolkit` для виправлення декількох проблем із форматами файлів.

Дивись також:*Загальні настанови щодо оновлення***Оновлення з 4.4 до 4.5**

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

Помітні зміни у налаштуваннях або залежностях:

- Перехід на нову версію може бути досить тривалим, якщо у вас доволі великі глосарії.
- Глосарії у новій версії зберігаються як звичайні складники.
- Вилучено програмний інтерфейс глосаріїв. Для доступу до глосаріїв скористайтесь звичайним програмним інтерфейсом перекладів.
- Змінено `INSTALLED_APPS`, тут слід додавати `weblate.metrics`.

Змінено в версії 4.5.1:

- Існує нова залежність від модуля *pyahocorasick*.

Дивись також:*Загальні настанови щодо оновлення*

Оновлення з 4.5 до 4.6

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

Помітні зміни у налаштуваннях або залежностях:

- Передбачено нові формати файлів. Можливо, вам варто включити їх до списку зміненого `WEBLATE_FORMATS`.
- У програмному інтерфейсі для створення складників тепер автоматично використовуються *Внутрішні адреси Weblate*, див. `POST /api/projects/(string:project)/components/`.
- Відбулися зміни у залежностях, і `PASSWORD_HASHERS` тепер пріоритетно використовує Argon2 для хешування паролів.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.6 до 4.7

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

Помітні зміни у налаштуваннях або залежностях:

- До `settings_example.py` внесено декілька змін. Найпомітнішими є зміна проміжного програмного забезпечення (`MIDDLEWARE`). Будь ласка, скоригуйте параметри відповідним чином.
- Для рушія комп'ютерного перекладу *DeepL* тепер використовують загальний параметр `MT_DEEPL_API_URL` з метою гнучкішої адаптації до різних моделей передплати. Використання параметра `MT_DEEPL_API_VERSION` припинено.
- Тепер потрібен Django 3.2.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.7 до 4.8

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

У цьому випуску ніяких додаткових кроків із оновлення виконувати не потрібно.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.8 до 4.9

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

- Метрику зберігання даних змінено. Оновлення для великих сайтів може тривати досить довго.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.9 до 4.10

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

- Роботу із окремими для проєкту групами змінено. Оновлення для сайтів із тисячами проєктів може тривати досить довго.
- До Django 4.0 внесено деякі несумісні із попередніми версіями зміни, див. [Backwards incompatible changes in 4.0](#). У поточній версії Weblate усе ще передбачено підтримку Django 3.2, якщо оновлення до сучасної версії є проблематичним. Серед найпомітніших змін, які можуть стосуватися Weblate, такі:
 - Відкинута підтримку PostgreSQL 9.6 — для роботи Django 4.0 потрібна PostgreSQL 10 або новіша версія.
 - Було змінено формат `CSRF_TRUSTED_ORIGINS`.
- У контейнері Docker нової версії використано Django 4.0. Опис змін наведено вище.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.10 до 4.11

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

- Для роботи Weblate з цієї версії потрібен Python 3.7 або новіша версія.
- Було змінено реалізацію *Керування правами доступу на рівні проєкту* — вилучено префікс проєкту з назв груп. Це стосується користувачів програмного інтерфейсу.
- У новій версії Weblate для виявлення набору символів використано модуль `charset-normalizer` замість модуля `chardet`.
- **Змінено у 4.11.1:** зміни у параметрі `REST_FRAMEWORK` (вилучення одно з модулів у `DEFAULT_AUTHENTICATION_CLASSES`).

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.11 до 4.12

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

- Ніяких особливих кроків роботи не треба.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.12 до 4.13

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

- При оновленні тепер буде автоматично оновлено *мови*; скористайтеся `UPDATE_LANGUAGES`, щоб відкрити оновлення.
- Обробку контексту і місця можна змінити для форматів файлів *Файли RC Windows*, *Файли HTML*, *Формат IDML* і *Текстові файли*. У більшості випадків, у новій версії контекст буде показано як місце.
- У новій версії служби комп'ютерного перекладу можна налаштувати з інтерфейсу користувача. Параметри з файла налаштувань буде імпортовано під час перенесення бази даних.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.13 до 4.14

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

- Перевірки форматування Java у новій версії відповідають прапорцям GNU gettext. Прапорці, які встановлено у Weblate буде автоматично перенесено, але для сторонніх скриптів вам доведеться скористатися `java-printf-format` замість `java-format` і `java-format` замість `java-messageformat`.
- Залежність *jellyfish* було замінено на *rapidfuzz*.
- **Змінено у 4.14.2:** Визнано застарілими небезпечні налаштування служб керування версіями за допомогою ключів програмного інтерфейсу та `_TOKEN/_USERNAME` замість списку `_CREDENTIALS`. У Docker, будь ласка, додайте відповідну інструкцію `_HOST`. Приклади наведено для `WEBLATE_GITHUB_HOST` та `GITHUB_CREDENTIALS`.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.14 до 4.15

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

- Для роботи нової версії Weblate потрібне розширення `btree_gin` у PostgreSQL. Під час процедури перенесення його буде встановлено, якщо процес має достатні привілеї. Див. *Створення бази даних у PostgreSQL* щодо налаштування вручну.
- В образі для Docker більше типово не вмикається режим діагностики. Якщо він вам потрібен, увімкніть його за допомогою змінної середовища `WEBLATE_DEBUG`.
- Перенесення бази даних може тривати годинами у великих екземплярах через потребу у повторному створенні деяких з покажчиків.
- **Змінено у 4.15.1:** було змінено типове значення для `DEFAULT_PAGINATION_CLASS` у решті параметрів бібліотек.

Дивись також:

Загальні настанови щодо оновлення

Оновлення з 4.15 до 4.16

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

- У новій версії биття Celery зберігає план завдань у базі даних. Для цього слід змінити `CELERY_BEAT_SCHEDULER` і `INSTALLED_APPS`.
- Припинено підтримку застарілого параметра системи керування версіями для реєстраційних даних, див. *Оновлення з 4.13 до 4.14*.
- Оновлення *django-crispy-forms* потребує змін у `INSTALLED_APPS`.
- Інтеграція *django-cors-headers* потребує змін у `INSTALLED_APPS` і `MIDDLEWARE`.

Дивись також:

Загальні настанови щодо оновлення

2.3.4 Оновлення з Python 2 до Python 3

У Weblate більше не передбачено підтримки версій Python, які є старішими за 3.6. Якщо ви все ще працюєте із застарілою версією, будь ласка, виконайте спочатку перенесення даних до Python 3 на наявній версії, а потім виконайте оновлення Weblate. Див. [Оновлення з Python 2 до Python 3 у документації до Weblate 3.11.1](#).

2.3.5 Перенесення даних з інших баз даних до PostgreSQL

Якщо ви працюєте з Weblate на основі бази даних, відмінної від PostgreSQL, вам слід перенести базу даних до PostgreSQL, оскільки із ним Weblate працює оптимально. Нижче наведено список кроків, які слід виконати для перенесення ваших даних з однієї бази даних до іншої. Будь ласка, не забудьте зупинити вебсервер та сервер Celery перед перенесенням даних, інакше дані може бути пошкоджено.

Створення бази даних у PostgreSQL

Зазвичай, варто зробити так, щоб Weblate працював із окремою базою даних і окремим обліковим записом користувача:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser -D -P weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Перенесення даних за допомогою дампів JSON Django

Найпростішим підходом до перенесення є використання дампів JSON Django. Такий спосіб добре працює для малих встановлень. Для більших сайтів вам варто скористатися pgloader, див. [Перенесення даних до PostgreSQL за допомогою pgloader](#).

1. Додайте PostgreSQL як додаткове з'єднання із базою даних до `settings.py`:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
        # Additional database options
        "OPTIONS": {
            # In case of using an older MySQL server, which has MyISAM as a
            ↪ default storage
            # 'init_command': 'SET storage_engine=INNODB',
            # Uncomment for MySQL older than 5.7:
            # 'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",
            # If your server supports it, see the Unicode issues above
            "charset": "utf8mb4",
```

(continues on next page)

(continued from previous page)

```

        # Change connection timeout in case you get MySQL gone away error:
        "connect_timeout": 28800,
    },
},
"postgresql": {
    # Database engine
    "ENGINE": "django.db.backends.postgresql",
    # Database name
    "NAME": "weblate",
    # Database user
    "USER": "weblate",
    # Database password
    "PASSWORD": "password",
    # Set to empty string for localhost
    "HOST": "database.example.com",
    # Set to empty string for default
    "PORT": "",
},
}

```

2. Запустіть перенесення і відкиньте усі дані, які було вставлено до таблиць:

```

weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --database=postgresql

```

3. Створіть дамп застарілої бази даних а імпортуйте його до PostgreSQL

```

weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql

```

4. Скоригуйте `DATABASES` так, щоб типово використовувалася лише база даних PostgreSQL, вилучіть застаріле з'єднання.

Після цього Weblate має бути готовим для запуску з базою даних PostgreSQL.

Перенесення даних до PostgreSQL за допомогою pgloader

pgloader — типовий інструмент для перенесення даних до PostgreSQL. Ви можете скористатися ним для перенесення бази даних Weblate.

1. Скоригуйте ваш `settings.py` так, щоб у ньому було використано базу даних PostgreSQL.
2. Перенесіть схему до бази даних PostgreSQL:

```

weblate migrate
weblate sqlflush | weblate dbshell

```

3. Запустіть **pgloader** для перенесення даних. Наведений нижче скрипт може бути використано для перенесення бази даних, але вам варто докладніше ознайомитися із **pgloader**, щоб зрозуміти принципи його роботи і скоригувати роботу так, щоб вона відповідала вашій конфігурації:

```

LOAD DATABASE
FROM      mysql://weblate:password@localhost/weblate
INTO      postgresql://weblate:password@localhost/weblate

WITH include no drop, truncate, create no tables, create no indexes, no_
↪foreign keys, disable triggers, reset sequences, data only

ALTER SCHEMA 'weblate' RENAME TO 'public'
;

```


2.3.6 Перенесення даних з Pootle

Перші версії Weblate було написано як заміник Pootle, тому у програмі передбачено підтримку перенесення облікових записів з Pootle. Ви можете створити дамп записів користувачів з Pootle і імпортувати їх за допомогою `importusers`.

2.4 Резервне копіювання і пересування Weblate

2.4.1 Резервне копіювання на рівні проєктів

Нове в версії 4.14.

Попередження: Підтримку відновлення даних з резервних копій передбачено лише для баз даних PostgreSQL або MariaDB 10.5+.

До резервних копій проєктів включено усі дані перекладу з Weblate (проєкт, складники, переклади, коментарі до рядків, пропозиції та перевірки). Такі резервні копії зручні для перенесення проєктів на інші екземпляри Weblate.

Створити резервну копію проєкту можна за допомогою пункту *Керування* ↓ *Резервні копії*. Відновити дані з резервної копії можна під час створення проєкту (див. *Додавання проєктів і складників перекладу*).

До поточних версій резервних копій не входять дані щодо керування доступом та дані журналу.

Коментарі і пропозиції потрапляють до резервної копії разом із даними щодо користувача, який їх створив. Під час імпортування програма пов'яже їх із відповідним користувачем. Якщо не буде виявлено користувача із відповідною назвою облікового запису, дані буде пов'язано із анонімним користувачем.

Створені резервні копії зберігатимуться на сервері, який визначається змінними `PROJECT_BACKUP_KEEP_DAYS` і `PROJECT_BACKUP_KEEP_COUNT` (типовим є зберігання не більше 3 резервних копій протягом 30 днів).

2.4.2 Автоматичне створення резервних копій за допомогою BorgBackup

Нове в версії 3.9.

У Weblate передбачено вбудовану підтримку створення резервних копій за допомогою `BorgBackup`. Borg створює шифровані резервні копії мінімального розміру, які можна безпечно зберігати у «хмарі». Керувати резервним копіюванням можна за допомогою інтерфейсу керування на вкладці *Резервні копії*.

Змінено в версії 4.4.1: До автоматичного резервного копіювання включено і базу даних PostgreSQL, і базу даних MySQL/MariaDB.

Резервні копії з використанням Borg є нарощувальними, а Weblate налаштовування на збереження таких резервних копій:

- Щоденне резервне копіювання до 14 днів
- Щотижневе резервне копіювання до 8 тижнів
- Щомісячне резервне копіювання до 6 місяців

Weblate
 Dashboard Projects Languages Checks

Manage / Backups

Backup process triggered

Weblate status
 Backups
 Translation memory
 Performance report
 SSH keys
 Alerts
 Repositories
 Users
 Teams

Appearance
 Tools
 Automatic suggestions
 Billing

Backup service: /tmp/tmp8o1nk_tnweblate

Backup service credentials
 March 1, 2023

Backup repository
 /tmp/tmp8o1nk_tnweblate

Passphrase
 2zwuwVXXKVI5dr\$pyvuga fL6FwBdzr6CYBTUXwT\$z^OmHO&gc^
 The passphrase is used to encrypt the backups and is necessary to restore them.

SSH key
 Download private key
 The private key is needed to access the remote backup repository.

Deleted the oldest backups
 March 1, 2023

Backup performed
 March 1, 2023

Repository initialization
 March 1, 2023

Turn off
 Perform backup
 Delete

Activate support package

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

 Please enter the activation token obtained when making the subscription.

Activate
 Purchase support package

Add backup service

Backup repository URL

 Use /path/to/repo for local backups or user@host:/path/to/repo or ssh://user@host:port/path/to/backups for remote SSH backups.

Add

Powered by Weblate 4.16
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

Ключ шифрування Borg

BorgBackup створює зашифровані резервні копії. Без пароля ви не зможете відновити дані з резервної копії. Пароль створюється при додаванні нової служби резервного копіювання, вам слід скопіювати його і зберігати його у безпечному місці.

Якщо ви користуєтесь *Передбачене у Weblate сховище резервних копій*, будь ласка, включіть до резервного копіювання і ваш закритий ключ SSH — його буде використано для доступу до ваших резервних копій.

Дивись також:

`borg init`

Налаштовування резервного копіювання

- Резервне копіювання бази даних можна налаштувати за допомогою `DATABASE_BACKUP`.
- Створення резервних копій можна налаштувати за допомогою `BORG_EXTRA_ARGS`.

2.4.3 Передбачене у Weblate сховище резервних копій

Найпростішим способом резервного копіювання вашого екземпляра Weblate є придбання служби резервного копіювання на weblate.org. Активацію можна виконати за декілька кроків:

1. Придбайте обслуговування із резервним копіюванням на <https://weblate.org/support/#backup>.
2. Введіть отриманий ключ в інтерфейсі керування, див. *Інтегрування підтримки*.
3. Weblate з'єднається із «хмарною» службою і отримати дані для доступу до резервних копій.
4. Увімкніть нові налаштування резервного копіювання на вкладці *Резервні копії*.
5. Створіть резервну копію реєстраційних даних Borg з метою уможливлення відновлення резервних копій, див. *Ключ шифрування Borg*.

Підказка: Крок вмикання усього вручну додано з метою забезпечення ваших даних. Без вашої згоди під час процедури реєстрації до репозиторію резервних копій не буде надіслано жодних даних.

2.4.4 Використання нетипового сховища резервних копій

Ви також можете скористатися власним сховищем для резервних копій. Для збереження резервних копій на віддаленому сервері можна скористатися SSH. На сервері сховища копій має бути встановлено **BorgBackup**.

Дивись також:

General у документації до Borg

Локальна файлова система

Рекомендуємо вказати абсолютний шлях до локальної резервної копії, наприклад `/шлях/до/резервної/копії`. Запис до каталогу має бути відкрито для користувача, від імені якого запущено Weblate (див. *Права доступу у файловій системі*). Якщо каталогу не існуватиме, Weblate спробує його створити, але програма повинна мати для цього відповідні права доступу.

Підказка: Якщо ви працюєте з Weblate у Docker, переконайтеся, що місце зберігання резервних копій відкрито як том з контейнера Weblate. Якщо ви цього не зробите, резервні копії буде втрачено після перезапуску контейнера Docker, в якому їх розміщено.

Одним із варіантів дій є розташування резервних копій на наявних томах. Наприклад, можна вибрати `/app/data/borgbackup`. Це наявний том у контейнері.

Ви також можете додати новий контейнер для резервних копій у файлі Docker Compose та скористатися, наприклад, `/borgbackup`:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

Каталог, де зберігатимуться резервні копії, повинен належати UID 1000, інакше Weblate не зможе записати їх туди.

Віддалені резервні копії

Для створення віддалених резервних копій вам слід встановити на іншому доступному з вашого екземпляра Weblate за допомогою SSH сервері [BorgBackup](#) з використанням ключа SSH Weblate:

1. Приготуйте сервер, на якому зберігатимуться ваші резервні копії.
2. Установіть на ньому сервер SSH (такий сервер типово встановлено у більшості дистрибутивів Linux).
3. Установіть на цьому сервері [BorgBackup](#); у більшості дистрибутивів Linux є відповідні пакунки (див. [Installation](#)).
4. Виберіть наявний запис користувача або створіть запис користувача, який використовуватиметься для резервного копіювання.
5. Додайте ключ SSH Weblate до запису користувача, щоб Weblate міг встановлювати з'єднання SSH із сервером без пароля (див. [Ключ SSH Weblate](#)).
6. Налаштуйте розташування резервних копій у Weblate як `користувач@вузол:/шлях/до/резервних/копій` або `ssh://користувач@вузол:порт/шлях/до/резервних/копій`.

Підказка: *Передбачене у Weblate сховище резервних копій* надає вам доступ до автоматичного створення резервних копій на віддалених серверах без жодних зайвих зусиль.

Дивись також:

[Ключ SSH Weblate](#), [General](#)

2.4.5 Відновлення з BorgBackup

1. Відновлення доступу до вашого репозиторію резервних копій і приготування вашого пароля до резервних копій.
2. Отримайте список усіх резервних копій на сервері за допомогою команди `borg list РЕПОЗИТОРІЙ`.
3. Відновіть бажану резервну копію до поточного каталогу за допомогою команди `borg extract РЕПОЗИТОРІЙ::АРХІВ`.
4. Відновіть базу даних із дампу SQL із розташування у каталозі `backup` каталогу даних Weblate (див. [Дампи даних для резервних копій](#)).
5. Скопіюйте налаштування Weblate (`backups/settings.py`, див. [Дампи даних для резервних копій](#)) у відповідне місце, див. [Коригування налаштувань](#).

Якщо застосовується контейнер Docker, файл налаштувань вже включено у контейнер і ви повинні відновити початкові змінні середовища. Файл `environment.yml` може допомогти вам у цьому (див. [Дампи даних для резервних копій](#)).

6. Скопіюйте увесь каталог відновлених даних до місця, яке налаштовано змінною `DATA_DIR`.

Якщо застосовується контейнер Docker, помістіть дані до тому даних, див. [Тому контейнера Docker](#).

Будь ласка, переконайтеся у тому, що визначено належних власників і права доступу до файлів, див. [Права доступу у файловій системі](#).

Сеанс Borg може виглядати ось так:

```
$ borg list /tmp/xxx
Enter passphrase for key /tmp/xxx:
2019-09-26T14:56:08                               Thu, 2019-09-26 14:56:08
↪ [de0e0f13643635d5090e9896bdaceb92a023050749ad3f3350e788f1a65576a5]
$ borg extract /tmp/xxx::2019-09-26T14:56:08
Enter passphrase for key /tmp/xxx:
```

Дивись також:

`borg list`, `borg extract`

2.4.6 Резервне копіювання вручну

Залежно від того, що саме ви хочете зберегти, створіть резервні копії типів даних, які Weblate зберігає у відповідних каталогах.

Підказка: Якщо ви виконуєте резервне копіювання вручну, вам варто увімкнути попередження Weblate щодо відсутності резервних копій додаванням `weblate.I028` до `SILENCED_SYSTEM_CHECKS` у `settings.py` або `WEBLATE_SILENCED_SYSTEM_CHECKS` для Docker.

```
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
```

База даних

Справжнє розташування сховища залежить від ваших налаштувань бази даних.

Підказка: Найважливішим сховищем даних є база даних. Налаштуйте регулярне резервне копіювання вашої бази даних. Без бази даних усі переклади буде втрачено.

Власна система резервного копіювання бази даних

Рекомендованим способом є створення дампу бази даних, за допомогою вбудованих інструментів засобу керування базою даних, зокрема `pg_dump` або `mysqldump`. Такий інструмент, зазвичай, працює краще за засіб резервне копіювання Django і може відновлювати таблиці з усіма даними.

Ви можете відновити цю резервну копію у новішому випуску Weblate — усі необхідні процедури із перенесення даних буде виконано під час запуску `migrate`. Будь ласка, зверніться до розділу [Оновлення Weblate](#), щоб ознайомитися із докладнішими відомостями щодо оновлення версій.

Резервне копіювання бази даних Django

Ви також можете створити резервну копію бази даних за допомогою команди `dumpdata` Django. Резервне копіювання у такий спосіб не прив'язане до засобу керування базою даних — ним можна скористатися, якщо ви хочете змінити модуль обробки бази даних.

Перш ніж відновлювати базу даних, вам слід встановити і запустити точну ту саму версію Weblate, що і на сервері, на якому виконувалося резервне копіювання. Це необхідно, оскільки структура бази даних змінюється між випусками, отже, використання неналежної версії може призвести до певного пошкодження даних. Після встановлення тієї самої версії запустіть усі процедури перенесення бази даних за допомогою команди `migrate`.

Щойно встановлення буде виконано, у базі даних з'являться деякі записи, які вже є у резервній копії. Рекомендуємо вилучити такі записи вручну за допомогою оболонки керування (див. [Виклик команд керування](#)):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Файли

Якщо у вас достатньо місця, рекомендуємо просто створити резервну копію усіх даних у `DATA_DIR`. Це найбезпечніший варіант, навіть якщо при цьому до резервної копії потраплять непотрібні вам файли. У наступних розділах докладно описано те, що має потрапити до резервної копії, і дані, які можна безпечно пропустити при резервному копіюванні.

Дампи даних для резервних копій

Змінено в версії 4.7: Дамп середовища було додано як `environment.yml`, щоб допомогти у відновленні середовищ Docker.

Зберігається у `DATA_DIR/backups`.

Weblate створює тут дамп різноманітних даних, і ви можете включити ці файли для отримання повнішої резервної копії. Файли оновлюються щоденно (це потребує запуску сервера Celery, див. [Фонові завдання з використанням Celery](#)). У поточній версії сюди включено:

- Параметри Weblate у файлі `settings.py` (розширена версія зберігається у `settings-expanded.py`).
- Резервна копія бази даних PostgreSQL у файлі `database.sql`.
- Дамп середовища як `environment.yml`.

Типово, резервні копії зберігаються у форматі простого тексту. Втім, її можна стиснути або повністю пропустити за допомогою параметра `DATABASE_BACKUP`.

Щоб відновити бази даних з резервної копії, завантажте її за допомогою інструментів бази даних, наприклад:

```
psql --file=database.sql weblate
```

Репозиторії систем керування версіями

Зберігається у `DATA_DIR / vcs`.

У репозиторіях систем керування версіями зберігаються копії базових репозиторіїв зі змінами, які внесено у Weblate. Якщо для всіх ваших складників перекладу ввімкнено *негайний запис до репозиторію після внеску*, усі зміни з Weblate буде включено до основного репозиторію, і вам не потрібно буде створювати резервні копії репозиторіїв на боці Weblate. Репозиторії можна буде просто знову клонувати з основного репозиторію без втрати даних.

Ключі SSH і GPG

Зберігається у `DATA_DIR / ssh` і `DATA_DIR / home`.

Якщо ви користуєтеся ключами SSH або GPG, які створено Weblate, вам слід створити резервні копії цих даних, інакше ви можете втратити закриті ключі, і вам доведеться створювати нові.

Вивантажені користувачем файли

Зберігаються у `DATA_DIR / media`.

Вам слід створити резервну копію вивантажених файлів (наприклад, *Візуальний контекст для рядків*).

Завдання Celery

У черзі завдань Celery можуть міститися корисні дані, але, зазвичай, створення їхньої резервної копії не виправдано. У найгіршому випадку ви втратите оновлення пам'яті перекладів, які ще не було оброблено. Такі дані рекомендовано оновлювати за цілими текстами та репозиторіями під час відновлення, тому ніяких проблем із втратою черги завдань не повинно бути.

Дивись також:

Фонові завдання з використанням Celery

Команда для резервного копіювання вручну

За допомогою завдання cron ви можете наказати системі виконувати щодня команду bash. Приклад:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz \
↳ backups vcs ssh home media fonts secret
```

За допомогою рядка у лапках після `XZ_OPT` ви можете вибрати параметри xz, наприклад об'єм пам'яті для стиснення; див. <https://linux.die.net/man/1/xz>

Ви можете скоригувати список тек і файлів за вашими потребами. Наприклад, щоб уникнути зберігання пам'яті перекладів (у теці резервних копій), ви можете скористатися таким:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz \
↳ backups/database.sql backups/settings.py vcs ssh home media fonts secret
```

2.4.7 Відновлення зі створеної вручну резервної копії

1. Відновлення усіх даних зі створеної вами резервної копії.
2. Оновлення усіх репозиторіїв за допомогою `updategit`.

```
weblate updategit --all
```

2.4.8 Пересування встановленого екземпляра Weblate

Перенесіть ваш екземпляр на іншу систему за допомогою настанов зі створення і відновлення резервних копій, які наведено вище.

Дивись також:

Оновлення з Python 2 до Python 3, Перенесення даних з інших баз даних до PostgreSQL

2.5 Розпізнавання

2.5.1 Реєстрація користувача

За типових налаштувань Weblate використовує `python-social-auth`, форму на сайті для обробки реєстрації нових користувачів. Після підтвердження адреси електронної пошти новий користувач може робити внески або проходити розпізнавання за допомогою однієї зі сторонніх служб.

Ви також можете вимкнути реєстрацію нових користувачів за допомогою параметра `REGISTRATION_OPEN`.

Спроби розпізнавання описано у *Обмеження частоти*.

2.5.2 Модулі розпізнавання

Для розпізнавання використовується вбудоване рішення Django, у якому передбачено різноманітні варіанти розпізнавань для соціальних мереж. Використання цього рішення означає, що ви можете імпортувати базу даних користувачів інших заснованих на Django проєктів (див. *Перенесення даних з Pootle*).

Django можна додатково налаштувати для розпізнавання в інших системах розпізнавання.

Дивись також:

У розділі *Параметри розпізнавання* описано, як налаштувати розпізнавання у офіційному образі Docker.

2.5.3 Розпізнавання за соціальними мережами

Завдяки [Welcome to Python Social Auth's documentation!](#), у Weblate передбачено підтримку розпізнавання за допомогою сторонніх служб, зокрема GitLab, Ubuntu, Fedora тощо.

Будь ласка, ознайомтеся із документацією щодо типових настанов з налаштування [Django Framework](#).

Примітка: Типово, Weblate покладається на сторонні служби розпізнавання для забезпечення коректності адреси електронної пошти. Якщо у якихось із служб, якими ви хочете скористатися, підтримки перевірки коректності адреси електронної пошти не передбачено, будь ласка, скористайтеся примусовою перевіркою адреси електронної пошти на боці Weblate, налаштувавши для таких служб параметр `FORCE_EMAIL_VALIDATION`. Приклад:

```
SOCIAL_AUTH_OPENSEUSE_FORCE_EMAIL_VALIDATION = True
```


Дивись також:[Pipeline](#)

Вмикання окремих модулів обробки є доволі простою справою — достатньо додати запис до параметра `AUTHENTICATION_BACKENDS` і, можливо, додати ключі, потрібні для вказаного способу розпізнавання. Зауважте, що у деяких модулях обробки типово не надається параметра адреси електронної пошти користувача — вам слід надсилати запит щодо неї явним чином, інакше Weblate не зможе належним чином зберігати авторські права учасників перекладу.

Підказка: Більшість серверів автентифікації вимагається протокол HTTPS. Після увімкнення HTTPS на вашому вебсервері налаштуйте Weblate, щоб повідомити про це належним чином, застосувавши `ENABLE_HTTPS`, або за допомогою `WEBLATE_ENABLE_HTTPS` у контейнері Docker.

Дивись також:[Модуль розпізнавання за допомогою соціальних мереж у Python](#)**Розпізнавання за OpenID**

Для служб на основі OpenID, зазвичай, достатньо просто вмикання. Вказаний нижче розділ вмикає розпізнавання за OpenID для систем OpenSUSE, Fedora і Ubuntu:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.suse.OpenSUSEOpenId",
    "social_core.backends.ubuntu.UbuntuOpenId",
    "social_core.backends.fedora.FedoraOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

Дивись також:[OpenID](#)**Розпізнавання за GitHub**

Вам слід зареєструвати програму OAuth на GitHub, а потім повідомити Weblate усі її реєстраційні дані:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.github.GithubOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = "GitHub Client ID"
SOCIAL_AUTH_GITHUB_SECRET = "GitHub Client Secret"
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
```

GitHub слід налаштувати, щоб адреса зворотного виклику була подібно до `https://example.com/accounts/complete/github/`.

Існують подібні модулі розпізнавання для «GitHub для організацій» та «GitHub для команд». Їхні параметри мають назви `SOCIAL_AUTH_GITHUB_ORG_*` і `SOCIAL_AUTH_GITHUB_TEAM_*`. Ці модулі потребують додаткових налаштувань області видимості — `SOCIAL_AUTH_GITHUB_ORG_NAME`

або `"SOCIAL_AUTH_GITHUB_TEAM_ID"`. Відповідними адресами зворотних викликів є `https://example.com/accounts/complete/github-org/` і `https://example.com/accounts/complete/github-teams/`.

Примітка: Надана Weblate адреса зворотного виклику під час розпізнавання містить дані щодо налаштованого домену. Якщо система повідомляє вам про помилки щодо невідповідності, вам варто внести виправлення, див. [Установіть належний домен сайта](#).

Дивись також:

GitHub

Розпізнавання за Bitbucket

Вам слід зареєструвати програму на Bitbucket, а потім повідомити Weblate усі її реєстраційні дані:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.bitbucket.BitbucketOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = "Bitbucket Client ID"
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = "Bitbucket Client Secret"
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True
```

Примітка: Надана Weblate адреса зворотного виклику під час розпізнавання містить дані щодо налаштованого домену. Якщо система повідомляє вам про помилки щодо невідповідності, вам варто внести виправлення, див. [Установіть належний домен сайта](#).

Дивись також:

Bitbucket

Google OAuth 2

Щоб скористатися OAuth 2 Google, вам слід зареєструвати програму на <https://console.developers.google.com/> і увімкнути програмний інтерфейс Google+.

Адреса переспрямування — `https://CEPBEP` `WEBLATE/accounts/complete/google-oauth2/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"
```

Примітка: Надана Weblate адреса зворотного виклику під час розпізнавання містить дані щодо налаштованого домену. Якщо система повідомляє вам про помилки щодо невідповідності, вам варто внести виправлення, див. [Установіть належний домен сайта](#).

Дивись також:

[Google](#)

Facebook OAuth 2

Як звично для служб OAuth 2, вам слід зареєструвати вашу програму за допомогою Facebook. Після цього, вам слід налаштувати Weblate на її використання:

Адреса переспрямування — `https://СЕРВЕР WEBLATE/accounts/complete/facebook/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
```

Примітка: Надана Weblate адреса зворотного виклику під час розпізнавання містить дані щодо налаштованого домену. Якщо система повідомляє вам про помилки щодо невідповідності, вам варто внести виправлення, див. [Установіть належний домен сайта](#).

Дивись також:

[Facebook](#)

GitLab OAuth 2

Для користування GitLab OAuth 2 вам слід зареєструвати програму на <https://gitlab.com/profile/applications>.

Адреса переспрямування — `https://СЕРВЕР WEBLATE/accounts/complete/gitlab/` і не забудьте позначити область видимості `read_user`.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.gitlab.GitLabOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITLAB_KEY = "Application ID"
SOCIAL_AUTH_GITLAB_SECRET = "Secret"
SOCIAL_AUTH_GITLAB_SCOPE = ["read_user"]

# If you are using your own GitLab
# SOCIAL_AUTH_GITLAB_API_URL = 'https://gitlab.example.com/'
```

Примітка: Надана Weblate адреса зворотного виклику під час розпізнавання містить дані щодо налаштованого домену. Якщо система повідомляє вам про помилки щодо невідповідності, вам варто внести виправлення, див. [Установіть належний домен сайта](#).

Дивись також:

GitLab

Microsoft Azure Active Directory

Weblate може бути налаштовано на використання типових або специфічних власників системи розпізнавання.

Адресою переспрямування є `https://СЕРВЕР WEBLATE/accounts/complete/azuread-oauth2/` для звичайного і `https://СЕРВЕР WEBLATE/accounts/complete/azuread-tenant-oauth2/` для специфічного для орендаря розпізнавання.

```
# Azure AD common

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread.AzureADOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET = ""
```

```
# Azure AD Tenant

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread_tenant.AzureADTenantOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET = ""
# Tenant ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID = ""
```

Примітка: Надана Weblate адреса зворотного виклику під час розпізнавання містить дані щодо налаштованого домену. Якщо система повідомляє вам про помилки щодо невідповідності, вам варто внести виправлення, див. [Установіть належний домен сайта](#).

Дивись також:

Microsoft Azure Active Directory

Ідентифікатор

Для користування Slack OAuth 2 вам слід зареєструвати програму на <<https://api.slack.com/apps>>.

Адреса переспрямування — `https://СЕРВЕР WEBLATE/accounts/complete/slack/`.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.slack.SlackOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SLACK_KEY = ""
SOCIAL_AUTH_SLACK_SECRET = ""
```

Примітка: Надана Weblate адреса зворотного виклику під час розпізнавання містить дані щодо налаштованого домену. Якщо система повідомляє вам про помилки щодо невідповідності, вам варто внести виправлення, див. *Установіть належний домен сайта*.

Дивись також:

[Slack](#)

Перевизначення імен методів автентифікації та піктограм

Коротке видиме ім'я та піктограму методу автентифікації можна змінити за допомогою параметрів `SOCIAL_AUTH_<NAME>_IMAGE` і `SOCIAL_AUTH_<NAME>_TITLE`. Наприклад, перевизначення імені для Auth0 має такий вигляд:

```
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

Вимикання розпізнавання за паролем

Розпізнавання за адресою електронної пошти і паролем можна вимкнути вилученням запису `social_core.backends.email.EmailAuth` з `AUTHENTICATION_BACKENDS`. Не вилучайте `weblate.accounts.auth.WeblateUserBackend` — цей запис потрібне для реалізації основних функціональних можливостей Weblate.

Вимикання розпізнавання за електронною поштою призведе до вимикання усіх функціональних можливостей, які пов'язано з електронною поштою: запрошення користувачів або можливості скидання паролів.

Порада: Ви все ще можете скористатися для створених тут вручну записів користувачів розпізнаванням за паролями для адміністративного інтерфейсу. Просто перейдіть до `/admin/login/`.

Наприклад, розпізнаванням з використанням лише надавача даних Open ID openSUSE Open ID можна скористатися ось так:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Розпізнавання за паролем

Типовий `settings.py` постачається із достатнім набором `AUTH_PASSWORD_VALIDATORS`:

- Паролі не можуть бути надто подібними до інших ваших особистих даних.
- Паролі мають складатися із принаймні 10 символів.
- Пароль не може бути паролем загального вжитку.
- Пароль не може складатися лише з цифр.
- Паролі не можуть складатися лише з одного символу або лише з пробілів.
- Паролі не можуть збігатися із паролями, які ви використовували раніше.

Ви можете налаштувати цей параметр так, щоб він відповідав вашим правилам щодо паролів.

Крім того, ви можете встановити `django-zxcvbn-password` — додаток, який дає доволі реалістичну оцінку складності пароля та надає змогу відмовляти у реєстрації паролів, які не є достатньо складними.

2.5.5 Розпізнавання за SAML

Нове в версії 4.1.1.

Будь ласка, виконайте настанови щодо налаштування розпізнавання за соціальними мережами у Python. Відмінності:

- У Weblate передбачено підтримку єдиного IDP, який має назву `weblate`, у `SOCIAL_AUTH_SAML_ENABLED_IDPS`.
- Адреса метаданих XML SAML — `/accounts/metadata/saml/`.
- Автоматично заповнено буде такі параметри: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT`, `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Приклад налаштувань:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.saml.SAMLAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SAML_SP_ENTITY_ID = f"https://{SITE_DOMAIN}/accounts/metadata/saml/"
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbn1+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
SOCIAL_AUTH_SAML_ORG_INFO = {
    "en-US": {
        "name": "example",
        "displayname": "Example Inc.",
        "url": "http://example.com"
    }
}
```

(continues on next page)

(continued from previous page)

```

}
SOCIAL_AUTH_SAML_TECHNICAL_CONTACT = {
    "givenName": "Tech Gal",
    "emailAddress": "technical@example.com"
}
SOCIAL_AUTH_SAML_SUPPORT_CONTACT = {
    "givenName": "Support Guy",
    "emailAddress": "support@example.com"
}

```

За типових налаштувань параметри запису користувача видобувають із вказаних нижче атрибутів. Налаштуйте ваш IDP на надання таких параметрів:

Атрибут	Довідник щодо адрес SAML
Повне ім'я	urn:oid:2.5.4.3
Ім'я	urn:oid:2.5.4.42
Прізвище	urn:oid:2.5.4.4
Електронна пошта	urn:oid:0.9.2342.19200300.100.1.3
Ім'я користувача	urn:oid:0.9.2342.19200300.100.1.1

Підказка: Наведений вище приклад і образ Docker визначають IDP із назвою `weblate`. Можливо, вам слід налаштувати цей рядок як *Relay* у вашому IDP.

Дивись також:

Налаштовування SAML у Docker, SAML

2.5.6 Розпізнавання за допомогою LDAP

Оптимальним способом уможливлення розпізнавання за допомогою LDAP є використання пакунка *django-auth-ldap*. Встановити його можна у звичний спосіб:

```

# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap

```

Підказка: Цей пакунок включено до контейнера Docker, див. *Установлення за допомогою Docker*.

Примітка: Маємо певні несумісності із модулем Python LDAP 3.1.0, які можуть вам завадити вам користуватися цією версією. Якщо система повідомляє вам про помилку `AttributeError: „module“ object has no attribute „_trace_level“`, зниження версії `python-ldap` до 3.0.0 може вирішити проблему.

Після встановлення пакунка, ви можете вставити його до скрипту розпізнавання Django:

```

# Add LDAP backed, keep Django one if you want to be able to sign in
# even without LDAP for admin account
AUTHENTICATION_BACKENDS = (
    "django_auth_ldap.backend.LDAPBackend",
    "weblate.accounts.auth.WeblateUserBackend",

```

(continues on next page)

(continued from previous page)

```

)

# LDAP server address
AUTH_LDAP_SERVER_URI = "ldaps://ldap.example.net"

# DN to use for authentication
AUTH_LDAP_USER_DN_TEMPLATE = "cn=%(user)s,o=Example"
# Depending on your LDAP server, you might use a different DN
# like:
# AUTH_LDAP_USER_DN_TEMPLATE = 'ou=users,dc=example,dc=com'

# List of attributes to import from LDAP upon sign in
# Weblate stores full name of the user in the full_name attribute
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "name",
    # Use the following if your LDAP server does not have full name
    # Weblate will merge them later
    # 'first_name': 'givenName',
    # 'last_name': 'sn',
    # Email is required for Weblate (used in VCS commits)
    "email": "mail",
}

# Hide the registration form
REGISTRATION_OPEN = False

```

Примітка: Вам слід вилучити `'social_core.backends.email.EmailAuth'` з параметра `django:AUTHENTICATION_BACKENDS`, інакше користувачі зможуть встановлювати пароль у Weblate і проходити розпізнавання за допомогою пароля. Збереження `'weblate.accounts.auth.WeblateUserBackend'` все одно потрібен для того, щоб встановити права доступу і полегшити роботу з анонімними користувачами. Цей параметр також надає вам змогу входити до системи за допомогою локального адміністративного облікового запису, якщо ви його створили (наприклад, за допомогою `createadmin`).

Використання пароля bind

Якщо ви не можете скористатися безпосередньою прив'язкою для розпізнавання, вам доведеться скористатися пошуком і вказати користувача для прив'язки пошуку. Приклад:

```

import ldap
from django_auth_ldap.config import LDAPSearch

AUTH_LDAP_BIND_DN = ""
AUTH_LDAP_BIND_PASSWORD = ""
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "ou=users,dc=example,dc=com", ldap.SCOPE_SUBTREE, "(uid=%(user)s)"
)

```


Інтеграція із Active Directory

```
import ldap
from django_auth_ldap.config import LDAPSearch, NestedActiveDirectoryGroupType

AUTH_LDAP_BIND_DN = "CN=ldap,CN=Users,DC=example,DC=com"
AUTH_LDAP_BIND_PASSWORD = "password"

# User and group search objects and types
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "CN=Users,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s)"
)

# Make selected group a superuser in Weblate
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    # is_superuser means user has all permissions
    "is_superuser": "CN=weblate_AdminUsers,OU=Groups,DC=example,DC=com",
}

# Map groups from AD to Weblate
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "OU=Groups,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(objectClass=group)"
)
AUTH_LDAP_GROUP_TYPE = NestedActiveDirectoryGroupType()
AUTH_LDAP_FIND_GROUP_PERMS = True

# Optionally enable group mirroring from LDAP to Weblate
# AUTH_LDAP_MIRROR_GROUPS = True
```

Дивись також:

Django Authentication Using LDAP, Authentication

2.5.7 Розпізнавання за допомогою CAS

Щоб скористатися розпізнаванням за допомогою CAS, встановіть пакунок подібний до *django-cas-ng*.

Першим кроком є розкриття поля адреси електронної пошти користувача за допомогою CAS. Цю можливість має бути налаштовано на самому сервері CAS, вона потребує роботи із принаймні версією 2 CAS, оскільки у версії 1 CAS не передбачено підтримки атрибутів взагалі.

Другим кроком є оновлення Weblate для використання вашого сервера CAS і атрибутів.

Щоб встановити *django-cas-ng*, виконайте такі дії:

```
pip install django-cas-ng
```

Після встановлення пакунка, ви можете вставити його до скрипту розпізнавання Django внесенням змін до файла `settings.py`:

```
# Add CAS backed, keep the Django one if you want to be able to sign in
# even without LDAP for the admin account
AUTHENTICATION_BACKENDS = (
    "django_cas_ng.backends.CASBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# CAS server address
CAS_SERVER_URL = "https://cas.example.net/cas/"

# Add django_cas_ng somewhere in the list of INSTALLED_APPS
INSTALLED_APPS = (... , "django_cas_ng")
```

Нарешті, для прив'язки поля адреси електронної пошти до об'єкта користувача можна скористатися сигналом. Щоб це спрацювало, вам слід імпортувати сигнал з пакунка *django-cas-ng* і з'єднати ваш код із цим сигналом. Виконання цього завдання у файлів параметрів може спричинити проблеми, тому пропонуємо вам такий варіант:

- У методі `django.apps.AppConfig.ready()` налаштувань вашого застосунку
- У файлі проекту `urls.py` (якщо моделей не існує)

```
from django_cas_ng.signals import cas_user_authenticated
from django.dispatch import receiver

@receiver(cas_user_authenticated)
def update_user_email_address(sender, user=None, attributes=None, **kwargs):
    # If your CAS server does not always include the email attribute
    # you can wrap the next two lines of code in a try/catch block.
    user.email = attributes["email"]
    user.save()
```

Дивись також:

Django CAS NG

2.5.8 Налаштовування стороннього розпізнавання за допомогою Django

Загалом, із Weblate має працювати будь-який додаток розпізнавання Django. Просто виконуйте настанови для цього додатка, просто не забудьте встановити модуль користувачів Weblate.

Дивись також:

Розпізнавання за допомогою LDAP, Розпізнавання за допомогою CAS

Типово, встановлення полягає у додаванні модуля розпізнавання до запису `AUTHENTICATION_BACKENDS` і встановленні застосунку для розпізнавання (якщо таку передбачено) до `INSTALLED_APPS`:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

2.6 Керування доступом

Weblate постачається із системою прав доступу із багатьма рівнями, які надають змогу визначати права доступу користувача для усього екземпляра системи або для певного її складника.

Змінено в версії 3.0: До версії Weblate 3.0 система прав доступу базувалася на Django, але тепер її побудовано окремо для Weblate. Якщо ви користуєтеся застарілою версією зверніться до документації для вашої версії.

2.6.1 Просте керування доступом

Якщо ви не адмініструєте увесь встановлений екземпляр Weblate і просто маєте доступ до керування певними проектами (як на [Hosted Weblate](#)), ваші варіанти керування доступом обмежено вказаними нижче параметрами. Якщо вам не потрібна складна конфігурація, цих варіантів має бути достатньо.

Керування доступом на рівні проєкту

Примітка: Ця функція недоступна для проєктів, які використовують план Libre на Hosted Weblate.

Ви можете обмежити доступ користувачів до окремих проєктів, вибравши інший варіант *Керування доступом*. Доступні варіанти:

Відкритий

Доступний до перегляду усіма, придатний до перекладу користувачами, які увійшли до системи.

Захищений

Доступний до перегляду усіма, але придатний до перекладу лише вибраними користувачами.

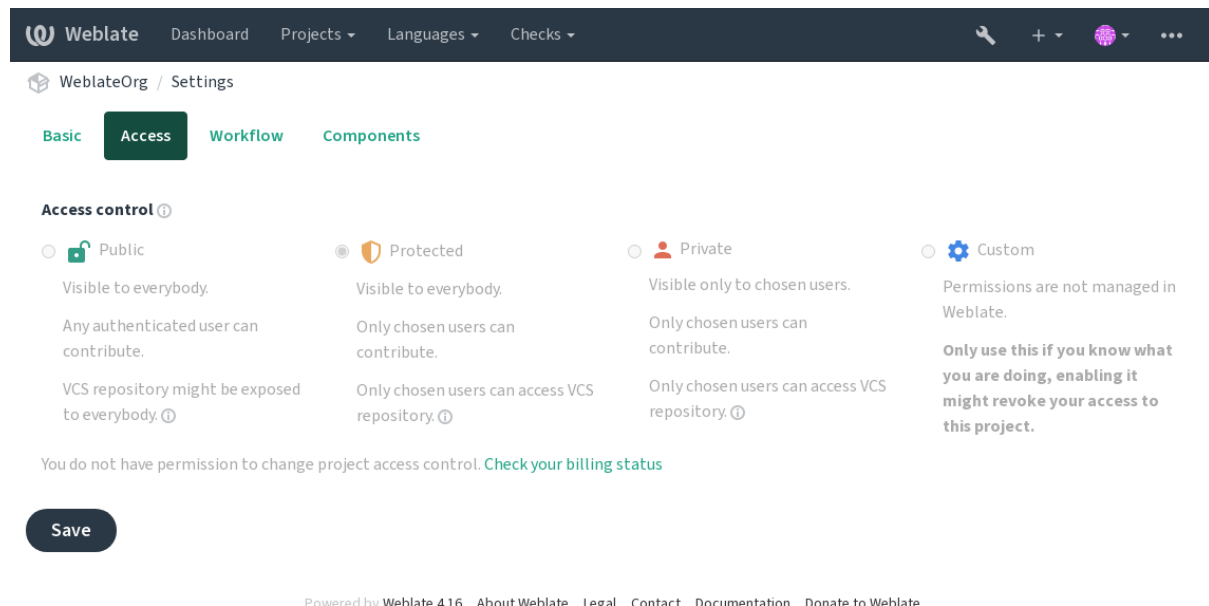
Приватний

Доступний до перегляду і перекладу лише вибраним користувачам.

Власний

Можливості *керування записами користувачів* буде вимкнено. Типово, усім користувачам буде заборонено виконувати будь-які дії над проєктом. Вам доведеться налаштувати усі права доступу за допомогою *нетипових правил доступу*.

Керування доступом може бути змінено на вкладці *Доступ* налаштувань (*Управління* ↓ *Налаштування*) відповідного проєкту.



Типове значення можна змінити за допомогою `DEFAULT_ACCESS_CONTROL`.

Примітка: Навіть для *приватних* проєктів буде розкрито певні дані про проєкт: статистику і підсумок про мови для всього екземпляра міститимуть дані для всіх проєктів, всупереч варіанту керування доступом. За цим можна визначити назву вашого проєкту та інші відомості.

Примітка: Набір прав доступу, який типово доступний для користувачів у проєктах *Public*, *Protected* і *Private*, може бути перевизначено адміністратором екземпляра Weblate за допомогою *нетипових параметрів*.

Дивись також:

Керування доступом

Керування правами доступу на рівні проєкту

Користувачі з правами доступу *Керування доступом до проєкту* (див. *Список привілеїв і вбудовані ролі*) можуть керувати записами користувачів у проєктах додаванням цих записів до команд. Початковий набір команд надається самим Weblate, а додаткові можна визначити коригуванням керування доступом. Ви можете обмежити команди певними мовами і призначити їм створені вами ролі доступу (див. *Список привілеїв і вбудовані ролі*).

Вказані нижче команди буде автоматично створено в усіх проєктах:

Для проєктів *Public*, *Protected* і *Private*:

Адміністрування

Включає усі можливі права доступу до проєкту.

Рецензування (лише якщо увімкнено *процес із рецензуванням*)

Може затверджувати переклади у процесі рецензування.

Лише для проєктів *Protected* і *Private*:

Переклад

Може перекладати проєкт і вивантажувати виконані поза межами вебредактора переклади.

Джерела

Можуть редагувати початкові рядки (якщо дозволено *параметрами проєкту*) і дані щодо початкового рядка.

Мови

Може керувати мовами перекладу (додавати або вилучати переклади).

Глосарій

Може змінювати глосарій (додавати або вилучати записи, а також вивантажувати дані).

Пам'ять

Може керувати пам'яттю перекладів.

Знімки екрана

Може керувати знімками вікон (додавати і вилучати знімки, а також пов'язувати зніми із початковими рядками).

Автоматичний переклад


Може використовувати автоматичний переклад.


VCS

Може управляти системою керування версіями та має доступ до експортованого репозиторію початкового коду.

Оплата

Має доступ до даних щодо фінансових рахунків і параметрів (див. *Оплата*).


Dashboard
Projects ▾
Languages ▾
Checks ▾


WeblateOrg / Access control

Users
Teams
API access

Users

Username	Full name	E-mail	Last sign in	Teams
testuser	Weblate Test	weblate@example.org	16 seconds ago	<div>Translate</div> <div></div>

Once all its permissions are removed, the user will be removed from the project.

Add a user

User to add

Please type in an existing Weblate account name or e-mail address.

Add

Block user

User to block

Please type in an existing Weblate account name or e-mail address.

Block duration

Block the user until I unblock

Block

Invite new user

E-mail

Username

Username may only contain letters, numbers or the following characters:
@ . + - _

Full name

Invite

Powered by Weblate 4.16
 [About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

Доступ до цих можливостей можна отримати за допомогою сторінки *Керування доступом*, доступ до якої можна отримати з меню проекту: *Управління* ↓ *Користувачі*.

Адміністратори команд

Нове в версії 4.15.

У кожній команді може бути адміністратор, який додає і вилучає користувачів у команді. Це корисно, якщо ви хочете побудувати самоврядні команди.

Запрошення нових користувачів

Крім того, окрім додавання наявного користувача до проєкту, можна запросити нових користувачів. Запис нового користувача буде створено негайно, але він лишатиметься неактивним до входу користувача до системи за допомогою посилання у запрошенні, яке буде надіслано електронною поштою. Для надсилання запрошення не потрібні права доступу на рівні сайту, достатньо буде прав доступу до керування доступом на рівні проєкту (наприклад, членства у команді *Administration*).

Підказка: Якщо запрошеним користувачем пропущено строк чинності запрошення, він або вона може встановити пароль, вказавши адресу електронної пошти запрошення у формі скидання пароля, оскільки обліковий запис для відповідного користувача вже створено.

Нове в версії 3.11: Передбачено можливість повторного надсилання запрошення користувачам електронною поштою (скасовує будь-які раніше надіслані запрошення).

Подібний тип запрошень доступний і на рівні сайту з *інтерфейсу керування* на вкладці *Користувачі*.

Блокування користувачів

Нове в версії 4.7.

Якщо якісь користувачі поведуться недостойно у вашому проєкті, у вас є можливість заблокувати їхню участь. Заблоковані користувачі все одно зможуть бачити проєкт, якщо у них є для цього дозволи, але вони не зможуть робити свій внесок.

Керування правами доступу на рівні проєктів

Ви можете налаштувати ваші проєкти як *Protected* або *Private* і керувати користувачами <manage-acl> окремих проєктів за допомогою інтерфейсу Weblate.

Типово, це запобігає наданню доступу *типовим командам* *Users* and *Viewers* через власні налаштування цих команд. Це не заборонить вам надавати права доступу до цих проєктів на рівня сайту шляхом зміни параметрів типових групи, створення нової команди або створення додаткових нетипових параметрів для окремого складника, як це описано у розділі *Нетипове керування доступом*.

Однією з головних переваг керування правами доступу з інтерфейсу користувача Weblate є те, що ви можете делегувати його іншим користувачам без надання їм прав доступу надкористувача. Для цього додайте користувачів до команди *Administration* проєкту.

2.6.2 Нетипове керування доступом

Примітка: Ця функція недоступна для проєктів, які використовують план Libre на Hosted Weblate.

Систему прав доступу засновано на командах і ролях, де ролі визначають набір прав доступу, а команди пов'язують їх із користувачами та перекладами, див. *Користувачі, ролі, команди і права доступу*, щоб дізнатися більше.

Найпотужнішими можливостями поточної версії системи керування доступом Weblate можна скористатися лише за допомогою *адміністративного інтерфейсу Django*. Цей інтерфейс надає змогу керувати правами доступу до будь-якого проєкту. Щоб скористатися ним, не обов'язково перемикатися на *керування доступом Custom*. Втім, щоб скористатися цим вам знадобляться права доступу надкористувача.

Якщо вам не цікаві подробиці реалізації і ви просто хочете створити достатньо прості налаштування на основі типових або не маєте доступу на рівні сайту до усього встановленого екземпляра Weblate (зокрема на *Hosted Weblate*), будь ласка, зверніться до розділу *Просте керування доступом*.

Типові налаштування

У цьому розділі міститься огляд деяких типових конфігурацій, які можуть бути для вас цікавим.

Керування правами доступу на рівні сайта

Для керування правами доступами для усього екземпляра одразу додайте користувачів до відповідних *типових команд*:

- *Users* (це виконується типово *автоматичною прив'язкою до команд*).
- *Reviewers* (якщо ви використовуєте *процес із рецензуванням* із спеціалізованими рецензентами).
- *Managers* (якщо ви хочете делегувати більшість дій з керування комусь іншому).

Вам слід зберігати налаштування усіх проєктів як *Public* (див. *Керування доступом на рівні проєкту*). Якщо цього не зробити, загальносайтові права доступу, які надаються членством у командах *Users* і *Reviewers*, ні на що не вплинуть.

Крім того, ви можете надати бажані додаткові права доступу типовим командам. Наприклад, ви можете надати права доступу для керування знімками вікон усім учасникам групи *Users*.

Ви також маєте визначити деякі нові нетипові команди. Якщо ви хочете зберегти керування вашими загальносайтовими правами доступу для цих команд, виберіть відповідне значення для параметра *Вибір проєкту* (наприклад, *Усі проєкти* або *Усі публічні проєкти*).

Нетипові права доступу до мов, складників або проєктів

Ви можете створити ваші власні спеціалізовані команди для керування правами доступу для певних об'єктів, зокрема мов, складників та проєктів. Хоча ці команди можуть додавати лише додаткові права доступу — ви не можете відкликати будь-які права доступу за загальносайтовими командами або командами проєкту додаванням ще однієї нетипової команди.

Приклад:

Якщо ви хочете (з будь-яких причин) дозволити переклад певною мовою (наприклад *чеською*) лише обмеженому набору надійних перекладачів, а переклади іншими мовами зробити загальнодоступними, вам слід зробити так:

1. Вилучити права доступу до перекладу *Czech* для всіх користувачів. За типових налаштувань це можна зробити зміною *типової команди Users*.

Table 1: Група *Users*

Вибір мови	<i>As defined</i>
Мови	Усі, окрім <i>Czech</i>

2. Додати спеціалізовану команду для перекладачів *Czech*.

Table 2: Група *Czech translators*

Ролі	<i>Power users</i>
Вибір проєкту	<i>All public projects</i>
Вибір мови	<i>As defined</i>
Мови	<i>Czech</i>

3. Додати користувачів, яким ви хочете надати права доступу, до цієї команди.

Як ви можете бачити, керування правами доступу у цей спосіб є потужним, але користуватися ним є марудною справою. Ви не можете делегувати його іншому користувачу без надання прав доступу надкористувача.

Користувачі, ролі, команди і права доступу

Моделі розпізнавання складаються з декількох об'єктів:

Права доступу

Персональні права доступу, які визначаються Weblate. Ви не можете визначати права доступу для користувачів — це завдання може бути виконане лише за допомогою призначення ролей.

Роль

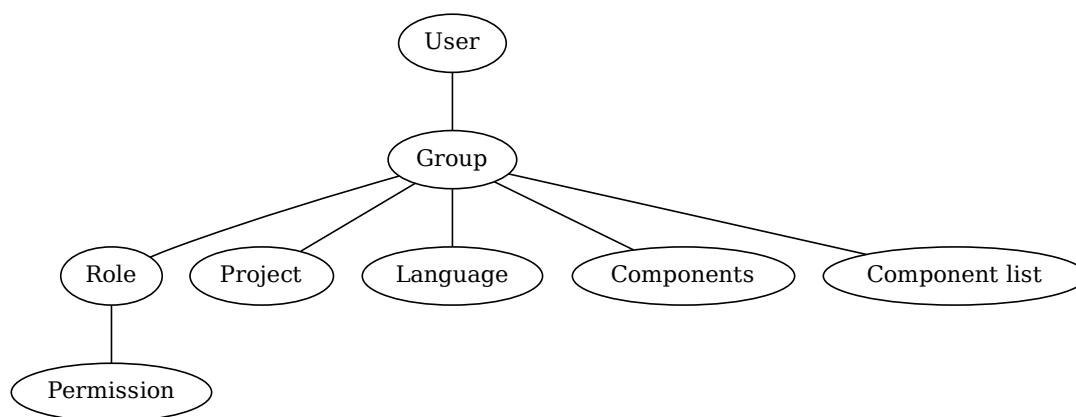
Роль визначає набір прав доступу. За допомогою ролі ви можете використовувати однакові набори прав доступу у декількох місцях. Ролі також спрощують адміністрування.

Користувач

Користувачі можуть бути учасниками декількох команд одночасно.

Група

Групи поєднують об'єкти ролей, користувачів та розпізнавання (проекти, мови і списки складників).



Примітка: Із командою може бути не пов'язано жодних ролей. У цьому випадку програма припускатиме доступ до навігації проектом для будь-кого (див. нижче).

Доступ до навігації проектом

Користувач має бути учасником команди, пов'язаної із проектом або будь-яким складником у проекті. Участь достатньо — для навігації проектом не потрібні якісь особливі права доступу (це використано у типовій команді *Переглядачі*, див. [Список команд](#)).

Доступ до навігації складником

Користувач отримує доступ до необмеженого складника, щойно він отримує доступ до проекту складника (і матимує усі права доступу, які надаються користувачів у проекті). Із увімкненим параметром *Обмежений доступ* доступ до складника потребує явного доступу до складника (або списку складників, який його містить).

Область команд

Область дії прав доступу, які призначаються ролями у командах, застосовуються такими правилами:

- Якщо група визначає якийсь *Список складників*, усі права доступу, надані учасникам цієї групи, надаються для всіх складників у списках складників, пов'язаних із командою, і доступ без додаткових прав доступу надається для всіх проєктів, частиною яких є ці складники. *Складники* і *Проєкти* буде проігноровано.
- Якщо команда визначає якийсь *Складник*, усі права доступу, надані учасникам цієї команди, надаються для всіх складників, пов'язаних із командою, і доступ без додаткових прав доступу надається для всіх проєктів, частиною яких є ці складники. *Проєкти* буде проігноровано.
- Інакше, якщо команда визначає якісь *Проєкти*, або безпосереднім визначенням їхнього списку, або встановленням для параметра *Вибір проєктів* значення, подібного до *Усі публічні проєкти*, усі ці права доступу застосовуються до усіх проєктів, які насправді надають ті самі права доступу для доступу до усіх *складників без обмежень* проєкту.
- Обмеження, які накладаються параметром *Мови* команди, застосовуються окремо, після перевірки, чи має користувач доступ до певних дій. А саме, вони застосовуються лише до дій безпосередньо пов'язаних із самим процесом перекладу, зокрема рецензуванням, збереженням перекладів, додаванням пропозицій тощо.

Підказка: Для автоматизації включення всіх мов або проєктів ви можете скористатися пунктами *Вибір мови* та *Вибір проєкту*.

Приклад:

Припустімо, маємо проєкт щось із такими складниками: щось/десь і щось/там і таку команду:

Table 3: Група *Spanish Admin-Reviewers*

Ролі	Рецензування рядків, Управління репозиторієм
Складники	щось/десь
Мови	<i>Spanish</i>

Учасники цієї команди матимуть такі права доступу (припускаємо параметри типової ролі):

- Загальний доступ (до навігації) до усього проєкту щось, включно із обома складниками у ньому: щось/десь і щось/там.
- Рецензування рядків у іспанському перекладі щось/десь (і тільки у ньому).
- Управління системою керування версіями для усього репозиторію щось/десь, наприклад внесення зміни з черги, які є результатами роботи перекладачів, для всіх мов.

Автоматичні призначення команд

У нижній частині сторінки редагування *Група адміністративного інтерфейсу Django* ви можете вказати *Автоматичні призначення команд*, тобто список формальних виразів, які використовуватимуться для автоматичного пов'язування створених записів користувачів до команди на основі адреси електронної пошти. Ця прив'язка виконуватиметься лише під час створення облікового запису.

Найпоширеніший випадок використання цієї можливості — пов'язування всіх нових користувачів до певної типової команди. Для цього вам, імовірно, слід зберегти типове значення (^.*\$) у полі регулярного виразу. Ще одним випадком використання цього параметра може бути типове надання певних додаткових прав доступу працівникам вашої компанії. Припускаючи, що всі вони користуються корпоративними адресами електронної пошти на вашому домені, досягти цього можна виразом, подібним до ^.*@mycompany.com.

Примітка: Автоматична прив'язка до команд для команд *Users* і *Viewers* виконуватиметься завжди при оновленні версії Weblate. Якщо ви хочете вимкнути її, встановіть для формального виразу прив'язки значення `^$`, якому не відповідає жоден рядок.

Примітка: У поточній версії немає способу пакетного додавання наявних користувачів до певної команди за допомогою інтерфейсу користувача. Для цього ви можете вдатися до використання *програмного інтерфейсу REST*.

Типові команди і ролі

Після встановлення буде створено типовий набір команд (див. [Список команд](#)).

Ці ролі та команди створюються під час встановлення. Актуальність вбудованих ролей завжди підтримується перенесенням бази даних під час оновлення. Ви не зможете внести зміни. Якщо ви хочете визначити власний набір прав доступу, визначте для нього нову роль.

Список привілеїв і вбудовані ролі

Область	Дозвіл	Ролі
Фінансові рахунки (див. Оплата)	Переглянути відомості про оплату	<i>Administration, Billing</i>
Зміни	Завантажити зміни	<i>Administration</i>
Коментарі	Опублікувати коментар	<i>Administration, Edit source, Power user</i>
	Вилучити коментар	<i>Administration</i>
	Вирішити коментар	<i>Administration, Review strings</i>
Складник	Змінити налаштування складників	<i>Administration</i>
	Блокувати складник, запобігаючи перекладу	<i>Administration</i>
Глосарій	Додати запис до глосарія	<i>Administration, Manage glossary, Power user</i>
	Змінити словникове гніздо	<i>Administration, Manage glossary, Power user</i>
	Видалити словникове гніздо	<i>Administration, Manage glossary, Power user</i>
	Вивантажити словникове гніздо	<i>Administration, Manage glossary, Power user</i>
Автоматичні пропозиції	Використовувати автоматичні пропозиції	<i>Administration, Edit source, Power user</i>
Пам'ять перекладів	Змінити пам'ять перекладів	<i>Administration, Manage translation memory</i>
	Вилучити пам'ять перекладів	<i>Administration, Manage translation memory</i>
Проекти	Змінити параметри проекту	<i>Administration</i>
	Керувати доступом до проекту	<i>Administration</i>
Звіти	Завантажити звіти	<i>Administration</i>
Знімки екрана	Додати знімок	<i>Administration, Manage screenshots</i>
	Змінити знімок	<i>Administration, Manage screenshots</i>
	Видалити знімок	<i>Administration, Manage screenshots</i>
Джерельні рядки	Редагувати додаткові відомості про рядок	<i>Administration, Edit source</i>
Рядки	Додати новий рядок	<i>Administration</i>
	Вилучити рядок	<i>Administration</i>
	Відхилити невдалу перевірку	<i>Administration, Edit source, Power user</i>
	Змінити рядки	<i>Administration, Edit source, Power user</i>
	Перерити рядки	<i>Administration, Review strings</i>
	Змінювати рядок з примусовими пропозиціями	<i>Administration, Review strings</i>
	Змінити джерельні рядки	<i>Administration, Edit source, Power user</i>
Пропозиції	Прийняти пропозицію	<i>Administration, Edit source, Power user</i>
	Додати пропозицію	<i>Administration, Edit source, Add suggestion</i>
	Вилучити пропозицію	<i>Administration, Power user</i>
	Проголосувати за пропозицію	<i>Administration, Edit source, Power user</i>

Table 4 – continued from previous page

Область	Дозвіл	Ролі
Переклади	Додати мову перекладу	<i>Administration, Power user, Manager</i>
	Виконати автоматичний переклад	<i>Administration, Automatic translation</i>
	Видалити наявний переклад	<i>Administration, Manage languages</i>
	Завантажити файл перекладу	<i>Administration, Edit source, Access repository</i>
	Додати кілька мов для перекладу	<i>Administration, Manage languages</i>
Вивантаження	Задати творця завантаженого перекладу	<i>Administration</i>
	Переписати наявні переклади завантаженням	<i>Administration, Edit source, Power user</i>
	Вивантажити переклади	<i>Administration, Edit source, Power user</i>
VCS	Доступ до внутрішнього репозиторію	<i>Administration, Access repository, Power user</i>
	Внести зміни у внутрішній репозиторій	<i>Administration, Manage repository</i>
	Записати зміни з внутрішнього репозиторію	<i>Administration, Manage repository</i>
	Скинути усі зміни у внутрішньому репозиторії	<i>Administration, Manage repository</i>
	Переглянути розташування основного репозиторію	<i>Administration, Access repository, Power user</i>
	Оновити внутрішній репозиторій	<i>Administration, Manage repository</i>
Привілеї на рівні сайту	Використовувати інтерфейс керування	
	Додавати нові проекти	
	Додати визначення мов	
	Керувати визначеннями мов	
	Керування командами	
	Управління користувачами	
	Керувати ролями	
	Керувати оголошеннями	
	Керувати пам'яттю перекладів	
	Керування автоматизацією	
	Керувати списками складників	

Примітка: Привілеї на рівні сайту не надаються будь-якій типовій ролі. Ці привілеї є доволі широкими і дуже близькими до стану надкористувача — більшість з них стосуються усіх проектів у встановленому вами Weblate.

Список команд

Під час встановлення (або після виконання *setupgroups*) буде створено вказані нижче команди. Ви можете внести зміни до цих груп. Втім, під час перенесення даних типові групи, якщо ви вилучите або перейменуєте їх, буде створено повторно.

Guests

Визначає права доступу для користувачів, які не пройшли розпізнавання.

У цій команді містяться записи лише анонімних користувачів (див. *ANONYMOUS_USER_NAME*).

Ви можете вилучати ролі з цієї команди для обмеження прав доступу користувачів, які не пройшли розпізнавання.

Типові ролі: *Add suggestion, Access repository*

Переглядачі

Ця роль забезпечує видимість відкритих проектів для всіх користувачів. Типово учасниками цієї команди є всі користувачі.

Типово, *автоматична прив'язка команд* створює усі нові облікові записи учасників цієї команди, коли вони долучаються.

Типові ролі: немає

Users

Типова команда для всіх користувачів.

Типово, *автоматична прив'язка команд* створює усі нові облікові записи учасників цієї команди, коли вони долучаються.

Типові ролі: *Power user*

Рецензенти

Група для рецензентів (див. *Процеси перекладу*).

Типові ролі: *Review strings*

Керівники

Група для адміністраторів.

Типові ролі: *Administration*

Попередження: Ніколи не вилучайте попередньо визначені команди і користувачів Weblate, оскільки це може призвести до неочікуваних проблем. Якщо ви не хочете скористатися цими можливостями, просто вилучіть усі привілеї з цих записів.

2.6.3 Додаткові обмеження доступу

Якщо ви хочете скористатися встановленим вами Weblate у менш відкритий спосіб, тобто дозволити нових користувачів лише на основі запрошень, зробити це можна налаштуванням Weblate так, що до нього матимуть доступ лише відомі користувачі. Для цього ви можете встановити для параметра *REGISTRATION_OPEN* значення *False* для запобігання реєстрації будь-яких нових користувачів і встановити для *REQUIRE_LOGIN* значення */ . **, щоб вимагати входу до системи для доступу до усіх сторінок сайту. Це, на базовому рівні, є способом блокування встановленого вами Weblate.

Підказка: Ви можете скористатися вбудованими *запрошеннями* для додавання нових користувачів.

2.7 Проекти перекладу

2.7.1 Упорядкування перекладів

Weblate упорядковує придатні до перекладу дані системи керування версіями проекту і складників у деревоподібну структуру.

- Об'єктом нижнього рівня є *проект*. Проект має містити усі переклади, які йому належать, разом (наприклад, переклади декількох версій застосунку та/або відповідної документації).
- Рівнем вище розташовано *складник*, який є, власне, складником, який перекладають. Ви визначаєте репозиторій системи керування версіями, яке слід використовувати, та маску файлів, які слід перекладати.
- Над *складником* розташовано окремі переклади, які керуються Weblate у автоматичному режимі як файли перекладу (які відповідають *масці файлів складника*, заданій у *складнику*) у репозиторії системи керування версіями.

У Weblate передбачено підтримку широкого діапазону форматів перекладів (двомовних і одномовних). Підтримку реалізовано на основі Translate Toolkit, див. *Підтримувані формати файлів*.

Примітка: Ви можете використовувати спільне клоноване із системи керування версіям репозиторій для декількох складників за допомогою можливості *Внутрішні адреси Weblate*. Наполегливо рекомендуємо скористатися цією можливістю, якщо у вашому проекті декілька складників використовують те саме репозиторій

системи керування версіями. Таким чином ви пришвидшите роботу системи і зменшите витрату місця на диску.

2.7.2 Додавання проєктів і складників перекладу

Змінено в версії 3.2: Включено інтерфейс для додавання проєктів і складників — у вас більше немає потреби у використанні *адміністративного інтерфейсу*.

Змінено в версії 3.4: У нових версіях процедуру додавання складників поділено на декілька кроків і реалізовано автоматичне визначення значень більшості параметрів.

На основі даних щодо ваших прав доступу, можна створювати нові проєкти та складники. Ці дії завжди доступні для користувачів із правами доступу *Додавання нових проєктів* і, якщо у вашому екземплярі використовується облік (наприклад, він використовується на <https://hosted.weblate.org/>, див. *Онлама*), ви також можете створювати рахунки на основі ваших тарифних планів з облікового запису користувача, який керує рахунками.

Дані поточного тарифного плану можна переглянути на окремій сторінці:

Billing plan

Current plan	Basic plan (Active)
Monthly price	19 EUR
Yearly price	199 EUR
Strings limit	Used 0
Languages limit	Used 0
Last invoice	2023-02-28 - 2023-03-02
Projects limit	Used 0 of 1
Projects	No projects currently assigned!

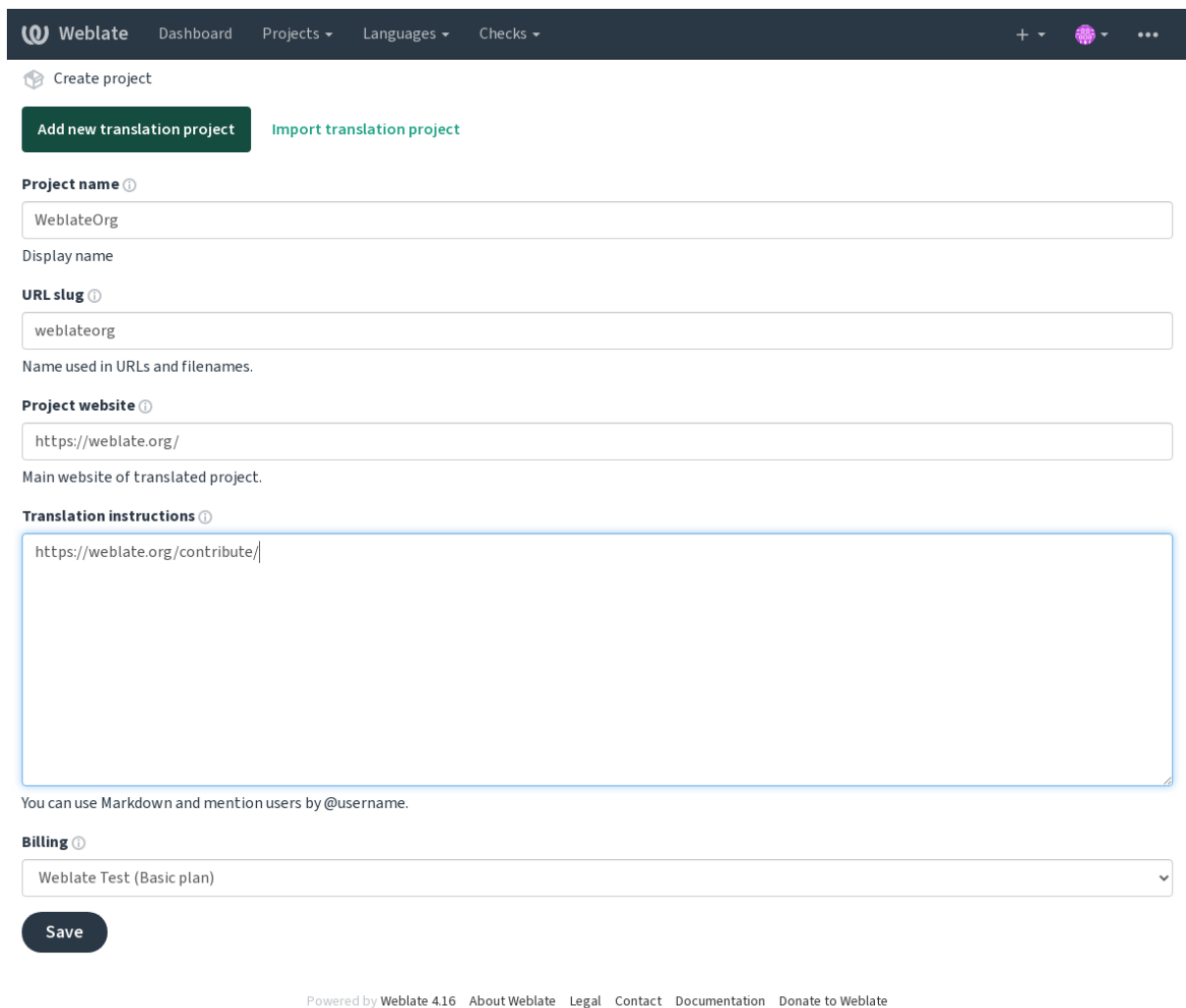
Invoices

Invoice period	Invoice amount	Download invoice
02/28/2023 - 03/02/2023	19.0 EUR	Not available

Buttons: Add new translation project, Terminate billing plan

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

З цієї сторінки або меню на панелі навігації ви можете ініціювати створення проєкту, заповнивши базові відомості щодо проєкту перекладу для повного його додавання:



The screenshot shows the 'Create project' form in the Weblate interface. The form includes several input fields: 'Project name' (containing 'WeblateOrg'), 'URL slug' (containing 'weblateorg'), and 'Project website' (containing 'https://weblate.org/'). There is also a large text area for 'Translation instructions' containing 'https://weblate.org/contribute/'. Below these fields is a 'Billing' dropdown menu set to 'Weblate Test (Basic plan)'. A 'Save' button is at the bottom of the form. The footer of the page includes links for 'Powered by Weblate 4.16', 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Create project

Add new translation project Import translation project

Project name ⓘ
WeblateOrg
Display name

URL slug ⓘ
weblateorg
Name used in URLs and filenames.

Project website ⓘ
https://weblate.org/
Main website of translated project.

Translation instructions ⓘ
https://weblate.org/contribute/

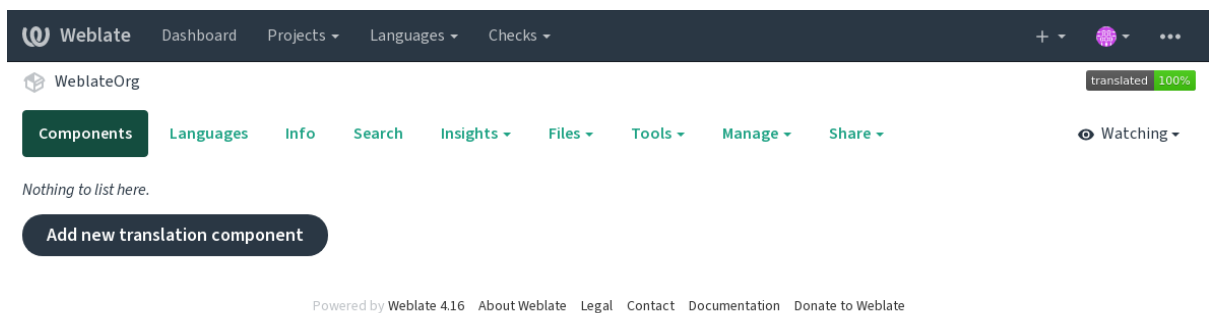
You can use Markdown and mention users by @username.

Billing ⓘ
Weblate Test (Basic plan)

Save

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Після створення проєкту ви перейдете безпосередньо до сторінки проєкту:



Створити складник перекладу можна за допомогою одинарного клацання на цьому пункті. Процедура створення складника є багатокроковою, більшість параметрів перекладу буде визначено автоматично. Передбачено декілька підходів до створення складника:

З контролю версій

Створює складник на основі віддаленого репозиторію системи керування версіями.

З наявного складника

Створює додатковий складник до наявного шляхом вибору інших файлів.

Додаткове відділення

Створює додатковий складник до наявного, але для іншої гілки у репозиторії.

Завантажити файли перекладів

Вивантажити файли перекладу до Weblate, якщо у вас немає системи керування версіями або ви не хочете інтегрувати її з Weblate. Пізніше ви можете оновити дані за допомогою вебінтерфейсу або *програмного інтерфейсу*.

Перекласти документ

Вивантажити один документ або файл для перекладу і перекласти його.

Почати спочатку

Створити порожній проєкт перекладу і додати рядки вручну.

Якщо у вас вже є наявні складники перекладу, ви також можете без проблем додати нові складники для додаткових файлів або гілки з використанням того самого репозиторію.

Спочатку вам слід заповнити назву і розташування репозиторію:

Create component

From version control Upload translations files Translate document Start from scratch

Create a new translation component from remote version control system repository.

Component name ⓘ

Language names

Display name

URL slug ⓘ

language-names

Name used in URLs and filenames.

☐ Use as a glossary ⓘ

Project ⓘ

WeblateOrg

Source language ⓘ

English

Language used for source strings in all components

Version control system ⓘ

Git

Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.

Source code repository ⓘ

https://github.com/WeblateOrg/demo.git

URL of a repository, use weblate://project/component to share it with other component.


Repository branch ⓘ

Repository branch to translate

Continue

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

На наступній сторінці вам буде показано список виявлених придатних до перекладу ресурсів:

 Weblate [Dashboard](#) [Projects](#) [Languages](#) [Checks](#) [+](#) [🌐](#) [...](#)

Create component

Add new translation component ⓘ

Choose translation files to import ⓘ

☐ Specify configuration manually

☐ File format `Android String Resource`, File mask `app/src/main/res/values-*/strings.xml`

☐ File format `gettext PO file`, File mask `weblate/langdata/locale/*/LC_MESSAGES/django.po`

☐ File format `gettext PO file`, File mask `weblate/locale/*/LC_MESSAGES/django.po`

☐ File format `gettext PO file`, File mask `weblate/locale/*/LC_MESSAGES/djangojs.po`

Continue

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

На останньому кроці ви можете переглянути дані щодо складника перекладу і вкажіть додаткові подробиці:

Web

late

Dashboard

Projects

Languages

Checks

Create component

Detected license as MIT, please check whether it is correct.

Add new translation component

Project

Web

lateOrg

Component name

Language names

Display name

URL slug

language-names

Name used in URLs and filenames.

Version control system

Git

Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.

Source code repository

https://github.com/Web

lateOrg/demo.git

URL of a repository, use weblate://project/component to share it with other component.

Repository branch

Repository branch to translate

Repository push URL

URL of a push repository, pushing is turned off if empty.

Push branch

Branch for pushing changes, leave empty to use repository branch

Repository browser

https://github.com/Web

lateOrg/demo/blob/{{branch}}/{{filename}}#L{{line}}

Link to repository browser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders. You might want to strip leading directory by using {{filename|parentdir}}.

File format

gettext PO file

File mask

app/src/main/res/values-*/strings.xmlweblate/langdata/locale/*/LC_MESSAGES/django.po

Path of files to translate relative to repository root, use * instead of language code, for example: po/* .po or locale/*/LC_MESSAGES/django.po.

Monolingual base language file

app/src/main/res/values/strings.xml

Filename of translation base file, containing all strings and their source; it is recommended for monolingual translation formats.

☒ Edit base file

Whether users will be able to edit the base file for monolingual translations.

Intermediate language file

Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.

Adding new translation

Create new language file

How to handle requests for creating new translations.

Template for new translations

weblate/langdata/locale/django.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Translation license

GNU General Public License v3.0 or later

Language code style

Default based on the file format

Customize language code used to generate the filename for translations created by Weblate.

Language filter

^(cs|he|hu)\$

Regular expression used to filter translation files when scanning for file mask.

Source language

English

Language used for source strings in all components

☐ Use as a glossary

You will be able to edit more options in the component settings after creating it.

Save

Дивись також:

Адміністративний інтерфейс Django, Налаштування проєкту, Налаштовування складників

2.7.3 Налаштування проєкту

Створіть проєкт перекладу, а потім додайте новий складник для перекладу до нього. Проєкт подібний до шафи, у якій складено самі переклади. Усі складники в одному проєкті мають спільні пропозиції і словник. Крім того, переклади автоматично переносяться на усі складники в одному проєкті (якщо поширення перекладів не було вимкнено у налаштуваннях складника), див. *Пам'ять перекладів*.

Дивись також:

/devel/integration

Ці основні атрибути допомагають перекладачам проєкту та надають їм відомості:

Назва проєкту

Докладна назва проєкту, використовується для показу назви проєкту.

Дивись також:

PROJECT_NAME_RESTRICT_RE

Ідентифікатор адреси

Назва проєкту, яку можна включати до адрес.

Вебсайт проєкту

Адреса, за якою перекладачі можуть знайти додаткові відомості щодо проєкту.

Цей параметр є обов'язковим, якщо таку поведінку не вимкнено за допомогою *WEBSITE_REQUIRED*.

Дивись також:

PROJECT_WEB_RESTRICT_HOST,
PROJECT_WEB_RESTRICT_RE

PROJECT_WEB_RESTRICT_NUMERIC,

Вказівки щодо перекладу

Текст із описом процедури локалізації у проєкті та будь-які інші відомості, які можуть бути корисними для перекладачів. Для форматування тексту та вставлення посилань можна скористатися кодом Markdown .

Установлення заголовка «Language-Team»

Визначає, чи має Weblate керувати заголовком Language-Team (у поточній версії ця можливість є застосовною лише до файлів *GNU Gettext*).

Використовувати спільну пам'ять перекладів

Визначає, чи слід використовувати спільну пам'ять перекладів, щоб дізнатися більше, див. *Спільна пам'ять перекладів*.

Типове значення можна змінити за допомогою `DEFAULT_SHARED_TM`.

Зробіть внесок у спільну пам'ять перекладів

Визначає, чи слід вносити дані до спільної пам'яті перекладів, щоб дізнатися більше, див. *Спільна пам'ять перекладів*.

Типове значення можна змінити за допомогою `DEFAULT_SHARED_TM`.

Керування доступом

Налаштувати керування доступом на рівні проєктів. Щоб дізнатися більше, див. *Керування доступом на рівні проєкту*.

Типове значення можна змінити за допомогою `DEFAULT_ACCESS_CONTROL`.

Увімкнути рецензії

Увімкнути процес перекладу із рецензуванням, див. *Спеціалізовані рецензенти*.

Увімкнути рецензування початкових даних

Увімкнути процес рецензування для початкових рядків, див. *Рецензування початкових рядків*.

Дивись також:

report-source, *Коментарі*

Увімкнути обробники

Визначає, чи буде використано *скрипти слідування* для цього репозиторію.

Дивись також:

Проміжний мовний файл, Шлюз якості для початкових рядків, Двомовні і одномовні формати, Визначення мов

Альтернативні назви мови

Визначте прив'язку кодів мов при імпортуванні перекладів до Weblate. Скористайтесь цим, якщо коди мов у ваших репозиторіях є нестандартними, і ви хочете мати однорідний вигляд у Weblate, або якщо ви хочете скористатися нестандартним іменуванням ваших файлів перекладу.

Типовим випадком використання є прив'язка американської англійської до англійської: `en_US:en`

Декілька прив'язок слід відокремлювати комою: `en_GB:en,en_US:en`

За допомогою нестандартного коду: `ia_FOO:ia`

Підказка: Коди мов прив'язуються при встановленні відповідності файлів перекладу, прив'язки встановлюються без врахування регістру, тому переконайтеся, що ви використовуєте коди початкових мов у тій самій формі, яку використано у назвах файлів.

Дивись також:

Додавання нових перекладів, Код мови, Обробка кодів мов

2.7.4 Налаштовування складників

Складник групує дані для перекладу. Ви вводите адресу репозиторію системи керування версіями і маску файлів, для яких виконуватиметься переклад, і Weblate автоматично отримує дані з цієї системи керування версіями і знаходить усі відповідні придатні до перекладу файли.

Дивись також:

`/devel/integration`

Ви можете знайти деякі приклади типових налаштувань у *форматах*.

Примітка: Рекомендуємо зберігати притомний розмір складників перекладу — поділіть переклад за будь-яким критерієм, який має сенс у вашому випадку (окремі застосунки або додатки, розділи книги або сторінки сайту).

Weblate без проблем обробляє переклади із десятками тисяч рядків, але для таких файлів важче розподілити роботу і виконати координацію між перекладачами у таких великих складниках перекладу.

Якщо визначення мови для перекладу пропущено, буде створено порожнє визначення із назвою, подібною до «cs_CZ (generated)». Вам слід скоригувати визначення і повідомити про нього авторам Weblate, щоб пропущені мови можна було включити до наступного випуску.

Складник містить усі важливі параметри для роботи із системою керування версіями і для отримання перекладів з неї:

Назва складника

Повна назва складника, яка використовується для показу назви складника.

Ідентифікатор складника

Назва складника, яку можна включати до адрес.

Проект складника

Налаштування проекту, якому належить складник.

Система керування версіями

Система керування версіями, якою слід скористатися. Див. докладний опис у розділі *Налаштування інтеграції із керуванням версіями*.

Дивись також:

Записування змін з Weblate

Репозиторій з джерелами

Репозиторій системи керування версіями, яке буде використано для записування змін.

Дивись також:

Див. *Доступ до репозиторіїв*, щоб дізнатися більше про визначення адрес.

Підказка: Це може бути або реальна адреса системи керування версіями або `weblate://проект/складник`, що вказує на те, що репозиторій має бути спільним із іншим складником. Див. *Внутрішні адреси Weblate*, щоб дізнатися більше.

Адреса для записування до репозиторію

Адреса репозиторію, яка використовуватиметься для запису даних до репозиторію. Цей параметр використовується лише для *Git* і *Mercurial*, а підтримку запису вимкнено для цих систем керування версіями, якщо значення цього параметра є порожнім.

Для пов'язаних сховищ не використовується — буде застосовано параметри з пов'язаного складника.

Дивись також:

Див. *Доступ до репозиторіїв*, щоб дізнатися більше про те, як вказати адресу репозиторію, і *Записування змін з Weblate*, щоб дізнатися більше про те, як записувати зміни до репозиторію з Weblate.

Переглядач репозиторію

Адреса навігатора репозиторієм, який використовується для показу початкового коду файлів (розташування використаних повідомлень). Якщо значення порожнє, такі посилання створення не буде. Ви можете скористатися *Розмітка шаблонів*.

Наприклад, для GitHub скористайтеся чимось таким: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

Якщо ваші шляхи є відносними щодо іншої теки (шлях містить `..`), вам варто вилучити початковий каталог за допомогою фільтра `parentdir` (див. *Розмітка шаблонів*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

Експортований URL репозиторію

Адреса, куди експортуються зміни, внесені Weblate. Це важливо, якщо не використано *Безперервна локалізація* або є потреба в об'єднанні змін вручну. Ви можете скористатися *Засіб експортування Git* для автоматизації цього для репозиторію Git.

Гілка репозиторію

Яку гілку слід отримати з репозиторію керування версіями, і де слід шукати переклади.

Для пов'язаних сховищ не використовується — буде застосовано параметри з пов'язаного складника.

Гілка для запису

Гілка, до якої слід записувати зміни. Не заповнюйте, якщо слід використовувати *Гілка репозиторію*.

Для пов'язаних сховищ не використовується — буде застосовано параметри з пов'язаного складника.

Примітка: У поточній версії передбачено підтримку лише для Git, GitLab і GitHub. Буде проігноровано для всіх інших інтеграцій із системами керування версіями.

Дивись також:

Записування змін з Weblate

Маска файлів

Маска файлів для перекладу, включно зі шляхом. У ній має бути один символ «*», який замінюватиме код мови (див. *Визначення мов*, щоб дізнатися про принципи обробки). Якщо ваш репозиторій містить понад один файл перекладу (наприклад більше доменів gettext), вам слід створити складник для кожного з них.

Наприклад, `po/* .po` або `locale/*/LC_MESSAGES/django.po`.

Якщо у назві вашого файла містяться спеціальні символи, зокрема `[`, `]`, їх слід екранувати: `[]` або `[]`.

Дивись також:

Двомовні і одномовні формати, Що означає повідомлення «Знайдено кілька файлів для однієї мови (en)»?

Основний файл для одномовного перекладу

Базовий файл, у якому містяться визначення рядків для *Одномовні складники*.

Дивись також:

Двомовні і одномовні формати, Що означає повідомлення «Знайдено кілька файлів для однієї мови (en)»?

Змінити основний файл

Чи слід дозволяти редагування базового файла для *Одномовні складники*.

Проміжний мовний файл

Проміжний файл мови для *Одномовні складники*. Здебільшого, це файл перекладу, який надається розробниками і використовується для створення самих рядків для перекладу.

Якщо встановлено, початкові рядки засновано на цьому файлі, але усі інші переклади засновано на *Основний файл для одномовного перекладу*. Якщо рядок не перекладено мовою джерела, переклад іншими мовами заборонено. Цей параметр надає *Шлюз якості для початкових рядків*.

Дивись також:

Шлюз якості для початкових рядків, Двомовні і одномовні формати, Що означає повідомлення «Знайдено кілька файлів для однієї мови (en)»?

Шаблон для нових перекладів

Базовий файл, який буде використано для створення нових перекладів, наприклад файл `.pot` у перекладах на основі gettext.

Підказка: Для багатьох одномовних форматів Weblate типово починає переклад з порожнього файла. Скористайтеся цим варіантом, якщо ви хочете при створенні перекладу усі рядки було показано як порожні значення.

Дивись також:

[adding-translation](#), *Додавання нових перекладів*, *Додавання нового перекладу*, *Двомовні і одномовні формати*, *Що означає повідомлення «Знайдено кілька файлів для однієї мови (en)»?*

Формат файлу

Формат файла перекладу, див. також *Підтримувані формати файлів*.

Адреса для повідомлень про помилки у рядках

Адреса електронної пошти для звітів щодо вад у кодї. На цю адресу також надсилатимуться сповіщення щодо отримання будь-яких коментарів щодо початкового коду, які було створено на Weblate.

Дозволити поширення перекладу

Ви можете вимкнути поширення перекладів на цей складник з інших складників у тому самому проєкті. Рішення залежить від того, що саме ви перекладаєте — іноді бажаним є використання однакових перекладів усюди.

Зазвичай, варто вимкнути цю можливість для одномовних перекладів, якщо ви не використовуєте однакові ідентифікатори у всьому проєкті.

Типове значення можна змінити за допомогою `DEFAULT_TRANSLATION_PROPAGATION`.

Дивись також:

Підтримання синхронізації перекладів у різних складниках

Пропонувати переклад

Чи приймаються пропозиції щодо перекладів у цьому складнику.

Голосування за пропозицію

Вмикає голосування за пропозиції, див. *Голосування за пропозицію*.

Автоматичне прийняття пропозиції

Автоматично приймати пропозиції, які перемогли у голосуванні, див. [Голосування за пропозицію](#).

Прапорці перекладу

Налаштування перевірок якості та іншої поведінки Weblate, див. [Налаштовування поведінки за допомогою прапорців](#).

Примусові перевірки

Список перевірок, які не можуть бути знехтувані, див. [Примусові перевірки](#).

Примітка: Примусове задання перевірки не вмикає її автоматично. Вам доведеться ще увімкнути перевірку за допомогою [Налаштовування поведінки за допомогою прапорців](#) у [Прапорці перекладу](#) або [Додаткові відомості щодо початкових рядків](#).

Ліцензія перекладу

Умови ліцензування перекладу (не обов'язково мають збігатися із умовами ліцензування на початковий код).

Угода із учасником

Угода користувача, яка має бути затверджена до того, як користувач зможе перекладати цей складник.

Додавання нового перекладу

Спосіб обробки запитів щодо створення нових записів мов. Доступні варіанти:

Зв'язатися з розробниками

Користувач може вибрати бажану мову, а супровідники проекту отримають про це сповіщення. Супровідники зможуть вирішувати, чи слід додавати мову до репозиторію.

Вказати посилання на вказівки з перекладу

Користувачеві буде надано посилання на сторінку, яка описує процедуру започаткування нових перекладів. Скористайтесь цим пунктом, якщо бажаним є формальніший підхід (наприклад, формування команди до того, як буде розпочато переклад).

Створити новий мовний файл

Користувач може вибрати мову, Weblate автоматично створить файл для неї, і переклад можна буде розпочати.

Вимкнути додавання нових перекладів

Користувач не зможе розпочинати переклад новою мовою.

Підказка: Адміністратори проекту можуть додавати нові переклади, навіть якщо тут ця можливість вимкнено, якщо це можливо (у [Шаблон для нових перекладів](#) або форматі файла передбачено підтримку можливості започаткування перекладу з порожнього файла).

Дивись також:

adding-translation, [Додавання нових перекладів](#)

Керування рядками

Нове в версії 4.5.

Визначає, чи зможуть користувачі Weblate додавати нові рядки або вилучати наявні. Вкажіть значення, яке відповідає вашій процедурі локалізації — способу, у яких мають впроваджуватися нові рядки перекладу.

У двомовних форматах рядки, як правило, видобуваються з початкового коду (наприклад, за допомогою `xgettext`), тому додавання нових рядків у Weblate має бути вимкнено (їх має бути відкинуто під час наступного оновлення файлів перекладу). У Weblate ви можете керувати рядками будь-якого перекладу — програма не вимагає синхронності рядків в усіх перекладах.

У одномовних форматах керування рядками можна виконувати лише у початковій мові — рядки автоматично додаються або вилучаються з перекладів. Рядки з'являтимуться у файлах перекладів, щойно їх буде перекладено.

Дивись також:

Двомовні і одномовні формати, *adding-new-strings*, `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Стиль коду мови

Налаштуйте код мови, використаний для створення назви файлу для перекладів, створених Weblate.

Дивись також:

Додавання нових перекладів, *Код мови*, *Обробка кодів мов*

Стиль злиття

Ви можете налаштувати, як будуть оброблятися оновлення з репозиторію основного коду. Реалізація залежить від системи керування версіями, див. *Налаштування інтеграції із керуванням версіями*.

Перебазування

Переносить внески Weblate, накладаючи їх на основне сховище, при оновленні. Так забезпечується якісніший журнал змін без додаткових внесків з об'єднання гілок.

Перенесення може призвести до проблем, якщо злиття є складним. Тому вам слід бути обережним із його вмиканням.

Можливо, вам доведеться увімкнути примусовий запис за допомогою режиму *Git з примусовим «push»* для *Система керування версіями*, особливо якщо запис відбувається до іншої гілки.

Злити

Зміни в основному репозиторії записуються до сховища Weblate. Під час об'єднання буде використано швидке переспрямування, якщо це можливо. Це найбезпечніший спосіб, але він може призвести до багатьох подань на об'єднання гілок.

Об'єднати без швидкого переспрямування

Зміни у основному сховищі записуються до сховища Weblate із окремим внеском об'єднання кожного разу (навіть якщо можливе швидке переспрямування). Усі зміни у Weblate з'являться як внесок для об'єднання у сховищі Weblate.

Типове значення можна змінити за допомогою `DEFAULT_MERGE_STYLE`.

Повідомлення щодо внеску, додавання, вилучення, повідомлення додатків та запитів щодо об'єднання

Повідомлення, яке буде використано при внесенні перекладу, див. *Розмітка шаблонів*.

Типове значення можна змінити за допомогою `DEFAULT_ADD_MESSAGE`, `DEFAULT_ADDON_MESSAGE`, `DEFAULT_COMMIT_MESSAGE`, `DEFAULT_DELETE_MESSAGE`, `DEFAULT_MERGE_MESSAGE`, `DEFAULT_PULL_MESSAGE`.

Відправляти при поданні

Визначає, чи слід записані зміни автоматично записувати до основного репозиторію коду. Якщо увімкнено, запис буде ініційовано одразу після того, як Weblate запише зміни до базового репозиторію (див. «*Лінії*» *внески*). Щоб насправді увімкнути запис, слід також налаштувати *Адресу запису до репозиторію*.

Вік змін для подання

Установлює, наскільки старими (у годинах) мають бути зміни, перш ніж їх буде внесено фоновим завданням або командою керування `commit_pending`. Усі зміни у складнику буде записано у момент, коли зміни у принаймні одному зі складників стануть старішими за вказаний час.

Типове значення можна змінити за допомогою `COMMIT_PENDING_HOURS`.

Підказка: Існують й інші ситуації, коли очікувані зміни може бути подано, див. «*Лінії*» *внески*.

Блокувати при помилці

Блокує складник (і пов'язані складники, див. *Внутрішні адреси Weblate*) після першої з помилки запису або об'єднання у основному репозиторії або після помилки отримання даних з основного репозиторію. Блокування у цьому випадку запобігає додаванню конфліктів, які доведеться вирішувати вручну.

Складник буде автоматично розблоковано, щойно у репозиторії не лишиться помилок.

Джерельна мова

Мова, яку буде використано для початкових рядків. Змініть цю мову, якщо ви перекладаєте з якоїсь мови, відмінної від англійської.

Підказка: Якщо ви перекладаєте двомовні файли з англійської, але хочете мати можливість для виправлення і у перекладі англійською, виберіть *Англійська (розробник)* як початкову мову, щоб уникнути конфлікту між назвою початкової мови і наявним перекладом.

У цьому випадку для одномовних перекладів ви можете скористатися проміжним перекладом, див. *Проміжний мовний файл*.

Фільтр мов

Регулярний вираз, який використовується для фільтрації перекладу при скануванні маски файлу. Цим можна скористатися для обмеження списку мов, якими керує Weblate.

Примітка: Вам слід вказати список кодів мов так, як їх використовуються у назвах файлів.

Ось декілька прикладів фільтрування:

Опис фільтра	Регулярний вираз
Лише вибрані мови	<code>^(cs de es)\$</code>
Виключити мови	<code>^(?! (it fr)\$) .+\$</code>
Фільтрувати лише дволітерні коди	<code>^[..]\$</code>
Виключити файли, які не є мовними	<code>^(?! (blank)\$) .+\$</code>
Включити усі файли (типовий варіант)	<code>^[^.] +\$</code>

Формальний вираз варіантів

Формальний вираз, який використовується для визначення варіантів рядка, див. `variants`.

Примітка: Більшість полів можуть редагувати власники або адміністратори проекту за допомогою інтерфейсу Weblate.

Дивись також:

Чи передбачено у Weblate підтримку інших систем керування версіями, окрім Git і Mercurial?, alerts

Пріоритет

Перекладачам спершу пропонуються складники із більшою перевагою.

Змінено в версії 4.15: У новій версії це також стосується упорядкування відповідних термінів з глосарія.

Обмежений доступ

Типово, складник видимий усім, хто має доступ до проекту, навіть якщо особа не може вносити зміни до складника. Це спрощує підтримання однорідності перекладу у межах проекту.

Обмеження доступу на рівні складника або списку складників має пріоритет над простими правами доступу до складника і не залежить від прав доступу на рівні проекту. Доступ до складника слід надавати явним чином. Зробити це можна надавши новій групі користувачів доступ і записавши до неї користувачів або за допомогою типових груп керування доступом *custom* або *private*.

Типове значення можна змінити за допомогою `DEFAULT_RESTRICTED_COMPONENT`.

Підказка: Це стосується і адміністраторів проекту — будь ласка, переконайтеся, що ви не втратите доступ до складника після перемикання його стану.

Спільне використання даних у проєктах

Ви можете вибрати додаткові проєкти, у яких буде показано складник. Це може бути корисним для бібліотек спільного використання, які є частиною декількох проєктів.

Примітка: Спільне використання складника не змінює параметрів доступу до нього. Він лише стає видимим під час навігації іншими проєктами. Користувачеві все одно потрібен буде доступ до самого складника, щоб здійснювати навігацію ним або перекладати його.

Використовувати як глосарій

Нове в версії 4.5.

Дозволяє використання цього складника як глосарія. Ви можете налаштувати, як його буде показано у списку *Колір глосарія*.

Доступ до глосарія в усіх проєктах, які визначено списком *Спільне використання даних у проєктах*.

Рекомендуємо увімкнути *Керування рядками* для глосаріїв з метою уможливлення додавання до них нових слів.

Дивись також:

Глосарій

Колір глосарія

Колір для глосарія, який буде використано для показу слів-відповідників.

2.7.5 Розмітка шаблонів

Weblate використовує просту мову розмітки у декількох місця, де потрібна обробка тексту. Її засновано на *The Django template language*, тому може бути доволі потужною.

У поточній версії це використовується у

- Форматуванні повідомлень про внесок, див. *Налаштування складників*
- Декількох додатках
 - *Виявлення складників*
 - *Генератор статистики*
 - *Виконання скриптів з додатка*

У шаблонах складників доступні такі змінні:

```
{{ language_code }}
    Код мови

{{ language_name }}
    Назва мови

{{ component_name }}
    Назва складника

{{ component_slug }}
    Ідентифікатор складника

{{ project_name }}
    Назва проєкту
```

```

{{ project_slug }}
    Ідентифікатор проекту
{{ url }}
    Адреса перекладу
{{ filename }}
    Назва файлу перекладу
{{ stats }}
    Статистика перекладу, має атрибути. Приклади наведено нижче.
{{ stats.all }}
    Загальна кількість рядків
{{ stats.fuzzy }}
    Кількість рядків, які потребують рецензування
{{ stats.fuzzy_percent }}
    Частка рядків, які потребують рецензування
{{ stats.translated }}
    Кількість перекладених рядків
{{ stats.translated_percent }}
    Частка перекладених рядків
{{ stats.allchecks }}
    Кількість рядків, які не пройшли перевірки
{{ stats.allchecks_percent }}
    Частка рядків, які не пройшли перевірки
{{ author }}
    Автор поточного внеску, доступне лише у області видимості внеску.
{{ addon_name }}
    Назва поточного виконаного додатка, доступна лише у повідомленні про внесок додатка.

```

У навігаторі репозиторію або шаблонах редактора доступні такі змінні:

```

{{branch}}
    поточна гілка
{{line}}
    рядок у файлі
{{filename}}
    назва файла, ви також можете вилучити початкові частини за допомогою фільтра parentdir; при-
    клад: {{filename|parentdir}}

```

Ви можете поєднати їх за допомогою фільтрів:

```
{{ component|title }}
```

Ви можете скористатися умовами:

```
{% if stats.translated_percent > 80 %}Well translated!{% endif %}
```

Передбачено додатковий тег для заміни символів:

```
{% replace component "-" " " %}
```

Ви можете поєднати його з фільтрами:

```
{% replace component|capfirst "-" " " %}
```

Також передбачено додатковий фільтр для обробки назв файлів:

```
Directory of a file: {{ filename|dirname }}
File without extension: {{ filename|striptext }}
File in parent dir: {{ filename|parentdir }}
It can be used multiple times: {{ filename|parentdir|parentdir }}
```

...та інші можливості шаблонів Django.

2.7.6 Швидкість імпортування

Отримання репозиторію системи керування версіями та імпортування перекладів до Weblate може бути тривалим процесом, залежно від розміру ваших перекладів. Ось декілька підказок:

Оптимізація налаштувань

Типові налаштування є корисними для тестування і діагностування Weblate, а для промислової конфігурації вам слід виконати певні коригування. Багато з налаштувань значно впливають на швидкодію. Будь ласка, ознайомтеся із розділом *Промислові налаштування*, щоб дізнатися більше, особливо з такими підрозділами:

- Налаштуйте Celery для виконання фонових завдань (див. *Фонові завдання з використанням Celery*)
- *Увімкніть кешування*
- *Використання потужного рушія бази даних*
- *Вимкнути режим діагностики*

Перевірка обмежень на ресурси

Якщо ви імпортуєте великі об'єми перекладів або репозиторіїв, ви можете зіткнутися із обмеженнями на ресурси на вашому сервері.

- Перевірте, чи достатньо вільної пам'яті. Кешування файлів перекладу з боку операційної системи може значно підвищити швидкодію.
- Вузьким місцем можуть бути дії із диском, якщо потрібно обробляти багато рядків — дані на диск записуватимуть одразу Weblate і база даних.
- Додаткові ядра процесора можуть підвищити швидкодію виконання фонових завдань (див. *Фонові завдання з використанням Celery*).

Вимкніть непотрібні перевірки

Деякі перевірки якості є доволі витратними щодо ресурсів. Якщо вони є непотрібними, ви можете заощадити певний час під час імпортування, якщо ви пропустите їх. Див. *CHECK_LIST*, щоб дізнатися більше про налаштування.

2.7.7 Автоматичне створення складників

Якщо у проєкт є з десяток файлів перекладу (наприклад, для різних доменів gettext або частин застосунків Android), у вас може виникнути потреба у імпортування цих файлів автоматично. Цього можна досягти або за допомогою командного рядка з використанням `import_project` чи `import_json`, або за допомогою встановлення додатка *Виявлення складників*.

Щоб скористатися додатком, вам слід спочатку створити складник для одного файла перекладу (виберіть той з них, який має найменшу ймовірність перейменування або вилучення у майбутньому), а потім встановити додаток для цього складника.

Для команд керування вам слід створити проєкт, який міститиме усі складники, а потім віддати команду `import_project` або `import_json`.

Дивись також:

Команди керування, Виявлення складників

2.8 Визначення мов

Для належного представлення різних перекладів потрібні дані щодо назви мови, напрямку запису тексту, визначення форм множини та коду мови.

2.8.1 Вбудовані визначення мов

До пакунка Weblate включено близько 600 мов. Кожного випуску список мов зростає. Під час оновлення Weblate (точніше, під час виконання команди **weblate migrate**, перегляньте *Загальні настанови щодо оновлення*) виконується оновлення бази даних мов для включення усіх визначень мов, які є у новому пакунку Weblate.

Оновлення можна вимкнути за допомогою параметра `UPDATE_LANGUAGES`. Ви також можете примусово оновити базу даних так, щоб вона відповідала вбудованим до Weblate даним, за допомогою команди `setuplang`.

Дивись також:

Розширення вбудованих визначень мов, поточні визначення мов

2.8.2 Обробка кодів мов

Під час обробки перекладів Weblate намагається прив'язати код мови (зазвичай, код зі списку ISO 639-1) з *маски файлів складника* до усіх наявних об'єктів мов.

Ви можете поліпшити цю прив'язку на рівні проєкту за допомогою *Альтернативні назви мови*.

Якщо точної прив'язки не буде знайдено, буде зроблено спробу встановлення найкращої відповідності до наявного запису мови. Буде виконано такі кроки:

- Пошук без врахування регістру символів.
- Нормалізація підкреслень та дефісів.
- Пошук вбудованих альтернативних назв мов.
- Пошук за назвою мови.
- Ігнорування типового коду країни для вибраної мови — вибираємо `cs` замість `cs_CZ`.

Якщо і це не допоможе, нове визначення мови буде створено за допомогою типових значень (напрям запису тексту зліва праворуч, одна форма множини). Автоматично створений запис мови із кодом `xx_XX` матиме назву `xx_XX (generated)`. Ви можете змінити цю назву пізніше за допомогою адміністративного інтерфейсу

(див. *Зміна визначень мов*) і повідомити про помилку у системі стеження за вадами (see *Участь у розробці Weblate*), щоб належне визначення було додано у майбутньому випуску Weblate.

Підказка: Якщо у списку мов ви бачите якісь небажані записи, вам варто скоригувати *Фільтр мов* для ігнорування відповідного файла при обробці перекладів.

Дивись також:

Код мови, Додавання нових перекладів

2.8.3 Зміна визначень мов

Ви можете змінити визначення мов за допомогою інтерфейсу роботи з мовами (адреса `/languages/`).

Під час редагування переконайтеся, що усі поля заповнено правильно (особливо, на форми множини та на-прямок запису тексту). Якщо ви цього не зробите, перекладачі не зможуть належним чином редагувати від-повідні переклади.

2.8.4 Неоднозначні коди мови і макромови

Здебільшого, не варто використовувати код макромови для перекладу. Типовим проблемним випадком є курдська мова, яку можна записувати арабською або латинською писемністю, залежно від варіанта. Щоб Weblate обробляв такі випадки належним чином, рекомендуємо використовувати лише коди окремих мов і уникати використання кодів макромов.

Дивись також:

Визначення макромов, Список макромов

2.8.5 Визначення мов

Кожен запис мови складається з таких полів:

Код мови

Код ідентифікації мови. Weblate надає перевагу дволітерним кодам, які визначаються стандартом [ISO 639-1](#), але використовує коди [ISO 639-2](#) або [ISO 639-3](#) для мов, які не мають дволітерного коду. Крім того, перед-бачено підтримку розширених кодів, які визначено стандартом [BCP 47](#).

Дивись також:

Обробка кодів мов, Додавання нових перекладів

Назва мови

Видима назва мови. Назви мов, які включено до Weblate, також локалізуються, залежно від мови інтерфейсу користувача.

Напрямок тексту

Визначає, записується мова справа ліворуч чи зліва праворуч. Ця властивість автоматично визначається належним чином для більшості мов.

Кількість форм множини

Кількість форм множини, які використовуються у мові.

Форма множини

Для визначення, яку форму множини буде використано для вказаної кількості, буде використано формулу множини, яка сумісна із Gettext.

Дивись також:

Множина, інструменти GNU gettext: форми множини, правила форм множини у мовах від консорціуму Unicode

Кількість мовців

Кількість людей, які розмовляють цією мовою.

2.8.6 Додавання нових перекладів

Змінено в версії 2.18: У версіях до 2.18 поведінка при додаванні нових перекладів була специфічною для формату файлів.

Weblate може автоматично розпочинати новий переклад для всіх форматів файлів.

У деяких форматах усе починається з порожнього файла, до якого додаються лише перекладені рядки (наприклад, *Рядкові ресурси Android*), у інших же від початку зберігаються усі ключі (наприклад, *GNU Gettext*). У форматах, які засновано на документах (наприклад, у *Формат OpenDocument*) починаємо з копії початкового документа, де усі рядки позначено як такі, що потребують редагування. У деяких випадках вміст залежить не від формату, а від оболонки, якою ви користуєтесь для обробки перекладу (наприклад, *файли JSON*).

Якщо ви вказуєте *шаблон для нових перекладів у складнику*, Weblate використає цей файл для започаткування перекладів. Усі наявні переклади буде вилучено з файла, коли ви так зробите.

Якщо вміст поля *Шаблон для нових перекладів* є порожнім, і у форматі файлів передбачено підтримку відповідних файлів, буде створено порожній файл, до якого додаватимуться нові рядки, щойно буде отримано їхній переклад.

За допомогою поля *Стиль коду мови* ви можете налаштувати код мови, який буде використано у назвах створених файлів:

Типовість на основі формату файлу

Залежить від формату файлів. Для більшості форматів використовується POSIX.

Стиль POSIX з підкресленням як роздільником

Типово, використовується у gettext і пов'язаних інструментів. Створює коди мов, подібні до pt_BR.

Стиль POSIX з підкресленням як роздільником, включно із кодом країни

Код мови у стилі POSIX. Включає код країни, навіть якщо у цьому немає потреби (наприклад, cs_CZ).

Стиль VCP з дефісом як роздільником

Типово, використовується на вебплатформах. Створює коди мов, подібні до pt-BR.

Стиль VCP з дефісом як роздільником, включно з кодом країни

Код мови у стилі VCP. Включає код країни, навіть якщо у цьому немає потреби (наприклад, cs-CZ).

Стиль ВСП з дефісом як роздільником, включно із застарілими кодами мов

Використовує застарілі коди для китайської та позначень у стилі ВСП.

Стиль ВСП з дефісом як роздільником, малими літерами

Позначення у стилі ВСП, усі малими літерами (наприклад, `cs-cz`).

Стиль метаданих Apple App Store

Стиль, що пасує для вивантаження метаданих до крамниці Apple App.

Стиль метаданих Google Play

Стиль, що пасує для вивантаження метаданих до крамниці Google Play.

Стиль Android

Використовується лише у застосунках Android. Створює коди мов, подібні до `pt-rBR`.

Стиль Linux

Локалі у формі, яку використано у Linux, з застарілими кодами для китайської та позначеннями у стилі POSIX.

Крім того, будь-які прив'язки, які визначено у *Альтернативні назви мови*, працюють і у зворотному напрямку.

Примітка: Weblate розпізнає будь-які з цих стилів при обробці файлів перекладів. Згадані вище параметри впливають лише на те, як створюватимуться нові файли.

Дивись також:

Код мови, Альтернативні назви мови, Обробка кодів мов

2.9 Безперервна локалізація

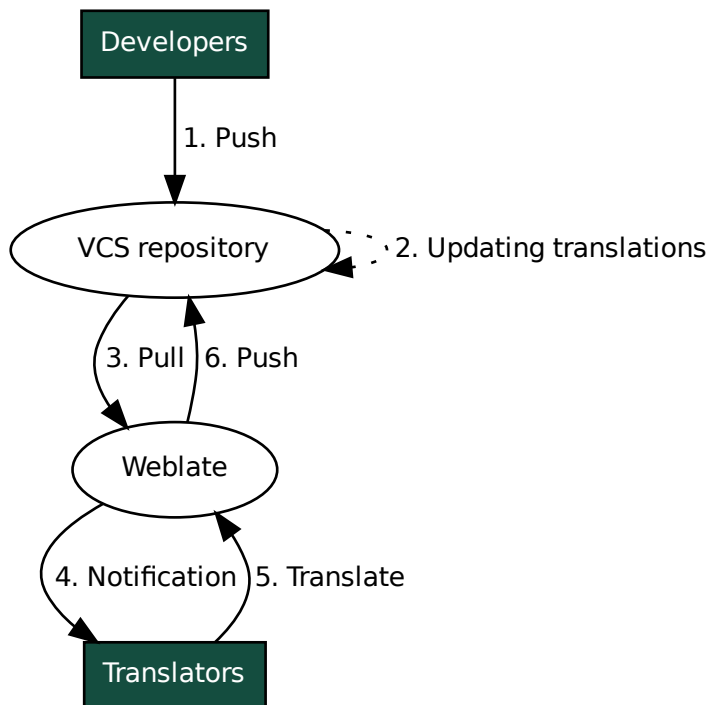
Готовою до використання є інфраструктура, за допомогою якої ви можете точно слідувати за розробкою проєкту. Перекладачі можуть працювати над перекладами неперервно, а не працювати над величезними обсягами перекладів нового тексту безпосередньо перед випуском.

Дивись також:

У `/dev/integration` описано базові способи інтеграції вашої розробки із Weblate.

Ось процедура:

1. Розробники змінюють код і записують його до репозиторію системи керування версіями.
2. Якщо потрібно, оновлюються файли перекладу (це залежить від формату файлів, див. *Чому Weblate показує застарілі рядки перекладу, хоча шаблон перекладу було оновлено?*).
3. Weblate отримує зміни з репозиторію системи керування версіями, див. *Оновлення репозиторіїв*.
4. Щойно Weblate виявить зміни в перекладах, перекладачів буде сповіщено відповідно до налаштувань підписки.
5. Перекладачі подають переклади за допомогою вебінтерфейсу Weblate або вивантажують зроблені поза інтернетом переклади.
6. Щойно переклад буде завершено, Weblate надсилає зміни до локального репозиторію (див. *«Лінії» внески*) і записує їх до основного репозиторію, якщо має право на виконання цієї дії (див. *Записування змін з Weblate*).



2.9.1 Оновлення репозиторіїв

Вам слід налаштувати певний спосіб оновлення репозиторіїв з початкового коду.

- Скористайтеся *Обробники сповіщень* для інтеграції із більшістю типових служб зберігання коду:
 - Автоматичне отримання змін з *GitHub*
 - Автоматичне отримання змін з *GitLab*
 - Автоматичне отримання змін з *Bitbucket*
 - Автоматичне отримання змін з *Pagure*
 - Автоматичне отримання змін зі репозиторіїв *Azure*
 - Автоматичне отримання змін зі репозиторіїв *Gitea*
- Увімкніть оновлення вручну або у керуванні репозиторієм, або за допомогою *Програмний інтерфейс REST Weblate* або *Клієнт Weblate*
- Увімкніть *AUTO_UPDATE* для автоматичного оновлення всіх складників у вашому екземплярі Weblate
- Виконайте *updategit* (із вибором проекту або додаванням `--all` для оновлення усього)

Кожного разу, коли оновлюватиме Weblate, буде увімкнено додатки остаточної обробки, див. *Додатки*.

Уникання конфліктів об'єднання

Конфлікти об'єднання з даними Weblate виникають тоді, коли якийсь файл було змінено одразу у Weblate і поза ним. Існує два підходи до усунення можливості конфліктів — заборона редагування поза Weblate або інтеграція Weblate до вашого процесу оновлення так, щоб зміни було отримано до оновлення файлів поза системою Weblate.

Перший підхід спрощує роботу із одномовними файлами — ви можете додавати нові рядки у Weblate і покластися на редагування файлів у вбудованому редакторі. Для двомовних файлів, зазвичай, перекладу передую певна процедура видобування рядків для створення придатних до перекладу файлів із початкового коду. У деяких випадках цей процес може бути поділено на дві частини — спочатку під час видобування створюється шаблон (наприклад, файл POT gettext створюється за допомогою **xgettext**), а потім, у подальшому процесі, об'єднується із поточними перекладами (файли PO gettext оновлюються за допомогою **msgmerge**). Ви можете виконати другий крок у Weblate, і програмні засоби забезпечать включення усіх змін у черзі до виконання цієї дії.

Другий підхід можна реалізувати за допомогою *Програмний інтерфейс REST Weblate* — змусити Weblate записати усі зміни з черги і заблокувати переклад, доки ви виконуєте зміни на вашому боці.

Скрипт для виконання оновлень може виглядати ось так:

```
# Lock Weblate translation
wlc lock
# Push changes from Weblate to upstream repository
wlc push
# Pull changes from upstream repository to your local copy
git pull
# Update translation files, this example is for Django
./manage.py makemessages --keep-pot -a
git commit -m 'Locale updates' -- locale
# Push changes to upstream repository
git push
# Tell Weblate to pull changes (not needed if Weblate follows your repo
# automatically)
wlc pull
# Unlock translations
wlc unlock
```

Якщо у одному репозиторії зберігаються дані декількох складників, вам слід заблокувати їх окремо:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Примітка: У прикладі використано програму *Клієнт Weblate*, яка потребує налаштування (ключів до програмного інтерфейсу), щоб Weblate можна було керувати віддалено. Ви також можете досягти цього використанням замість wlc будь-якого клієнта HTTP, наприклад curl, див. *Програмний інтерфейс REST Weblate*.

Дивись також:

Клієнт Weblate

Автоматичне отримання змін з GitHub

Weblate постачається із вбудованою підтримкою GitHub.

Якщо ви користуєтесь Hosted Weblate, рекомендованим підходом є встановлення застосунки [Weblate](#). У такий спосіб ви отримаєте належну конфігурацію без зайвої метушні із налаштуванням. Нею також можна використати для записування змін назад до репозиторію.

Щоб отримувати сповіщення щодо кожного запису до репозиторію GitHub, додайте вебскрипт Weblate до параметрів репозиторію (*Webhooks*), як це показано на наведеному нижче зображенні:

The screenshot shows the GitHub 'Add webhook' interface. On the left is a sidebar with navigation links: Options, Collaborators & teams, Branches, Webhooks (highlighted), Integrations & services, Deploy keys, and Alerts. The main content area is titled 'Webhooks / Add webhook' and contains the following fields and options:

- Webhooks / Add webhook**: Section header.
- Introduction**: A paragraph explaining that a POST request will be sent to the URL with details of subscribed events. It mentions data formats like JSON and x-www-form-urlencoded.
- Payload URL ***: A text input field containing `https://hosted.weblate.org/hooks/github/`.
- Content type**: A dropdown menu set to `application/x-www-form-urlencoded`.
- Secret**: An empty text input field.
- SSL verification**: A checkbox labeled 'By default, we verify SSL certificates when delivering payloads.' with a red button to 'Disable SSL verification'.
- Which events would you like to trigger this webhook?**: Three radio button options:
 - ☒ Just the push event.
 - ☐ Send me everything.
 - ☐ Let me select individual events.
- Active**: A checked checkbox with the text 'We will deliver event details when this hook is triggered.'
- Add webhook**: A green button at the bottom.

The footer of the page shows copyright information for GitHub, Inc. (2018) and links to Terms, Privacy, Security, Status, Help, Contact GitHub, API, Training, Shop, Blog, and About.

Щоб отримати адресу даних, допишіть `/hooks/github/` до адреси сайту вашого Weblate. Наприклад, для служби Hosted Weblate це `https://hosted.weblate.org/hooks/github/`.

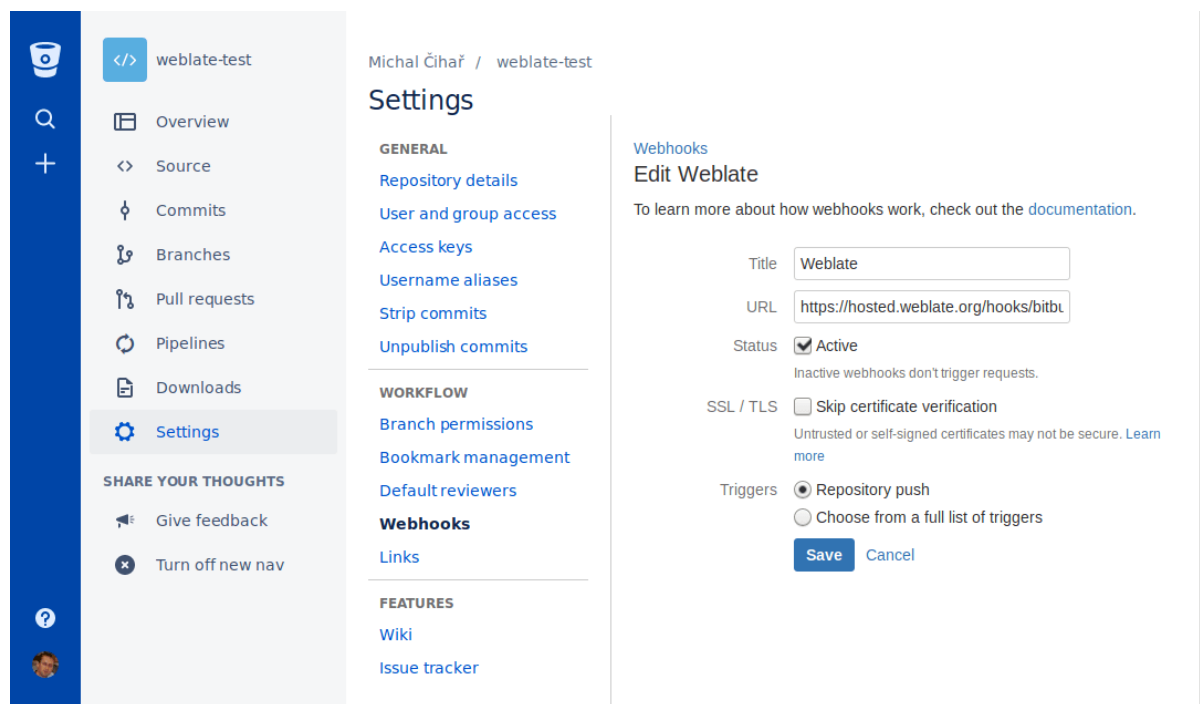
Ви можете лишити типові значення для решти параметрів (Weblate може одночасно обробляти обидва типи даних і споживатиметься лише подію *push*).

Дивись також:

POST /hooks/github/, Доступ до penozumopiie з Hosted Weblate

Автоматичне отримання змін з Bitbucket

У Weblate передбачено підтримку вебскриптів Bitbucket. Додайте вебскрипт, який вмикає запис до репозиторію із адресою призначення `/hooks/bitbucket/` у вашому встановленому Weblate (наприклад, `https://hosted.weblate.org/hooks/bitbucket/`).



Дивись також:

POST /hooks/bitbucket/, Доступ до репозиторіїв з Hosted Weblate

Автоматичне отримання змін з GitLab

У Weblate передбачено підтримку скриптів GitLab. Додайте вебскрипт проєкту із адресою призначення `/hooks/gitlab/` у встановленому вами Weblate (наприклад, `https://hosted.weblate.org/hooks/gitlab/`).

Дивись також:

POST /hooks/gitlab/, Доступ до репозиторіїв з Hosted Weblate

Автоматичне отримання змін з Pagure

Нове в версії 3.3.

У Weblate передбачено підтримку скриптів Pagure. Додайте вебскрипт із адресою призначення `/hooks/pagure/` у встановленому вами Weblate (наприклад, `https://hosted.weblate.org/hooks/pagure/`). Зробити це можна за допомогою пункту *Activate Web-hooks* у розділі *Project options*:

The screenshot shows the 'Project Settings' page for a project named 'nijel-test'. The left sidebar contains a list of settings categories: Project Settings, Project Details, Default Branch, Private Web Hook Key, API Keys, Project Options (selected), Public Notifications, Users & Groups, Deploy Keys, Hooks, Priorities, Roadmap, Close Status, Custom Issue Fields, Reports, Tags, Quick Replies, Regenerate Repos, Give Project, and Delete Project. The main content area is titled 'Project Options' and contains several checkboxes for enabling various features. The 'Activate Web-hooks' section shows a text input with the URL 'https://hosted.weblate.org/hooks/pagure/' and buttons for 'Update' and 'Test web-hook'. Below this, there is a 'Learn more about' section with links to documentation for Flags, Tracker read-only, Pull-request access only, Roadmap on Issue page, and fedmsg notifications.

Project Options

- ☐ Activate always merge
- ☐ Activate disable non fast-forward merges
- ☐ Activate Enforce signed-off commits in pull-request
- ☒ Activate fedmsg notifications
- ☒ Activate Issue tracker
- ☐ Activate Issue tracker read only
- ☐ Activate Issues default to private
- Activate Minimum score to merge pull-request :
- ☐ Activate notify on commit flag
- ☐ Activate notify on pull-request flag
- ☐ Activate Only assignee can merge pull-request
- ☐ Activate open metadata access to all
- ☐ Activate project documentation
- ☐ Activate pull request access only
- ☒ Activate pull requests
- ☒ Activate stomp notifications

Activate Web-hooks :

[Update](#) [Test web-hook](#)

Learn more about

- [Flags](#)
- [Tracker read-only](#)
- [Pull-request access only](#)
- [Roadmap on Issue page](#)
- [fedmsg notifications](#)

Дивись також:

POST /hooks/pagure/, Доступ до репозиторіїв з Hosted Weblate

Автоматичне отримання змін зі репозиторіїв Azure

Нове в версії 3.8.

У Weblate передбачено підтримку веб скриптів репозиторіїв Azure. Додайте вебскрипт для події *Code pushed* із адресою призначення `/hooks/azure/` у встановленому вами Weblate (наприклад, `https://hosted.weblate.org/hooks/azure/`). Зробити це можна за допомогою пункту *Service hooks* у розділі *Project settings*.

Дивись також:

Вебскрипти у підручнику розробника Azure, *POST /hooks/azure/, Доступ до репозиторіїв з Hosted Weblate*

Автоматичне отримання змін зі репозиторіїв Gitea

Нове в версії 3.9.

У Weblate передбачено підтримку вебскриптів Gitea. Додайте *Gitea Webhook* для події *Push events* із адресою призначення `/hooks/gitea/` у встановленому вами Weblate (наприклад, `https://hosted.weblate.org/hooks/gitea/`). Зробити це можна за допомогою пункту *Webhooks* у розділі *Settings* репозиторію.

Дивись також:

Вебскрипти у підручнику з Gitea, *POST /hooks/gitea/*, *Доступ до репозиторіїв з Hosted Weblate*

Автоматичне отримання змін зі репозиторіїв Gitee

Нове в версії 3.9.

У Weblate передбачено підтримку вебскриптів Gitee. Додайте *WebHook* для події *Push* із адресою призначення `/hooks/gitee/` у встановленому вами Weblate (наприклад, `https://hosted.weblate.org/hooks/gitee/`). Зробити це можна за допомогою пункту *WebHooks* у розділі *Management* репозиторію.

Дивись також:

Підручник щодо вебскриптів у Gitee, *POST /hooks/gitee/*, *Доступ до репозиторіїв з Hosted Weblate*

Автоматичне щоденне оновлення репозиторіїв

Weblate автоматично отримує вміст віддалених репозиторіїв щодня для поліпшення швидкодії при наступному об'єднанні змін. Якщо хочете, можете перетворити це на щоденні об'єднання, увімкнувши *AUTO_UPDATE*.

2.9.2 Записування змін з Weblate

Для кожного складника перекладу може бути налаштовано адресу запису (див. *Адреса для записування до репозиторію*). Якщо таку назву налаштовано, Weblate зможе записувати зміни до віддаленого репозиторію. Крім того, Weblate можна налаштувати на автоматичне записування змін при кожному внеску (це типова поведінка, див. *Відправляти при поданні*). Якщо ви не хочете, щоб зміни записувалися автоматично, ви можете записувати їх вручну у розділі *Супровід репозиторію* або за допомогою програмного інтерфейсу: *wlc push*.

Параметри запису до репозиторію є різними для різних *Налаштування інтеграції із керуванням версіями*. Подробиці можна знайти у цьому розділі.

Якщо вам не потрібне безпосереднє записування до репозиторію від Weblate, передбачено підтримку запитів щодо об'єднання *Запити щодо злиття GitHub*, *Запити щодо об'єднання GitLab*, *Запити щодо злиття Gitea*, *Запити щодо об'єднання Pagure* або рецензувань *Gerrit*. Ви можете активувати їх вибором *GitHub*, *GitLab*, *Gitea*, *Gerrit* або *Pagure* як *Система керування версіями* у розділі *Налаштування складників*.

Загалом, у Git, GitHub і GitLab можна скористатися такими параметрами:

Бажане налаштування	Система керування версіями	Адреса для записування до репозиторію	Гілка для запису
Без запису	<i>Git</i>	<i>empty</i>	<i>empty</i>
Записувати безпосередньо	<i>Git</i>	Адреса SSH	<i>empty</i>
Записувати до окремої гілки	<i>Git</i>	Адреса SSH	Назва гілки
Запит щодо об'єднання у GitHub з відгалуження	<i>Запити щодо злиття GitHub</i>	<i>empty</i>	<i>empty</i>
Запит щодо об'єднання у GitHub з гілки	<i>Запити щодо злиття GitHub</i>	SSH URL ¹	Назва гілки
Запит щодо злиття на GitLab з відгалуження	<i>Запити щодо об'єднання GitLab</i>	<i>empty</i>	<i>empty</i>
Запит щодо злиття на GitLab з гілки	<i>Запити щодо об'єднання GitLab</i>	SSH URL ^{c. 301, 1}	Назва гілки
Запит щодо злиття на Gitea з відгалуження	<i>Запити щодо злиття Gitea</i>	<i>empty</i>	<i>empty</i>
Запит щодо злиття на Gitea з гілки	<i>Запити щодо злиття Gitea</i>	SSH URL ¹	Назва гілки
Запит щодо злиття на Pagure з відгалуження	<i>Запити щодо об'єднання Pagure</i>	<i>empty</i>	<i>empty</i>
Запит щодо злиття на Pagure з гілки	<i>Запити щодо об'єднання Pagure</i>	SSH URL ¹	Назва гілки

Примітка: Ви також можете увімкнути автоматичний запис змін після внесків Weblate. Зробити це можна у *Відправляти при поданні*.

Дивись також:

Опис налаштування ключів SSH можна знайти у розділі *Доступ до репозиторіїв*. Дані щодо того, яким чином Weblate визначає потребу у внесенні змін, можна знайти у розділі *«Лінійні» внески*.

Захищені гілки

Якщо ви використовуєте Weblate для захищеної гілки, ви можете налаштувати його на використання запитів щодо об'єднання і увімкнути рецензування перекладів (може бути проблематичним для мов, яких ви не знаєте). Альтернативним підходом є відмова від цього обмеження для користувача Weblate, який записуватиме дані до репозиторію.

Наприклад, у GitHub це можна зробити у налаштуваннях репозиторію:

¹ Може бути порожнім, якщо у *репозиторії* передбачено запис змін.

☒ **Require pull request reviews before merging**

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

Required approving reviews: 1 ▾

☐ **Dismiss stale pull request approvals when new commits are pushed**

New reviewable commits pushed to a matching branch will dismiss pull request review approvals.


☐ **Require review from Code Owners**

Require an approved review in pull requests including files with a designated code owner.


☒ **Restrict who can dismiss pull request reviews**

Specify people or teams allowed to dismiss pull request reviews.

People and teams that can dismiss reviews.

**Organization and repository administrators**

These members can always dismiss.

**weblate**

Weblate push user

×

2.9.3 Взаємодія із іншими

Weblate спрощує взаємодію із іншими учасниками проекту за допомогою програмного інтерфейсу.

Дивись також:

Програмний інтерфейс REST Weblate

2.9.4 «Лінійні» внески

Weblate групує внески від одного автора так, щоб остаточний внесок був якомога більшим. Це значно зменшує кількість внесків, але призводить до того, що у вас може виникнути потреба явним чином ініціювати внески, якщо ви хочете підтримувати синхронізацію з репозиторієм системи керування версіями, наприклад, для злиття (таке злиття типово увімкнено для групи *Managers*, див. *Список привілеїв і вбудовані ролі*).

Зміни у цьому режимі вносяться, щойно буде виконано будь-яку з таких умов:

- Хтось інший вніс зміни до вже зміненого рядка.
- Сталося злиття коду з основної гілки розробки.
- Надіслано запит на явний внесок.
- Надіслано запит щодо отримання файла.
- Зміна є старішою за часовий проміжок, визначений як *Вік змін для подання у Налаштовування складників*.

Підказка: Внески створюються для кожного складника. Отже, якщо у вас багато складників, у вас буде багато внесків. Щоб зменшити кількість, ви можете скористатися додатком *Сполучити Git подання*.

Якщо ви хочете вносити зміни частіше і без перевірки віку, ви можете запланувати регулярне задання для створення внесків. Зробити це можна за допомогою пункту *Періодичні завдання* і *адміністративного інтерфейсу*. Спочатку створіть бажаний *Інтервал* (наприклад 120 секунд). Далі, додайте періодичне завдання і виберіть `weblate.trans.tasks.commit_pending`, як *Завдання* з `{"hours": 0}` у полі *Аргументи ключових слів* та бажаним інтервалом.

2.9.5 Обробка репозиторію зі скриптами

Спосіб взаємодії Weblate з репозиторієм коду можна змінити за допомогою *Додатки*. Зверніться до розділу *Виконання скриптів з додатка*, щоб дізнатися про те, як виконувати зовнішні скрипти з додатків.

2.9.6 Підтримання синхронізації перекладів у різних складниках

Якщо у вас декілька складників перекладу, у вас може виникнути потреба у тому, щоб однакові рядки мали однакові переклади. Досягти такої синхронізації можна на декількох рівнях.

Синхронізація перекладу

Якщо увімкнено *Дозволити поширення перекладу* (типова поведінка, див. *Налаштовування складників*), усі нові переклади автоматично виконуватимуться в усіх складниках із відповідними рядками. Авторство таких перекладів буде визначено належним чином — автором в усіх складниках вважатиметься той, хто здійснив переклад в одному із складників.

Примітка: Поширення перекладів потребує використання однакового ключа для одномовних форматів перекладу. Це слід мати на увазі при створенні ключів перекладу.

Перевірка коректності

Перевірка *Неузгодженість* вважається непройденою, якщо рядки є різними. Ви можете скористатися нею для рецензування таких відмінностей вручну і вибору належного перекладу.

Автоматичний переклад

Автоматичний переклад на основі різних складових може бути використано як спосіб синхронізації перекладів у різних складниках. Ви можете увімкнути автоматичний переклад вручну (див. *Автоматичний переклад*) або налаштувати його автоматичне виконання при оновленні репозиторію за допомогою додатка (див. *Автоматичний переклад*).

2.10 Ліцензування перекладів

Ви можете вказати, за якими умовами ліцензування розповсюджуватимуться переклади. Це особливо важливо, якщо переклади відкриті для всіх, щоб заздалегідь визначити умови використання перекладів.

Вам слід вказати дані ліцензування *складника*. Вам слід уникати вимог щодо узгодження умов ліцензування для учасників команди перекладу, хоча висування таких вимог є можливим.

2.10.1 Умови ліцензування

Якщо вказати дані щодо ліцензування (назву ліцензії та адресу), ці відомості буде показано у розділі даних щодо перекладу відповідного *складника*.

Зазвичай, це найкраще місце для розташування даних ліцензування, якщо вимоги не висуваються явним чином. Якщо ваш проєкт або переклад не є вільними, вам, найімовірніше, слід попередньо висунути вимоги щодо ліцензування.

2.10.2 Угода із учасником

Якщо ви вкажете угоду учасника команди перекладу, лише користувачі, які погодяться із нею, зможуть взяти участь у перекладі. Це буде явним чином видимий крок при доступі до перекладу:

Contribution to this translation requires you to agree with a contributor agreement. [View contributor agreement](#)

Language	Translated	Unfinished	Unfinished words	Unfinished characters	Checks	Suggestions	Comments
Czech 🇨🇪 GPL-3.0	✓						
Hebrew 🇮🇱 GPL-3.0	✓						
Hungarian 🇭🇺 GPL-3.0	81%	4	5	32			
English 🇬🇧 GPL-3.0	✓						

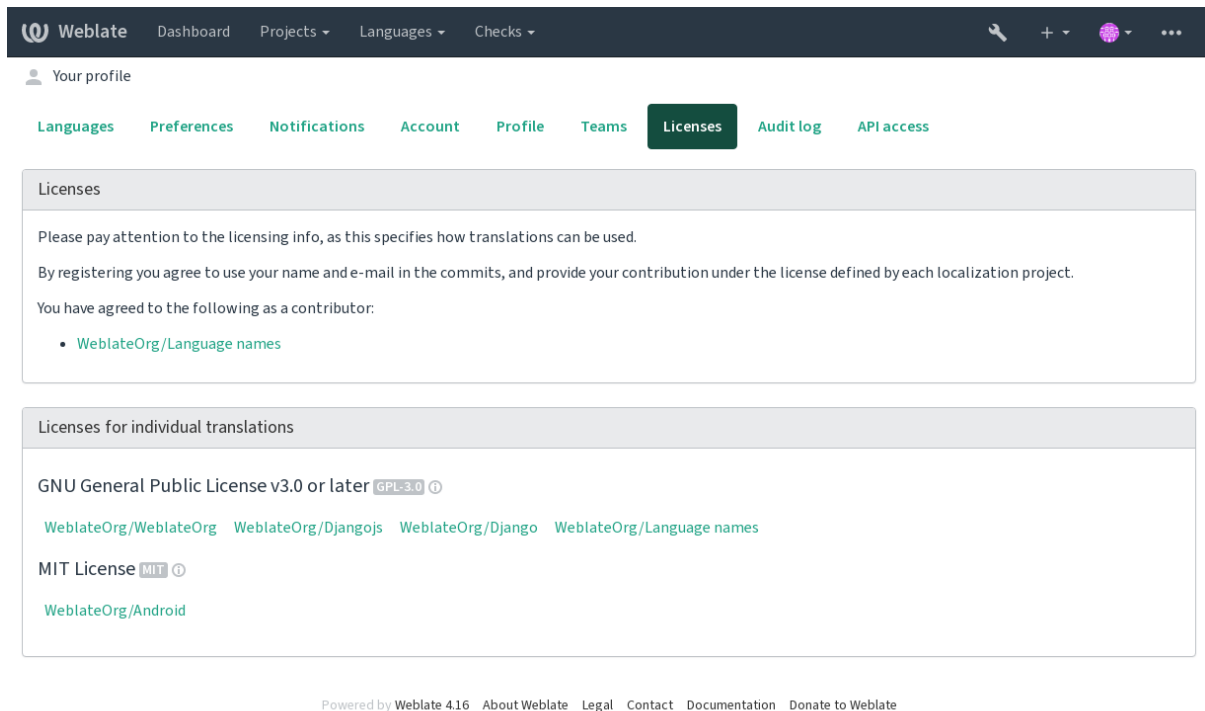
[Start new translation](#)

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Введений текст буде форматовано за абзацами. Може бути включено зовнішні посилання. Не можна використовувати розмітку HTML.

2.10.3 Ліцензування для користувачів

Будь-який користувач може ознайомитися із усіма ліцензійними угодами щодо перекладу усіх відкритих проєктів екземпляра Weblate за допомогою сторінки профілю:



The screenshot shows the Weblate web application interface. At the top is a dark navigation bar with the Weblate logo and links for Dashboard, Projects, Languages, and Checks. Below this is a user profile section with a dropdown menu containing: Languages, Preferences, Notifications, Account, Profile, Teams, Licenses (highlighted), Audit log, and API access. The 'Licenses' page content includes a header 'Licenses', a warning about licensing info, a statement about agreeing to terms, and a list of licenses: 'WeblateOrg/Language names'. A second section titled 'Licenses for individual translations' lists 'GNU General Public License v3.0 or later' and 'MIT License', each with associated project links like 'WeblateOrg/WebplateOrg' and 'WeblateOrg/Djangojs'.

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

2.11 Процес перекладу

2.11.1 Голосування за пропозицію

Типово, надсилати пропозиції може кожен, а приймати пропозиції можуть зареєстровані користувачі. Можна скористатися голосуванням за пропозиції, коли рядок використовується, лише якщо за нього проголосувала більшість зареєстрованих користувачів. Для цього у налаштуваннях *складника* слід увімкнути *Голосування за пропозиції* і встановити порогове значення *Автоматичне прийняття пропозиції* для визначення порогового значення для прийнятих пропозицій (до голосів буде включено голос користувача, який вніс пропозицію, якщо він може голосувати).

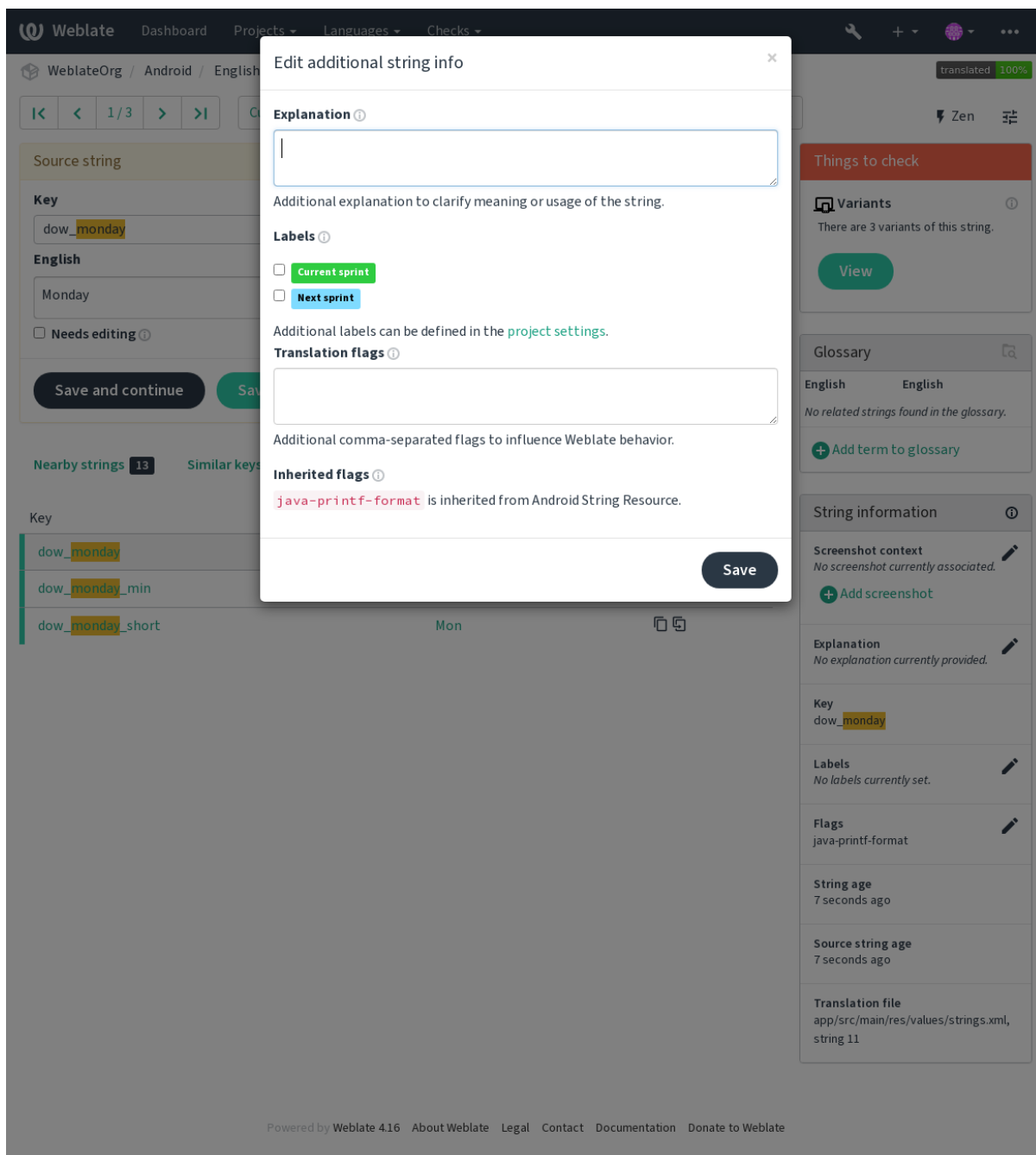
Примітка: Щойно буде налаштовано автоматичне прийняття, звичайні користувачі втратять право безпосередньо зберігати переклади або приймати пропозиції. Це можна перевизначити за допомогою прав доступу *Змінювати рядок з примусовими пропозиціями* (див. *права доступу*).

Ви можете поєднувати ці можливості за допомогою *керування доступом* у одну за допомогою таких конфігурацій:

- Користувачі пропонують переклади і голосують за пропозиції, а обмежена група контролює, що саме буде прийнято. - Увімкніть голосування. - Вимкніть автоматичне прийняття. - Забороніть користувачам зберігати переклади.
- Користувачі пропонують переклади та голосують за переклади з автоматичним прийняттям, щойно буде досягнуто визначеної кількості позитивних голосів. - Увімкніть голосування. - Установіть бажану кількість голосів для автоматичного прийняття.
- Необов'язкове голосування за пропозиції. (Може, якщо треба, використовуватися користувачами, якщо вони не певні щодо перекладу і надають декілька пропозицій.) - Лише увімкніть голосування.

2.11.2 Додаткові відомості щодо початкових рядків

Удоскональте процес перекладу наданням даних щодо рядків, зокрема пояснень, пріоритетності рядків, прапорців перевірки та візуального контексту. Деякі з цих даних може бути видобути з файлів перекладу, а деякі можна додати під час редагування додаткових відомостей щодо рядка:



Доступ до можливості можна отримати безпосередньо з інтерфейсу перекладу, натиснувши піктограму «Редагувати», яку розташовано поруч із написом *Контекст знімка вікна* або *Прапорці*.

Weblate / Django / Czech / Translate

Dashboard Projects Languages Checks

<

>

11 / 26

All strings ▾

Position and priority ▾

Zen

Translation

Explanation

Help text for automatic translation tool

English

Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.

Czech

Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.

☐ Needs editing ⓘ

Save and continue

Save and stay

Suggest

Skip

Nearby strings 26

Comments

Automatic suggestions

Other languages 4

History

Context	English	Czech	Actions
	Files	Soubory	
	Automatic translation	Automatický překlad	
	Add new translation string	Add new translation string	
	Translation status	Stav překladu	
	Singular %(count)s word Plural %(count)s words	One %(count)s slovo Few %(count)s slova Many %(count)s slov	
	Other components	Další součásti	
	Translation file	Soubor s překladem	
	Download	Stáhnout	
	Browse all translation changes	Procházet všechny změny v překladu.	
	Automatic translation takes existing translations in this project and applies them to the current component. It can be used to push translations to a different branch, to fix inconsistent translations or to translate a new component using translation memory.	Automatický překlad použije stávající překlady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí překladové paměti.	
	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.	
	You can add new translation string here, it will automatically appear in all translations.	Zde můžete přidat nový řetězec k překladu, automaticky se objeví ve všech jazycích.	
	The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.	Nahráný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nezapomeňte to povolit.	
	The uploaded file will be merged with the current translation.	Nahráný soubor bude sloučen se stávajícími překlady.	
	The fulltext search might not work properly as the fulltext index for this translation is not yet up to date.	Fulltextové vyhledávání nemusí fungovat správně, protože fulltextový index pro tento překlad ještě není plně zpracován.	
	Review	Kontroloa	
	Review translations touched by other users.	Zkontrolovat překlady od ostatních uživatelů.	
	Start review	Začít kontrolu	
	Percent	Procenta	
	Total	Celkem	
	Failing check	Neúspěšných kontrol	
	Last activity	Poslední aktivity	
	Last change	Poslední změna	
	Last author	Poslední autor	
Question for a mathematics-based CAPTCHA, the %s is an arithmetic problem	What is %s?	Kolik to je?	
	The string uses three dots (...) instead of an ellipsis character (...)		

Glossary

English Czech

machine translation strojový překlad [weblate.org](#)

project projekt [weblate.org](#)

+ Add term to glossary

String information ⓘ

Screenshot context

No screenshot currently associated.

+ Add screenshot

Explanation

Help text for automatic translation tool

Labels

No labels currently set.

Flags

No flags currently set.

Source string location

website/templates/translation.html#1212

String age

2 seconds ago

Source string age

2 seconds ago

Translation file

website/locale/cs/LC_MESSAGES/django.po, string 11

Установлення пріоритетності рядків

Нове в версії 2.0.

Пріоритетність рядків може бути змінено так, щоб рядки з вищою пріоритетністю було перекладено раніше, за допомогою прапорця пріоритет.

Підказка: Цим можна скористатися для логічного упорядкування перекладу.

Дивись також:

Перевірки якості

Прапорці перекладу

Нове в версії 2.4.

Змінено в версії 3.3: Раніше ця можливість мала назву *Прапорці перевірок якості*. У поточній версії ця можливість налаштовує не лише перевірки.

Налаштування перевірок якості та іншої поведінки Weblate, див. *Налаштовування поведінки за допомогою прапорців*.

Прапорці рядка також успадковуються від *прапорців складника у складнику* та прапорців з файла перекладу (див. *Підтримувані формати файлів*).

Дивись також:

Перевірки якості, Налаштовування поведінки за допомогою прапорців

Пояснення

Змінено в версії 4.1: У попередніх версіях це називалося *додатковим контекстом*.

Скористайтесь поясненням для того, що зробити зрозумілішою область або використання перекладу. Ви можете скористатися кодом Markdown для включення посилань та визначення розмітки.

Візуальний контекст для рядків

Нове в версії 2.9.

Ви можете вивантажити знімок вікна, на якому показано якийсь початковий рядок, який використовується у вашій програмі. Це допоможе перекладачам зрозуміти, де він використовується, і як його має бути перекладено.

Оновлений знімок вікна показано на бічній панелі контексту перекладу:

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows a translation task for the string 'Help text for automatic translation tool'. The English source text is 'Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.' The Czech target text is 'Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.' The interface includes buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. A sidebar on the right contains a 'Glossary' section with terms like 'machine translation' and 'project', and a 'String information' section with details like 'Source string location' and 'String age'.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Окрім *Додаткові відомості щодо початкових рядків*, знімки вікон мають окремий інтерфейс керування за допомогою меню *Знаряддя*. Вивантажуйте знімки вікон, пов'яжіть з ними початкові рядки вручну, або скористайтеся для цього оптичним розпізнаванням символів.

Після вивантаження знімка вікна цей інтерфейс обробляє керування і прив'язку початкових рядків:

WebateDashboardProjectsLanguagesChecks

WebateOrg / Django / Screenshots / Automatic translation

Screenshot has been uploaded, you can now assign it to source strings.

Assigned source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			
Screenshot is shown to add visual context for all listed source strings.			

Assign source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			

Source string search

Search

Automatically recognize

Image

Source string

Hello, world!

OneOrangutan has %d banana.

OtherOrangutan has %d bananas.

Try Weblate at <http://demo.weblate.org/>!

Thank you for using Weblate.

Screenshot is shown to add visual context for all listed source strings.

Edit screenshot

Screenshot name

Automatic translation

Image

Currently: screenshots/screenshot.png

Change:

Choose File

No file chosen

Upload JPEG or PNG images up to 2000x2000 pixels.

Save

Screenshot details

Created	now
Uploaded by	testuser
Language	English

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

Delete

2.12 Перевірки і виправлення

2.12.1 Нетипові автоматичні виправлення

Ви можете також реалізувати ваші власні виправлення на додачу до стандартних і включити їх до `AUTOFIX_LIST`.

Автоматичні виправлення є потужним засобом, але вони можуть пошкодити дані. Будьте обережні із написанням автоматичних виправлень.

Наприклад, наведене нижче автоматичне виправлення має замінити усі випадки використання рядка `foo` у перекладі на `bar`:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

from django.utils.translation import gettext_lazy as _

from weblate.trans.autofixes.base import AutoFix

class ReplaceFooWithBar(AutoFix):
    """Replace foo with bar."""

    name = _("Foobar")

    def fix_single_target(self, target, source, unit):
        if "foo" in target:
            return target.replace("foo", "bar"), True
        return target, False
```

Щоб встановити нетипові перевірки, вкажіть повний шлях до класу Python у `AUTOFIX_LIST`, див. *Немуніві перевірки якості, додатки та автоматичні виправлення*.

2.12.2 Налаштовування поведінки за допомогою прапорців

Ви можете скоригувати поведінку Weblate за допомогою прапорців. Зробити це можна на рівні початкових рядків (див. *Додаткові відомості щодо початкових рядків*) або на рівні *Налаштовування складників (Прапорці перекладу)*. Для деяких форматів файлів можна вказати прапорці безпосередньо у записі формату (див. *Підтримувані формати файлів*).

Прапорці слід відокремлювати комами, а параметри — двокрапками. Ви можете скористатися лапками для включення до рядка пробілів або спеціальних символів. Приклад:

```
placeholders:"special:value":"other value", regex:.*
```

Можна використовувати одинарні і подвійні лапки. Спеціальні символи слід екранувати символом зворотної похилої риски:

```
placeholders:"quoted \"string\"":'single \'quoted\''
```

Ось список прапорців, які приймаються у поточній версії:

rst-text

Вважати текст документом reStructuredText, впливає на *Переклад не змінено*.

dos-eol

Використовує позначки кінців рядків DOS замість позначок кінців рядків Unix (`\r\n` замість `\n`).

read-only

Рядок призначено лише для читання, його не слід редагувати на Weblate, див. *Рядки лише для читання*.

priority:N

Пріоритетність рядка. Рядки із високою пріоритетністю надаються для перекладу першими. Типовим значенням пріоритетності є 100. Чим вищою є пріоритетність, тим раніше рядок буде запропоновано до перекладу.

max-length:N

Обмежити максимальну довжину рядка N символами, див. *Максимальна довжина перекладу*.

xml-text

Вважати текст документом XML, впливає на *Синтаксис XML* і *Розмітка XML*.

font-family:НАЗВА

Визначає сімейство шрифтів для перевірок обробки, див. *Керування шрифтами*.

font-weight:ВАГА

Визначити вагу шрифту для перевірок обробки, див. *Керування шрифтами*.

font-size:РОЗМІР

Визначити розмір шрифту для перевірок обробки, див. *Керування шрифтами*.

font-spacing:ІНТЕРВАЛ

Визначити інтервал між літерами для перевірок обробки, див. *Керування шрифтами*.

icu-flags:ПРАПОРЦІ

Визначити прапорці для налаштування поведінки перевірки якості *MessageFormat ICU*.

icu-tag-prefix:ПРЕФІКС

Встановити обов'язковий префікс для тегів XML для перевірки якості *MessageFormat ICU*.

placeholders:НАЗВА:НАЗВА2:...

Рядки-замінники, які слід очікувати у перекладі, див. *Заповнювачі*.

replacements:ВІД:ДО:ВІД2:ДО2...

Заміни, які слід виконати при перевірці параметрів тексту у результаті (наприклад, у *Найбільший розмір перекладу* або *Максимальна довжина перекладу*). Типовим випадком використання є розгортання замінників для забезпечення відповідності тексту, навіть за використання довгих значень. Приклад: `replacements:%s:"John Doe"`.

variants:ДЖЕРЕЛО

Позначити цей рядок варіантом рядка з тим самим джерелом. Див. *variants*.

regex:ФОРМАЛЬНИЙ_ВИРАЗ

Формальний вираз для зіставлення файлів перекладу, див. *Регулярний вираз*.

forbidden

Позначає заборонений переклад у глосарії, див. *Заборонені переклади*.

strict-same

Наказати перевірці «Незмінений переклад» уникати використання «чорного» списку вбудованих слів, див. *Переклад не змінено*.

check-glossary

Увімкнути перевірку якості *Не слідує глосарію*.

angularjs-format

Пропустити перевірку якості *Рядок інтерполяції AngularJS*.

c-format

Увімкнути перевірку якості *Формат C*.

c-sharp-format

Увімкнути перевірку якості *Формат C#*.

es-format

Увімкнути перевірку якості *Літерали шаблонів ECMAScript*.

i18next-interpolation

Увімкнути перевірку якості *i18next інтерполяція*.

icu-message-format

Увімкнути перевірку якості *MessageFormat ICU*.

java-printf-format

Увімкнути перевірку якості *Формат Java*.

java-format

Увімкнути перевірку якості *Формат Java Message*.

javascript-format

Увімкнути перевірку якості *Формат JavaScript*.

lua-format

Увімкнути перевірку якості *Формат Lua*.

object-pascal-format

Увімкнути перевірку якості *Форматування Object Pascal*.

percent-placeholders

Увімкнути перевірку якості *Відсоток заповнювачів*.

perl-format

Увімкнути перевірку якості *Формат Perl*.

php-format

Увімкнути перевірку якості *Формат PHP*.

python-brace-format

Увімкнути перевірку якості *Формат дужок Python*.

python-format

Увімкнути перевірку якості *Формат Python*.

qt-format

Увімкнути перевірку якості *Формат Qt*.

qt-plural-format

Увімкнути перевірку якості *Формат Qt у множині*.

ruby-format

Увімкнути перевірку якості *Формат Ruby*.

scheme-format

Увімкнути перевірку якості *Формат Scheme*.

vue-format

Увімкнути перевірку якості *Форматування I18n Vue*.

md-text

Вважати текст документом Markdown. Увімкнути перевірки якості *Посилання Markdown*, *Markdown посилання* та *Синтаксис Markdown*.

case-insensitive

Коригує поведінку перевірок так, щоб при їхньому виконанні не було враховано регістр літер. У поточній версії стосується лише перевірки якості *Заповнювачі*.

safe-html

Увімкнути перевірку якості *Небезпечний HTML-код*.

url

Рядок має складатися лише з URL-адреси. Увімкнути перевірку якості *URL*.

ignore-all-checks

Ігнорувати усі перевірки якості.

ignore-bbcode

Пропустити перевірку якості *Розмітка BBCode*.

ignore-duplicate

Пропустити перевірку якості *Послідовне повторення слів*.

ignore-check-glossary

Пропустити перевірку якості *Не слідує глосарію*.

ignore-double-space

Пропустити перевірку якості *Подвійний пробіл*.

ignore-angularjs-format

Пропустити перевірку якості *Рядок інтерполяції AngularJS*.

ignore-c-format

Пропустити перевірку якості *Формат C*.

ignore-c-sharp-format

Пропустити перевірку якості *Формат C#*.

ignore-es-format

Пропустити перевірку якості *Літерали шаблонів ECMAScript*.

ignore-i18next-interpolation

Пропустити перевірку якості *i18next інтерполяція*.

ignore-icu-message-format

Пропустити перевірку якості *MessageFormat ICU*.

ignore-java-format

Пропустити перевірку якості *Формат Java Message*.

ignore-java-printf-format

Пропустити перевірку якості *Формат Java*.

ignore-javascript-format

Пропустити перевірку якості *Формат JavaScript*.

ignore-lua-format

Пропустити перевірку якості *Формат Lua*.

ignore-object-pascal-format

Пропустити перевірку якості *Форматування Object Pascal*.

ignore-percent-placeholders

Пропустити перевірку якості *Відсоток заповнювачів*.

ignore-perl-format

Пропустити перевірку якості *Формат Perl*.

ignore-php-format

Пропустити перевірку якості *Формат PHP*.

ignore-python-brace-format

Пропустити перевірку якості *Формат дужок Python*.

ignore-python-format

Пропустити перевірку якості *Формат Python*.

ignore-qt-format

Пропустити перевірку якості *Формат Qt*.

ignore-qt-plural-format

Пропустити перевірку якості *Формат Qt у множині*.

ignore-ruby-format

Пропустити перевірку якості *Формат Ruby*.

ignore-scheme-format

Пропустити перевірку якості *Формат Scheme*.

ignore-vue-format

Пропустити перевірку якості *Форматування HTML Vue*.

ignore-translated

Пропустити перевірку якості *Вже перекладено*.

ignore-inconsistent

Пропустити перевірку якості *Неузгодженість*.

ignore-kashida

Пропустити перевірку якості *Використано літеру кашида*.

ignore-md-link

Пропустити перевірку якості *Посилання Markdown*.

ignore-md-reflink

Пропустити перевірку якості *Markdown посилання*.

ignore-md-syntax

Пропустити перевірку якості *Синтаксис Markdown*.

ignore-max-length

Пропустити перевірку якості *Максимальна довжина перекладу*.

ignore-max-size

Пропустити перевірку якості *Найбільший розмір перекладу*.

ignore-escaped-newline

Пропустити перевірку якості *Не збіглися \n*.

ignore-end-colon

Пропустити перевірку якості *Невідповідність двокрапки*.

ignore-end-ellipsis

Пропустити перевірку якості *Невідповідність багатокрапок*.

ignore-end-exclamation

Пропустити перевірку якості *Невідповідність знаків оклику*.

ignore-end-stop

Пропустити перевірку якості *Невідповідність кінцевої крапки*.

ignore-end-question

Пропустити перевірку якості *Невідповідність знаків питання*.

ignore-end-semicolon

Пропустити перевірку якості *Невідповідність крапки з комою*.

ignore-newline-count

Пропустити перевірку якості *Розбіжність розривів рядків*.

ignore-plurals

Пропустити перевірку якості *Бракує множини*.

ignore-placeholders

Пропустити перевірку якості *Заповнювачі*.

ignore-punctuation-spacing

Пропустити перевірку якості *Щільність знаків*.

ignore-regex

Пропустити перевірку якості *Регулярний вираз*.

ignore-same-plurals

Пропустити перевірку якості *Однакова множина*.

ignore-begin-newline

Пропустити перевірку якості *Перехід у новий рядок*.

ignore-begin-space

Пропустити перевірку якості *Початкові пропуски*.

ignore-end-newline

Пропустити перевірку якості *Кінцевий знак нового рядка*.

ignore-end-space

Пропустити перевірку якості *Кінцеві пропуски*.

ignore-same

Пропустити перевірку якості *Переклад не змінено*.

ignore-safe-html

Пропустити перевірку якості *Небезпечний HTML-код*.

ignore-url

Пропустити перевірку якості *URL*.

ignore-xml-tags

Пропустити перевірку якості *Розмітка XML*.

ignore-xml-invalid

Пропустити перевірку якості *Синтаксис XML*.

ignore-zero-width-space

Пропустити перевірку якості *Простір нульової ширини*.

ignore-ellipsis

Пропустити перевірку якості *Трикрапка*.

ignore-icu-message-format-syntax

Пропустити перевірку якості *Синтаксис ICU MessageFormat*.

ignore-long-untranslated

Пропустити перевірку якості *Тривалий час не перекладено*.

ignore-multiple-failures

Пропустити перевірку якості *Кратні невдалі перевірки*.

ignore-unnamed-format

Пропустити перевірку якості *Декілька змінних без назв*.

ignore-optional-plural

Пропустити перевірку якості *Немає форм множини*.

Примітка: Загалом, правило для будь-якої перевірки має назву `ignore-*` і використовує ідентифікатор перевірки, отже ви можете скористатися цим навіть для ваших власних перевірок.

Ці прапорці можна використовувати у параметрах *Налаштування складників*, параметрах окремих початкових рядків та у самому файлі перекладу (наприклад, у GNU gettext).

2.12.3 Примусові перевірки

Нове в версії 3.11.

Ви можете налаштувати список перевірок, які не можна ігнорувати, встановленням значення *Примусові перевірки у Налаштування складників*. Усі перевірки зі списку не може бути проігноровано у інтерфейсі користувача, а будь-який рядок, який не пройде перевірку, буде позначено міткою *Потребує редагування* (див. *Стани перекладу*).

Примітка: Вмикання примусової перевірки не робить її доступною автоматично. Перевірку можна увімкнути додаванням відповідного прапорця до рядка або прапорців складника.

Дивись також:

Додаткові відомості щодо початкових рядків, Прапорці перекладу

2.12.4 Керування шрифтами


Нове в версії 3.7.

Підказка: Шрифти, які вивантажено на Weblate, використовуються лише для реалізації перевірки *Найбільший розмір перекладу*, вони не впливають на інтерфейс користувача Weblate.

Перевірка *Найбільший розмір перекладу*, яка використовується для обчислення розмірностей обробленого тексту, потребує вибору шрифту, який має бути завантажено до Weblate і вибрано за допомогою прапорця перекладу (див. *Налаштування поведінки за допомогою прапорців*).

Засіб керування шрифтами Weblate у розділі *Шрифти* меню *Управління* вашого проєкту з перекладу надає інтерфейс для вивантаження шрифтів та керування ними. Можна вивантажувати шрифти TrueType або OpenType, налаштовувати групи шрифтів і використовувати їх у перевірці.

Групи шрифтів надають вам змогу визначити різні шрифти для різних мов, що типово знадобиться для мов, які записують нелатинськими символами:





 Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

WebOrg / Font groups / default-font

Font group

Name	default-font		
Default font	Source Sans 3 Bold		
Japanese	language override	Droid Sans Fallback Regular	Remove
Korean	language override	Droid Sans Fallback Regular	Remove
Delete			

Add language override

Language

----- ▾

Font

----- ▾

Save

Edit font group

Font group name

default-font

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

Source Sans 3 Bold ▾

Default font is used unless per language override matches.

Save

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Групи шрифтів визначаються назвою, яка не може містити пробілів або спеціальних символів. Так зроблено, щоб групи можна було без проблем використовувати у визначеннях перевірок:

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Fonts

Font groups
 Fonts

Group name	Default font	Language overrides	
default-font	Source Sans 3 Bold	Japanese: Droid Sans Fallback Regular Korean: Droid Sans Fallback Regular	Edit

Add font group

Font group name

 Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

 Default font is used unless per language override matches.

Save

Powered by Weblate 4.16
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

Гарнітура шрифту та стиль автоматично визначаються після вивантаження шрифтів:

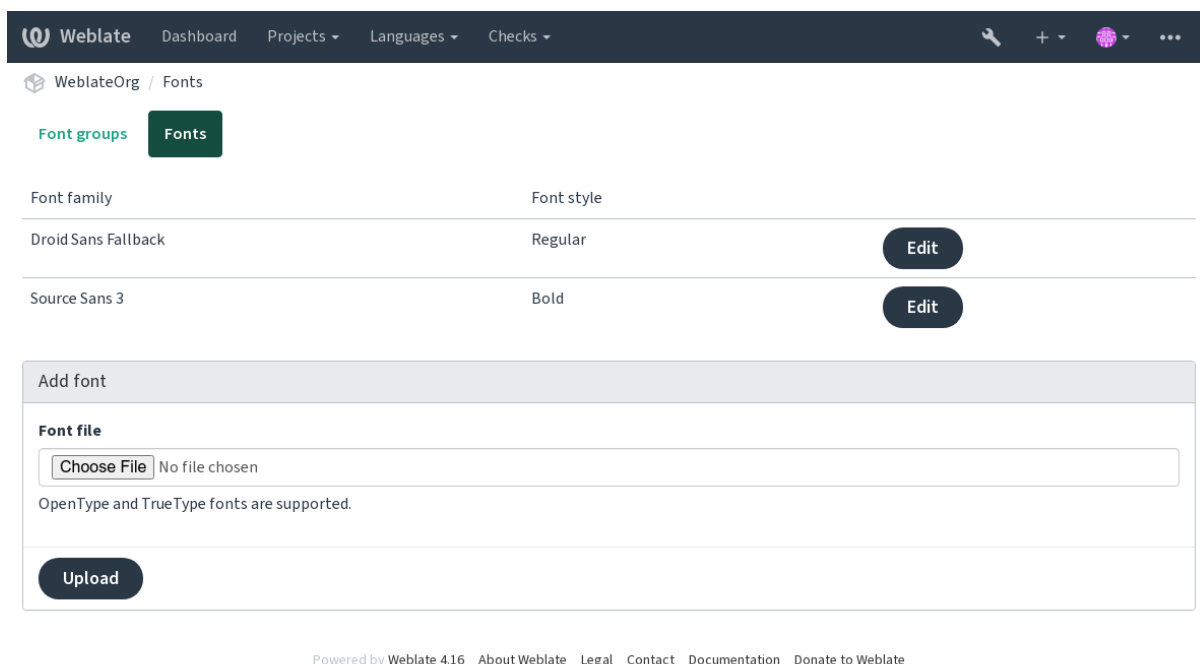
Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Fonts / Droid Sans Fallback Regular

Font	
Font family	Droid Sans Fallback
Font style	Regular
File size	3939852
Created	now
Uploaded by	testuser
Used in groups	
Delete	

Powered by Weblate 4.16
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

Ви можете завантажувати до Weblate декілька шрифтів:



Щоб скористатися шрифтами для перевірки довжини рядка, передайте відповідні прапорці (див. [Налаштування поведінки за допомогою прапорців](#)). Вам, ймовірно, потрібні такі:

max-size:500

Визначає максимальну ширину у пікселях.

font-family:ubuntu

Визначає групу шрифтів для використання її визначенням ідентифікатора.

font-size:22

Визначає розмір шрифту у пікселях.

2.12.5 Написання власних перевірок

Широкий діапазон перевірок якості вбудовано до програми (див. [Перевірки якості](#)), хоча вони можуть не покривати усі випадки того, що ви хочете перевірити. Список виконуваних перевірок можна скоригувати за допомогою параметра `CHECK_LIST`. Також ви можете додавати нетипові перевірки.

1. Підклас `weblate.checks.Check`
2. Встановити декілька атрибутів.
3. Реалізуйте метод `check` (якщо ви хочете мати справу із формами множини у вашому коді) або метод `check_single` (який виконує це завдання за вас).

Приклади:

Щоб встановити нетипові перевірки, вкажіть повний шлях до класу Python у `CHECK_LIST`, див. [Немунові перевірки якості, додатки та автоматичні виправлення](#).

Перевірка того, чи не містить текст перекладу «foo»

Це доволі проста перевірка, яка просто перевіряє, чи не пропущено у перекладі рядок «foo».

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""Simple quality check example."""

from django.utils.translation import gettext_lazy as _
from weblate.checks.base import TargetCheck

class FooCheck(TargetCheck):
    # Used as identifier for check, should be unique
    # Has to be shorter than 50 characters
    check_id = "foo"

    # Short name used to display failing check
    name = _("Foo check")

    # Description for failing check
    description = _("Your translation is foo")

    # Real check code
    def check_single(self, source, target, unit):
        return "foo" in target
```

Перевірка того, чи є форми множини у перекладі чеською різними

Перевіримо з використанням даних щодо мови, чи дві форми множини у чеському перекладі є різними.

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""Quality check example for Czech plurals."""

from django.utils.translation import gettext_lazy as _
from weblate.checks.base import TargetCheck

class PluralCzechCheck(TargetCheck):
    # Used as identifier for check, should be unique
    # Has to be shorter than 50 characters
    check_id = "foo"

    # Short name used to display failing check
    name = _("Foo check")

    # Description for failing check
    description = _("Your translation is foo")

    # Real check code
    def check_target_unit(self, sources, targets, unit):
        if self.is_language(unit, ("cs",)):
            return targets[1] == targets[2]
        return False
```

(continues on next page)


(continued from previous page)


```
def check_single(self, source, target, unit):  
    """We don't check target strings here."""  
    return False
```

2.13 Налаштовування автоматичних пропозицій

Змінено в версії 4.13: До Weblate 4.13 служби слід було налаштовувати у *налаштуваннях*.

Передбачено вбудовану підтримку декількох служб комп'ютерного перекладу. Цю підтримку може бути увімкнено адміністратором на рівні сайта або параметрів проєкту:


Dashboard
Projects ▾
Languages ▾
Checks ▾


WeblateOrg / Automatic suggestions

Configured automatic suggestion services ⓘ

There are no services currently installed.

Available automatic suggestion services ⓘ

AWS ⓘ	Install
Amagama ⓘ	Install
Apertium APy ⓘ	Install
Baidu ⓘ	Install
DeepL ⓘ	Install
Glosbe ⓘ	Install
Google Translate ⓘ	Install
Google Translate API v3 ⓘ	Install
IBM ⓘ	Install
LibreTranslate ⓘ	Install
Microsoft Terminology ⓘ	Install
Microsoft Translator ⓘ	Install
ModernMT ⓘ	Install
MyMemory ⓘ	Install
Netease Sight ⓘ	Install
SAP Translation Hub ⓘ	Install
Weblate ⓘ	Install
Weblate Translation Memory ⓘ	Install
Yandex ⓘ	Install
Youdao Zhiyun ⓘ	Install
tmserver ⓘ	Install

Some services will ask for additional configuration during installation.

Примітка: Переклади регламентуються відповідними умовами використання, тому переконайтеся, що ви зможете використовувати їх у бажаний для вас спосіб.

Служби перекладу з початкової мови виконують переклад за налаштуваннями у [Налаштування складників](#), див. [Джерельна мова](#).

Дивись також:

[Автоматичні пропозиції](#)

2.13.1 Amagama

Ідентифікатор служби

amagama

Налаштування

Для цієї служби налаштувань не передбачено.

Особливий екземпляр [tmserver](#), яким керують розробники Virtaal.

Дивись також:

[Installing amaGama, Amagama, пам'ять перекладів amaGama](#)

2.13.2 Apertium APy

Ідентифікатор служби

apertium-apy

Налаштування

url	Адреса API
-----	------------

Платформа перекладу вільного програмного забезпечення, яка забезпечує переклади обмеженим набором мов.

Рекомендованим способом використання Apertium є запуск вашого власного сервера Apertium-APy.

Дивись також:

[Сайт Apertium, Документація з Apertium APy](#)

2.13.3 AWS

Нове в версії 3.1.

Ідентифікатор служби

aws

Налаштування

key	Ідентифікатор ключа доступу
secret	Закритий ключ API
region	Назва регіону

Amazon Translate — служба перекладу на основі нейронних машин для перекладу тексту англійською мовою і з англійської мови для декількох підтримуваних мов.

Дивись також:

[Документація до Amazon Translate](#)

2.13.4 Baidu

Нове в версії 3.2.

Ідентифікатор служби

baidu

Налаштування

key	Ідентифікатор клієнта
secret	Пароль клієнта

Служба комп'ютерного перекладу, яка надається Baidu.

Ця служба використовує програмний інтерфейс — вам слід отримати у Baidu ідентифікатор і ключ до програмного інтерфейсу, щоб скористатися нею.

Дивись також:

[Програмний інтерфейс Baidu Translate](#)

2.13.5 DeepL

Нове в версії 2.20.

Ідентифікатор служби

deepl

Налаштування

url	Адреса API
key	Ключ до API

DeepL є платною службою, яка надає якісні комп'ютерні переклади декількома мовами. Вам слід придбати передплату на *DeepL API* або можете скористатися застарілим тарифним планом *DeepL Pro (classic)*.

Адреса програмного інтерфейсу API, яку слід використовувати для служби DeepL. На час написання цього підручника використовувався програмний інтерфейсу версії 1, а також безкоштовна та оплачувана версія 2 програмного інтерфейсу.

`https://api.deepl.com/v2/` (типова у Weblate)

Призначено для використання програмного інтерфейсу у межах оплачуваного плану. Передплата є заснованою на використанні.

`https://api-free.deepl.com/v2/`

Призначено для використання програмного інтерфейсу у межах безкоштовного плану. Передплата є заснованою на використанні.

`https://api.deepl.com/v1/`

Призначено для інструментів комп'ютерного перекладу. Придатна до використання із передплатою для окремих користувачів.

Раніше DeepL вважала Weblate засобом комп'ютеризованого перекладу, тому слід було використовувати версію 1 програмного інтерфейсу, але тепер слід використовувати версію 2. Через це, типовим є використання версії 2. Ви можете змінити версію на 1, якщо у вас є передплата для системи комп'ютеризованого перекладу, і ви хочете, щоб Weblate скористався нею.

Найпростішим способом визначення версії є відкриття адреси, подібної до вказаної нижче, у вашому браузері:

https://api.deepl.com/v2/translate?text=Hello&target_lang=FR&auth_key=XXX

Замініть XXX на ваше значення ключа розпізнавання (auth_key). Якщо ви отримуєте об'єкт JSON, який містить «Bonjour», адресу вказано правильно. Якщо отримано щось інше, спробуйте інші три варіанти.

У Weblate передбачено підтримку формальності DeepL, програма вибере відповідник на основі мови (наприклад, `de@formal` і `de@informal`).

Дивись також:

[Сайт DeepL](#), [Ціни DeepL](#), [Документація з програмного інтерфейсу DeepL](#)

2.13.6 Glosbe

Ідентифікатор служби

`glosbe`

Налаштування

Для цієї служби налаштувань не передбачено.

Вільний словник та пам'ять перекладів для майже будь-якої із актуальних мов.

Доступ до програмного інтерфейсу є безплатним, але використання перекладів підлягає умовам ліцензування використаного джерела початкових даних. Кількість викликів з однієї адреси IP обмежено протягом встановленого періоду часу для запобігання зловмисному використанню.

Дивись також:

[Сайт Glosbe](#)

2.13.7 Перекладач Google

Ідентифікатор служби

`google-translate`

Налаштування

<code>key</code>	Ключ до API
------------------	-------------

Служба комп'ютерного перекладу, яка надається Google.

Ця служба використовує програмний інтерфейс Перекладача Google — вам слід отримати ключ до програмного інтерфейсу і увімкнути облік у консолі програмного інтерфейсу Google.

Дивись також:

Документація з Перекладача Google <<https://cloud.google.com/translate/docs>>`_

2.13.8 Програмний інтерфейс Google Перекладач v3

Ідентифікатор служби

google-translate-api-v3

Налаштування

credentials	Дані службового облікового запису Google Перекладач
project	Проект Google Перекладач
location	Місце Google Перекладач

Служба комп'ютерного перекладу, яка надається хмарними службами Google.

Дивись також:

Документація до Google Перекладача, Автентифікація у хмарних службах за допомогою клієнтських бібліотек, Створення проєкту Google Перекладач, Місця рушія програм Google Cloud

2.13.9 LibreTranslate

Нове в версії 4.7.1.

Ідентифікатор служби

libretranslate

Налаштування

url	Адреса API
key	Ключ до API

LibreTranslate — безкоштовна служба комп'ютерного перекладу із відкритим кодом. Для доступу до відкритого екземпляра потрібен ключ до програмного інтерфейсу. Втім, LibreTranslate можна розгорнути на власному сервері. Існує також декілька дзеркал, програмним інтерфейсом яких можна користуватися безкоштовно.

<https://libretranslate.com/> (офіційний загальнодоступний екземпляр)

Потребує ключа до програмного інтерфейсу для використання поза межами сайту.

Дивись також:

Сайт LibreTranslate, репозиторій LibreTranslate, дзеркала LibreTranslate

2.13.10 Microsoft Terminology

Нове в версії 2.19.

Ідентифікатор служби

microsoft-terminology

Налаштування

Для цієї служби налаштувань не передбачено.

За допомогою програмного інтерфейсу служби термінології Microsoft ви можете програмно отримувати доступ до термінології, визначень та рядків інтерфейсу користувача, які доступні на порталі мов за допомогою вебслужби.

Дивись також:

Програмний інтерфейс служби термінології Microsoft

2.13.11 Microsoft Translator

Нове в версії 2.10.

Ідентифікатор служби

microsoft-translator

Налаштування

key	Ключ до API
base	<p>Доступні варіанти:</p> <p>api.cognitive.microsofttranslator.com – загальносвітовий (всесвітній)</p> <p>api-apc.cognitive.microsofttranslator.com – Азія та Тихоокеанський регіон</p> <p>api-eur.cognitive.microsofttranslator.com – Європа</p> <p>api-nam.cognitive.microsofttranslator.com – Північна Америка</p> <p>api.translator.azure.cn – Китай</p> <p>api.cognitive.microsofttranslator.us – хмара Azure уряду США</p>
endp	<p>URL-адреса служби розпізнання</p> <p>Регіональний або багатосервісний можна вказати за допомогою поля «Регіон» нижче.</p> <p>Доступні варіанти:</p> <p>api.cognitive.microsoft.com – всесвітній</p> <p>api.cognitive.azure.cn – Китай</p> <p>api.cognitive.microsoft.us – хмара Azure US Government</p>
regi	Регіон служби розпізнання

Служба комп'ютерного перекладу, яка надається Microsoft на порталі Azure як одна з когнітивних служб.

У Weblate реалізовано програмний інтерфейс перекладача версії 3.

Текстовий програмний інтерфейс перекладача версії 2

Ключ, яким ви користуєтеся для доступу до програмного інтерфейсу перекладача версії 2, може бути використано і для програмного інтерфейсу версії 3.

Текстовий програмний інтерфейс перекладача версії 3

Вам слід зареєструватися на порталі Azure і скористатися отриманим там ключем. З новими ключами Azure вам також слід встановити `region` у значення локалі вашої служби.

Підказка: Для Azure China, будь ласка, скористайтеся вашою кінцевою точкою з portalu Azure.

Дивись також:

Cognitive Services — програмний інтерфейс перекладу текстів, Портал Microsoft Azure, Base URLs, «Розпізнавання для ресурсу із багатьма службами», Розділ «Розпізнавання за допомогою жетона доступу»

2.13.12 ModernMT

Нове в версії 4.2.

Ідентифікатор служби

`modernmt`

Налаштування

<code>url</code>	Адреса API
<code>key</code>	Ключ до API

Дивись також:

Програмний інтерфейс ModernMT,

2.13.13 MyMemory

Ідентифікатор служби

`mymemory`

Налаштування

<code>email</code>	Ел. пошта для зв'язку
<code>username</code>	Ім'я користувача
<code>key</code>	Ключ до API

Величезна пам'ять перекладів із комп'ютерним перекладом.

Вільний анонімний доступ у поточній версії обмежено 100 запитами за день або 1000 запитами за день, якщо ви надасте адресу електронної пошти у `email`. Ви також можете попросити власників служби збільшити кількість запитів.

Дивись також:

Сайт MyMemory

2.13.14 Netease Sight

Нове в версії 3.3.

Ідентифікатор служби

netease-sight

Налаштування

key	Ідентифікатор клієнта
secret	Пароль клієнта

Служба комп'ютерного перекладу, яка надається NetEase.

Ця служба використовує програмний інтерфейс — вам слід отримати ключ і пароль від NetEase.

Дивись також:

[Платформа перекладу NetEase Sight](#)

2.13.15 Концентратор перекладів SAP

Ідентифікатор служби

sap-translation-hub

Налаштування

url	Адреса API	
key	Ключ до API	
usern	Ім'я користувача SAP	
passw	Пароль до SAP	
enabl	Увімкнути комп'ютерний переклад	
domai	Домен перекладу	Ідентифікатор домену перекладу. Приклад: ВС. Якщо ви не вкажете домен, метод шукатиме переклади в усіх доступних доменах.

Служба комп'ютерного перекладу, яка надається SAP.

Вам знадобиться обліковий запис SAP (і увімкнений концентратор перекладів SAP на «хмарній» платформі SAP), щоб скористатися цією службою.

Ви також можете налаштувати, чи слід також використовувати служби комп'ютерного перекладу на додачу до бази даних термінів.

Примітка: Щоб отримати доступ до програмного інтерфейсу Sandbox, вам слід встановити url і key.

Щоб отримати доступ до програмного інтерфейсу продукту, вам слід встановити значення url, username і password.

Дивись також:

[Програмний інтерфейс концентратора перекладів SAP](#), [Побудова базової адреси концентратора перекладів SAP](#)

2.13.16 tmserver

Ідентифікатор служби

tmserver

Налаштування

url	Адреса API
-----	------------

Ви можете запустити ваш власний сервер пам'яті перекладів скориставшись сервером із Translate-toolkit і повідомивши Weblate, що слід обмінюватися даними з цим сервером. Ви також можете скористатися сервером amaGama, який є удосконаленою версією tmserver.

1. Спочатку вам варто імпортувати якісь дані до пам'яті перекладів:

```
build_tmdb -d /var/lib/tm/db -s en -t cs locale/cs/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t de locale/de/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t fr locale/fr/LC_MESSAGES/django.po
```

2. Запустіть tmserver для очікування на ваші запити:

```
tmserver -d /var/lib/tm/db
```

3. Налаштуйте Weblate на обмін даними зі службою, типовою адресою є `http://localhost:8888/tmserver/`.

Дивись також:

[tmserver](#) [Installing amaGama](#), [Amagama](#), [пам'ять перекладів Amagama](#)

2.13.17 Перекладач IBM Watson

Ідентифікатор служби

ibm

Налаштування

url	Адреса API
key	Ключ до API

Перекладач IBM Watson перекладає текст з однієї мови іншою. Служба пропонує декілька специфічних для домену моделей.

Дивись також:

[Перекладач Watson](#), [Документація до хмарного програмного інтерфейсу IBM](#)

2.13.18 Weblate

Ідентифікатор служби

weblate

Налаштування

Для цієї служби налаштувань не передбачено.

Служба комп'ютерного перекладу Weblate може надавати переклади рядків, які вже перекладено у базах даних Weblate. Програма шукатиме точні відповідники у наявних рядках.

2.13.19 Пам'ять перекладів Weblate

Нове в версії 2.20.

Ідентифікатор служби

`weblate-translation-memory`

Налаштування

Для цієї служби налаштувань не передбачено.

Скористайтесь *пам'яттю перекладів* як службою комп'ютерного перекладу. У цей спосіб можна перекласти усі рядки, які раніше було перекладено (або вивантажено до пам'яті перекладів).

2.13.20 Yandex

Ідентифікатор служби

`yandex`

Налаштування

<code>key</code>	Ключ до API
------------------	-------------

Служба комп'ютерного перекладу, яка надається Yandex.

Ця служба використовує програмний інтерфейс перекладу — вам слід отримати ключ до програмного інтерфейсу від Yandex.

Дивись також:

Програмний інтерфейс перекладу Yandex, [Yandex.Translate](#)

2.13.21 Youdao Zhiyun

Нове в версії 3.2.

Ідентифікатор служби

`youdao-zhiyun`

Налаштування

<code>key</code>	Ідентифікатор клієнта
<code>secret</code>	Пароль клієнта

Служба комп'ютерного перекладу, яка надається Youdao.

Ця служба використовує програмний інтерфейс — вам слід отримати у Youdao ідентифікатор і ключ до програмного інтерфейсу, щоб скористатися нею.

Дивись також:

Служба перекладу природними мовами Youdao Zhiyun

2.13.22 Нетиповий комп'ютерний переклад

Ви також можете реалізувати власні служби комп'ютерного перекладу за допомогою декількох рядків коду мовою Python. У цьому прикладі реалізовано комп'ютерний переклад для фіксованого списку мов за допомогою модуля `dictionary` Python:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""Machine translation example."""

import dictionary

from weblate.machinery.base import MachineTranslation

class SampleTranslation(MachineTranslation):
    """Sample machine translation interface."""

    name = "Sample"

    def download_languages(self):
        """Return list of languages your machine translation supports."""
        return {"cs"}

    def download_translations(
        self,
        source,
        language,
        text: str,
        unit,
        user,
        search: bool,
        threshold: int = 75,
    ):
        """Return tuple with translations."""
        for t in dictionary.translate(text):
            yield {"text": t, "quality": 100, "service": self.name, "source": text}
```


Ви можете додати ваш власний клас до `WEBLATE_MACHINERY` і Weblate розпочне використовувати його.

2.14 Додатки

Нове в версії 2.19.

Додатки надають змогу налаштувати й автоматизувати процедуру перекладу. Адміністратори можуть додавати додатки та керувати ними за допомогою меню *Керувати* ↓ *Додатки* для кожного відповідного складника перекладу для адміністраторів.

Підказка: Ви також можете налаштувати додатки за допомогою настанов з розділів [API](#), `DEFAULT_ADDONS` або `install_addon`.



[Weblate](#)
[Dashboard](#)
[Projects](#)
[Languages](#)
[Checks](#)

[WeblateOrg](#) / [Language names](#) / [Add-ons](#)

Installed add-ons


There are no add-ons currently installed.

Available add-ons


Automatic translation

Automatically translates strings using machine translation or other components.


Install


Add missing languages

Ensures a consistent set of languages is used for all components within a project.

project wide


Install


Component discovery

Automatically adds or removes project components based on file changes in the version control system.


repository wide

Install


Bulk edit


Bulkedit flags, labels, or states of strings.

Install


Statistics generator


Generates a file containing detailed info about the translation status.

Install


Prefill translation with source


Fills in translation strings with source string.

Install


Pseudolocale generation


Generates a translation by adding prefix and suffix to source strings automatically.

Install


Contributors in comment


Updates the comment part of the PO file header to include contributor names and years of contributions.

Install


Customize gettext output


Allows customization of gettext output behavior, for example line wrapping.

Install


Generate MO files


Automatically generates a MO file for every changed PO file.

Install


Update PO files to match POT (msgmerge)

Updates all PO files (as configured by "File mask") to match the POT file (as configured by "Template for new translations") using msgmerge.


Install


Squash Git commits

Squash Git commits prior to pushing changes.

repository wide


Install


Stale comment removal

Set a timeframe for removal of comments.

project wide

Install


Stale suggestion removal

Set a timeframe for removal of suggestions.

project wide

Install

Some add-ons will ask for additional configuration during installation.

2.14.1 Вбудовані додатки

Автоматичний переклад

Нове в версії 3.9.

Ідентифікатор додатка

`weblate.autotranslate.autotranslate`

Налаштування

<code>mode</code>	Режим	Доступні варіанти: авто- мати- чного пере- кладу
<code>filter</code>	Фільтр пошуку	Будь ласка, зауважте, що переклад усіх рядків призведе до відкидання усіх наявних перекладів. Доступні варіанти: <code>all</code> — усі рядки <code>nottranslated</code> — неперекладені рядки <code>todo</code> — незавершені рядки <code>fuzzy</code> — рядки, які позначено для редагування <code>check:inconsistent</code> — не пройдено перевірку: різні переклади одного рядка
<code>auto</code>	Джерело автоматизованих перекладів	Доступні варіанти: <code>others</code> — інші складники перекладу <code>mt</code> — комп'ютерний переклад
<code>comp</code>	Складники	Введіть опис складника для використання як джерела, залиште порожнім, щоб використовувати всі складники в поточному проєкті.
<code>engine</code>	Рушій комп'ютерного перекладу	
<code>threshold</code>	Поріг оцінки	

Перемикачі

оновлення складників, щоденно

Автоматично перекладає рядки за допомогою машинного перекладу або інших складників.

Причини вмикання:

- У складнику з'явилися нові рядки.
- Автоматичне щомісячне вмикання для всіх складників. Можна налаштувати за допомогою `BACKGROUND_TASKS`.

Дивись також:

Автоматичний переклад, Підтримання синхронізації перекладів у різних складниках

CDN локалізації JavaScript

Нове в версії 4.2.

Ідентифікатор додатка

`weblate.cdn.cdnjs`

Налаштування

<code>thresh</code>	Порогове значення перекладу	Порогове значення для включення перекладів.
<code>css_sel</code>	Вибір CSS	Засіб вибору CSS для виявлення придатних до локалізації елементів.
<code>cookie_name</code>	Назва куки мови	Назва куки, яка зберігатиме налаштування мови.
<code>files</code>	Видобути рядки з файлів HTML	Список назв файлів у поточному репозиторії або віддалених адрес, які слід обробити для отримання придатних до перекладу рядків.

Перемикачі

щоденна обробка, обробка репозиторію після внеску, обробка репозиторію після оновлення

Оприлюднює переклади у мережах надання даних для використання у локалізації JavaScript або HTML.

Можна скористатися для локалізації статичних сторінок HTML або завантаження локалізації у коді JavaScript.

Створює унікальну адресу для вашого складника, яку ви можете включати до сторінок HTML для їхньої локалізації. Див. `weblate-cdn`, щоб дізнатися більше.

Дивись також:

`cdn-addon-config`, `weblate-cdn`, `cdn-addon-extract`, `cdn-addon-html`

Вилучити порожні рядки

Нове в версії 4.4.

Ідентифікатор додатка

`weblate.cleanup.blank`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

обробка репозиторію після внеску, обробка репозиторію після оновлення

Вилучає рядки без перекладу з файлів перекладу.

Використовуйте це, якщо не хочете, щоб порожні рядки з'являлись у файлах перекладу (наприклад, коли у бібліотеці локалізації їх показано як пропущені рядки, а не повернено до початкового рядка).

Дивись також:

Чи оновлює Weblate файли перекладів між перекладами?

Очищення файлів перекладу

Ідентифікатор додатка

`weblate.cleanup.generic`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

обробка репозиторію до оновлення, обробка репозиторію після оновлення

Оновіть всі файли перекладу, щоб вони відповідали одномовному основному файлу. Для більшості форматів файлів це означає видалення застарілих перекладацьких ключів, які вже відсутні у основному файлі.

Дивись також:

Чи оновлює Weblate файли перекладів між перекладами?

Додавання нових мов

Ідентифікатор додатка

`weblate.consistency.languages`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

щоденна обробка, обробка репозиторію після додавання

Забезпечує використання однорідного набору мов для всіх складників проекту.

Пошук пропущених мов відбувається кожні 24 години і при додаванні нових мов у Weblate.

На відміну від інших, цей додаток стосується усього проекту.

Підказка: Автоматичний переклад нових доданих рядків за допомогою [Автоматичний переклад](#).

Виявлення складників

Ідентифікатор додатка

`weblate.discovery.discovery`

Налаштування

<code>match</code>	Регулярний вираз для зіставлення файлів перекладу	
<code>file_for_name</code>	Формат файлу	
<code>base_file_name</code>	Налаштувати назву складника	
<code>base_file</code>	Визначити назву основного файлу для одномовного перекладу	Лишіть порожнім для двомовних файлів перекладу.
<code>new_base_file</code>	Визначити основний файл для нового перекладу	Ім'я файлу, використане для створення нових перекладів. Для gettext виберіть pot-файл.
<code>intermediate_file</code>	Проміжний мовний файл	Назва проміжного файла перекладу. Здебільшого, це файл перекладу, який надається розробниками і використовується для створення самих рядків для перекладу.
<code>language</code>	Фільтр мов	Формальний вираз для фільтрування файлів перекладу під час сканування за маскою файлів.
<code>copy_additional</code>	Клонувати додатки з основного складника до нових	
<code>remove</code>	Видалити складники для файлів, яких не існує	
<code>confirm</code>	Я підтверджую, що всі вище наведені збіги виглядають правильними	

Перемикачі

обробка репозиторію після оновлення

Автоматично додає або вилучає складники до проекту на основі змін файла у системі керування версіями.

Вмикається під час кожного оновлення у системі керування версіями. Загалом, подібне до команди керування `import_project`. Один зі способів стежити за декількома складниками перекладу у одній системі керування версіями.

Відповідність встановлюватиметься за регулярними виразами, що надає можливість створювати складні налаштування, але для користування ними потрібні певні знання. Деякі приклади для типових випадків використання можна знайти у розділі довідки щодо додатків.

Після натискання кнопки *Зберегти* буде показано попередній перегляд відповідних складників, за допомогою якого ви зможете перевірити, чи відповідають налаштування вашим потребам:

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Language names / Add-ons / Component discovery

Configure add-on

Please review and confirm the matched components.

Component	Matched files
The following components would be created	
Djangojs (djangojs)	File mask: <code>weblate/locale/*/LC_MESSAGES/djangojs.po</code> <code>weblate/locale/cs/LC_MESSAGES/djangojs.po</code> (cs) <code>weblate/locale/hu/LC_MESSAGES/djangojs.po</code> (hu) <code>weblate/locale/he/LC_MESSAGES/djangojs.po</code> (he)
Django (django)	File mask: <code>weblate/locale/*/LC_MESSAGES/django.po</code> <code>weblate/locale/hu/LC_MESSAGES/django.po</code> (hu) <code>weblate/locale/cs/LC_MESSAGES/django.po</code> (cs) <code>weblate/locale/he/LC_MESSAGES/django.po</code> (he)

☐ I confirm the above matches look correct

Regular expression to match translation files against

`weblate/locale/(?P<language>[^\s]*)/LC_MESSAGES/(?P<component>[^\s]*)\.po`

File format

gettext PO file

Customize the component name

`{{ component|title }}`

Define the monolingual base filename

Leave empty for bilingual translation files.

Define the base file for new translations

`weblate/locale/{{ component }}.pot`

Filename of file used for creating new translations. For gettext choose .pot file.

Intermediate language file

Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.

Language filter

`^(cs|he|hu)$`

Regular expression to filter translation files against when scanning for file mask.

☒ Clone add-ons from the main component to the newly created ones

☐ Remove components for inexistent files

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description
<code>(?P<language>[^\s.]*)(?P<component>[^\s]*)\.po</code>	<code>cs/application.po</code> <code>cs/website.po</code> <code>de/application.po</code> <code>de/website.po</code>	One folder per language containing translation files for components.
<code>locale/(?P<language>[^\s.]*)/LC_MESSAGES/(?P<component>[^\s]*)\.po</code>	<code>locale/cs/LC_MESSAGES/application.po</code> <code>locale/cs/LC_MESSAGES/website.po</code> <code>locale/de/LC_MESSAGES/application.po</code> <code>locale/de/LC_MESSAGES/website.po</code>	Usual structure for storing gettext PO files.
<code>src/locale/(?P<component>[^\s.]*)\. (?P<language>[^\s.]*)\.po</code>	<code>src/locale/application.cs.po</code> <code>src/locale/website.cs.po</code> <code>src/locale/application.de.po</code> <code>src/locale/website.de.po</code>	Using both component and language name within filename.
<code>locale/(?P<language>[^\s.]*)(?P<component>[^\s.]*)(?P<language>[^\s.]*)\.po</code>	<code>locale/cs/application/cs.po</code> <code>locale/cs/website/cs.po</code> <code>locale/de/application/de.po</code> <code>locale/de/website/de.po</code>	Using language in both path and filename.
<code>res/values-(?P<language>[^\s.]*)/strings-(?P<component>[^\s.]*)\.xml</code>	<code>res/values-cs/strings-about.xml</code> <code>res/values-cs/strings-help.xml</code> <code>res/values-de/strings-about.xml</code> <code>res/values-de/strings-help.xml</code>	Android resource strings, split into several files.
<code>(?P<originalHierarchy>.*)(?P<component>[^\s.]*)/src/main/resources/ApplicationResources_(?P<language>[^\s.]*)\.properties</code>	<code>parent/module1/submodule/src/main/resources/ApplicationResources_fr.properties</code> <code>parent/module1/submodule/src/main/resources/ApplicationResource_es.properties</code> <code>parent/module2/src/main/resources/ApplicationResource_de.properties</code> <code>parent/module2/src/main/resources/ApplicationResource_ro.properties</code>	Multi-module Maven project with Java properties translations.

You can use Django template markup in both component name and the monolingual base filename, for example:

`{{ component }}`
Component filename match

`{{ component|title }}`
Component filename with upper case first letter

Save

Підказка: Додаток виявлення складників використовує *внутрішні адреси*. Це зручний спосіб використовувати однакові налаштування системи керування версіями для кількох складників одночасно. Пов'язані складники використовують локальний репозиторій головного складника шляхом вписування `weblate://проект/основний-складник` до поля *Репозиторій з джерелами* (у Керування ↓ Параметри ↓ Система керування версіями) кожного відповідного складника. Це також надає змогу заощадити час на налаштування та ресурси системи.

Дивись також:

Розмітка шаблонів

Масове редагування

Нове в версії 3.11.

Ідентифікатор додатка

`weblate.flags.bulk`

Налаштування

q	Запит	
state	Стан для встановлення	Доступні варіанти: -1 — не змінювати 10 — потребує редагування 20 — перекладено 30 — затверджено
add_flag	Додати наступні прапорці перекладу	
remove_flag	Вилучити наступні прапорці перекладу	
add_label	Мітки для додавання	
remove_label	Мітки для видалення	

Перемикачі

оновлення складника

Надає можливість пакетного редагування прапорців, міток або стану рядків.

Автоматизуйте встановлення міток для нових рядків — почніть з пошукового запиту `NOT has:label` і додавайте мітки, доки для всіх рядків буде встановлено потрібні мітки. Ви також можете виконувати будь-які інші автоматизовані дії для метаданих Weblate.

Приклади:

Table 5: Автоматичне встановлення міток нових рядків

Запит щодо пошуку	<code>NOT has:label</code>
Мітки для додавання	<code>recent</code>

Table 6: Позначення всіх рядків журналу змін Файли метаданих крамниці застосунків придатними лише для читання

Запит щодо пошуку	<code>language:en AND key:changelogs/</code>
Додати наступні прапорці перекладу	<code>read-only</code>

Дивись також:

Масове редагування, Налаштовування поведінки за допомогою прапорців, labels

Позначити незмінені переклади як такі, що потребують редагування

Нове в версії 3.1.

Ідентифікатор додатка

`weblate.flags.same_edit`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

обробка модуля після створення

Кожного разу, коли новий перекладний рядок імпортується з VCS, і він збігається із початковим рядком, його буде позначено як такий, що потребує редагування у Weblate. Це особливо корисно для форматів файлів, які включають початкові рядки, навіть ті, які не перекладено.

Підказка: Можливо, ви також захочете зробити строгішою перевірку *Переклад не змінено*, додавши до *Прапорці перекладу* прапорець `strict-same`.

Дивись також:

Стани перекладу

Позначити нові джерельні рядки як „потребують редагування“**Ідентифікатор додатка**

`weblate.flags.source_edit`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

обробка модуля після створення

Щоразу, коли новий джерельний рядок імпортується з VCS, він позначається, що потребує редагування в Weblate. Таким чином, ви можете легко фільтрувати та редагувати початкові рядки, написані розробниками.

Дивись також:

Стани перекладу

Позначити нові переклади як такі, що потребують редагування**Ідентифікатор додатка**

`weblate.flags.target_edit`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

обробка модуля після створення

Кожного разу, коли новий перекладний рядок імпортується з VCS, він позначається як такий, що потребує редагування в Weblate. Таким чином, ви можете легко фільтрувати та редагувати джерельні рядки, створені розробниками.

Дивись також:

[Стани перекладу](#)

Генератор статистики

Ідентифікатор додатка

`weblate.generate.generate`

Налаштування

<code>filename</code>	Назва створеного файла
<code>template</code>	Вміст створеного файлу

Перемикачі

обробка репозиторію до внеску

Створює файл з докладними відомостями щодо стану перекладу.

Ви можете скористатися шаблоном Django одразу для назви файла і вмісту. Докладний опис розмітки можна знайти у розділі *Розмітка шаблонів*.

Наприклад, створення файла резюме для кожного перекладу:

Назва створеного файла

`locale/{{ language_code }}.json`

Вміст

```
{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}",
}
```

Дивись також:

[Розмітка шаблонів](#)

Заповнити переклади початковими рядками

Нове в версії 4.11.

Ідентифікатор додатка

`weblate.generate.prefill`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

оновлення складників, щоденно

Заповнює рядки перекладу початковими рядками.

Усі неперекладені рядки складника буде заповнено початковими рядками та позначено такими, що потребують редагування. Скористайтеся цим, якщо у файла перекладу всі переклади мають бути непорожніми.

Створення псевдолокалей

Нове в версії 4.5.

Ідентифікатор додатка

`weblate.generate.pseudolocale`

Налаштування

<code>source</code>	Джерельні рядки	
<code>target</code>	Цільовий переклад	Усі рядки у цьому перекладі буде перезаписано
<code>prefix</code>	Фіксований префікс рядка	
<code>var_prefix</code>	Змінний префікс рядка	
<code>suffix</code>	Фіксований суфікс рядка	
<code>var_suffix</code>	Змінний суфікс рядка	
<code>var_multipl</code>	Множник змінної частини	Кількість повторів змінної частини, залежно від довжини початкового рядка.
<code>include_rea</code>	Включити рядки лише для читання	

Перемикачі

оновлення складників, щоденно

Автоматично створює переклад додаванням префікса і суфікса до початкових рядків.

Псевдолокалі корисні для пошуку рядків, які не приготовано до локалізації. Результат досягається зміною усіх придатних до перекладу початкових рядків таким чином, щоб рядок, який не змінено, було просто виявити під час запуску застосунку із відповідними рядками псевдолокальної мови.

Також можливий пошук рядків, локалізовані відповідники яких можуть не відповідати макету.

Використання змінних частин уможлиблює пошук рядків, які можуть не вміщатися до інтерфейсу користувача після локалізації — воно розширює текст на основі довжини початкового рядка. Змінні частини повторюються за довжиною тексту за множителем. Наприклад, `Hello world` зі змінним суфіксом `_` і змінним множителем `1` стане `Hello world_____` — суфікс буде повторено один раз для кожного символу у початковому рядку.

Рядки буде створено за таким взірцем:

Фіксований префікс рядка Змінний префікс рядка Джерельний рядок Змінний суфікс рядка Фіксований суфікс рядка

Підказка: Ви можете використовувати справжні мови для перевірки, але у Weblate доступні виділені псевдолокалі — `en_XA` and `ar_XB`.

Підказка: Ви можете скористатися цим додатком для того, щоб почати переклад новою локаллю наявної мови або подібною до іншої мовою. Щойно ви додасте переклад до складника, перейдіть до цього додатка. *Приклад:* якщо у вас вже є локаль `fr`, і ви хочете започаткувати переклад локаллю `fr_CA`, просто встановіть `fr` як джерело, `fr_CA` як ціль і лишіть поля префікса і суфікса порожніми.

Вилучте додаток після заповнення параметрів перекладу, щоб запобігти зміні перекладів після копіювання з боку Weblate.

Перекладачі в коментарі

Ідентифікатор додатка

`weblate.gettext.authors`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

обробка репозиторію до внеску

Оновлює коментар у заголовку файла PO так, щоб до нього було включено імена перекладачів та роки перекладу.

Заголовок файла PO виглядатиме якимось так:

```
# Michal Čihař <michal@weblate.org>, 2012, 2018, 2019, 2020.
# Pavel Borecki <pavel@example.com>, 2018, 2019.
# Filip Hron <filip@example.com>, 2018, 2019.
# anonymous <noreply@weblate.org>, 2019.
```

Оновити змінну ALL_LINGUAS у файлі «configure»

Ідентифікатор додатка

`weblate.gettext.configure`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

обробка репозиторію після додавання, щоденна обробка

Оновлює змінну ALL_LINGUAS у `configure`, `configure.in` або будь-яких файлах `configure.ac` при додаванні нового перекладу.

Налаштувати виведення gettext

Ідентифікатор додатка

`weblate.gettext.customize`

Налаштування

width	Об- ті- ка- ння дов- гих ряд- ків	Типово gettext переносить рядки на 77-у символі та на символах перенесення рядків, з параметром <code>—no-wrap</code> перенесення виконується лише на символах перенесення рядків. Доступні варіанти: 77 – перенесення рядків на 77 символах на символах перенесення рядків (типово для <code>xgettext</code>) 65535 – перенесення рядків лише після символу нового рядка (подібно до <code>«xgettext —no-wrap»</code>) -1 — без перенесення рядків
-------	--	---

Перемикачі

обробка після завантаження збережених даних

Уможливорює налаштування виведення gettext, наприклад, перенесення рядків.

Пропонує такі варіанти дій:

- Перенесення рядків на 77 символах на символах перенесення рядків

- Перенесення рядків лише після символу нового рядку
- Без перенесення рядків

Примітка: Типово gettext переносить рядки на 77-у символі та на символах перенесення рядків, з параметром `--no-wrap` перенесення виконується лише на символах перенесення рядків.

Оновити файл LINGUAS

Ідентифікатор додатка

`weblate.gettext.linguas`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

обробка репозиторію після додавання, щоденна обробка

Оновлює файл LINGUAS, коли додано новий переклад.

Генерувати мо-файли

Ідентифікатор додатка

`weblate.gettext.mo`

Налаштування

<code>path</code>	Шлях до згенерованого МО-файлу	Якщо не вказано, використовуватиметься розташування РО-файлу.
-------------------	--------------------------------	---

Перемикачі

обробка репозиторію до внеску

Автоматично створює мо-файл для кожного зміненого ро-файла.

Розташування створеного файла МО можна налаштувати. У полі для нього використовується *мова розмітки*.

Оновити файли ро, порівнявши із pot (msgmerge)

Ідентифікатор додатка

`weblate.gettext.msgmerge`

Налаштування

<code>previous</code>	Зберігати попередні msgid перекладених рядків
<code>no_location</code>	Видалить місця перекладених рядків
<code>fuzzy</code>	Використовувати нечіткий збіг

Перемикачі

обробка репозиторію після оновлення

Оновлює усі файли РО (які налаштовано у *Маска файлів*) так, щоб вони відповідали файлу РОТ (який налаштовано у *Шаблон для нових перекладів*), за допомогою **msgmerge**.

Спрацьовує щоразу, коли нові зміни витягуються із основного репозиторію даних, щоб вони відповідали `ref:component-new_base`. Ви можете налаштувати більшість параметрів командного рядка **msgmerge** за допомогою налаштувань додатка.

Дивись також:

Чи оновлює Weblate файли перекладів між перекладами?

Сполучити Git подання

Ідентифікатор додатка

weblate.git.squash

Налаштування

squash	Сполучення по-дань	Доступні варіанти: all — усі внески в один language — окремо для кожної мови file — окремо для кожного файла author — окремо для кожного автора
append	Дописати кінцеві-ки для об'єднаного повідомлення про внесок	Рядки-кінцевіки подібні до заголовків повідомлень електронної пошти за RFC 822. Їх дописують в кінці повідомлення про внесок у довільній формі, наприклад, «Co-authored-by: ...».
commit	Повідомлення щодо подання	Це повідомлення про внесок буде використано замість поєднаних повідомлень про внески з об'єднаних внесків.

Перемикачі

обробка репозиторію після внеску

Об'єднати Git подання перед відправленням змін.

Внески Git можна об'єднувати до запису змін з використанням одного з таких режимів:

- Всі подання в одне
- Для мови
- Для файлу
- Для кожного автора

Початкові повідомлення про внесок зберігаються, але авторство втрачається, якщо не позначено пункт *За автором* або повідомлення про внесок не налаштовано на його включення.

Початкові повідомлення про внески можна, якщо хочеться, перевизначити нетиповим повідомленням про внесок.

Трейлери (рядки внеску, подібні до Co-authored-by: ...) можна, якщо хочеться, вилучити з початкових повідомлень про внесок і дописати наприкінці об'єднаного повідомлення про внесок. Це також створює належний запис авторських прав Co-authored-by: для кожного перекладача.

Налагодити виведення JSON

Ідентифікатор додатка

weblate.json.customize

Налаштування

sort_keys	Сортувати ключі JSON	
indent	Відступи JSON	
style	Стиль відступів JSON	Доступні варіанти: spaces — пробіли tabs — табуляції

Перемикачі

обробка після завантаження збережених даних

Дозволяє налагодити виведення JSON, наприклад відступи та упорядкування.

Форматувати файл властивостей Java**Ідентифікатор додатка**`weblate.properties.sort`**Налаштування***Цей додаток не має налаштувань.***Перемикачі**

обробка репозиторію до внеску

Форматує й упорядковує файл властивостей Java.

- Встановлює усіди символи нового рядка Unix.
- Встановлює форматування керівних послідовностей Unicode (якщо такі є) з використанням великих літер.
- Вилучає порожні рядки і коментарі.
- Упорядковує рядки за ключовими значеннями.
- Відкидає рядки-повтори.

Видалення застарілого коментаря

Нове в версії 3.7.

Ідентифікатор додатка`weblate.removal.comments`**Налаштування**

age	Дні для збереження
-----	--------------------

Перемикачі

щоденно

Вказати термін вилучення застарілих коментарів.

Це може бути корисним для вилучення застарілих коментарів, які могли перестати бути актуальними. Користайтесь цією можливістю обережно, оскільки те, що коментар застарів, не означає, що він перестав бути важливим.

Видалити застарілу пропозицію

Нове в версії 3.7.

Ідентифікатор додатка`weblate.removal.suggestions`**Налаштування**

age	Дні для збереження	
votes	Поріг голосування	Поріг для видалення. Це поле не впливає при вимкненні голосування.

Перемикачі

щоденно

Встановити строк вилучення пропозицій.

Це може бути дуже корисним у поєднанні із голосуванням за пропозиції (див. *Експертне рецензування*) для вилучення пропозицій, які не отримують достатньої кількості позитивних голосів протягом вказаного часового проміжку.

Оновити файли RESX

Нове в версії 3.9.

Ідентифікатор додатка

`weblate.resx.update`

Налаштування

Цей додаток не має налаштувань.

Перемикачі

обробка репозиторію після оновлення

Оновити усі файли перекладу, щоб вони відповідали одномовному основному файлу. Невикористані рядки вилучаються, а нові додаються як копії початкового рядка.

Підказка: Скористайтеся *Очищення файлів перекладу*, якщо ви хочете лише вилучити застарілі переклади.

Дивись також:

Чи оновлює Weblate файли перекладів між перекладами?

Налагодити виведення XML

Нове в версії 4.15.

Ідентифікатор додатка

`weblate.xml.customize`

Налаштування

<code>closing_tags</code>	Включати завершальний тег для порожніх тегів XML
---------------------------	--

Перемикачі

обробка після завантаження збережених даних

Дозволяє налаштовувати поведінку виводу XML, наприклад, завершальні теги замість самозавершених тегів для порожнього вмісту.

Налагодити виведення YAML

Нове в версії 3.10.2.

Ідентифікатор додатка

`weblate.yaml.customize`

Налаштування

inde	Від- ступи YAML	
width	Обті- кання довгих рядків	Доступні варіанти: 80 — переносити рядок після 80 знаків 100 — перенесення рядків після 100 знаків 120 — перенесення рядків після 120 знаків 180 — перенесення рядків після 180 знаків 65535 — без перенесення рядків
line	Роз- риви рядків	Доступні варіанти: dos — DOS (\r\n) unix — UNIX (\n) mac — MAC (\r)

Перемикачі

обробка після завантаження збережених даних

Дозволяє налаштувати поведінку виводу YAML, наприклад, довжину рядка або нові рядки.

2.14.2 Налаштовування списку додатків

Список додатків налаштовується `WEBLATE_ADDONS`. Щоб додати додаток, просто до цього параметра включіть абсолютну назву класу.

2.14.3 Написання додатка

Ви також можете написати власні додатки — створіть підклас від `weblate.addons.base.BaseAddon`, щоб визначити метадані додатка і реалізувати зворотний виклик, який виконуватиме обробку.

Дивись також:

Розробка додатків

2.14.4 Виконання скриптів з додатка

Додатками також можна скористатися для виконання зовнішніх скриптів. Цю можливість інтегровано до Weblate, але тепер вам потрібно написати певний код обгортки вашого скрипту до стану додатка.

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""Example pre commit script."""

from django.utils.translation import gettext_lazy as _

from weblate.addons.events import EVENT_PRE_COMMIT
from weblate.addons.scripts import BaseScriptAddon

class ExamplePreAddon(BaseScriptAddon):
    # Event used to trigger the script
    events = (EVENT_PRE_COMMIT,)
    # Name of the addon, has to be unique
    name = "weblate.example.pre"
```

(continues on next page)

(continued from previous page)

```
# Verbose name and long description
verbose = _("Execute script before commit")
description = _("This add-on executes a script.")

# Script to execute
script = "/bin/true"
# File to add in commit (for pre commit event)
# does not have to be set
add_file = "po/{{ language_code }}.po"
```

Настанови щодо встановлення наведено у розділі *Нетипові перевірки якості, додатки та автоматичні виправлення*.

Скрипт виконується у поточному каталозі, яким є кореневий каталог репозиторію системи керування версіями для будь-якого заданого складника.

Крім того, можна скористатися такими змінними середовища:

WL_VCS

Використана система керування версіями.

WL_REPO

Адреса основного репозиторію із кодом.

WL_PATH

Абсолютний шлях до репозиторію системи керування версіями.

WL_BRANCH

Нове в версії 2.11.

Гілка репозиторію, яку налаштовано у поточному складнику.

WL_FILEMASK

Маска файлів для поточного складника.

WL_TEMPLATE

Назва файла шаблону для одномовного перекладу (може бути порожньою).

WL_NEW_BASE

Нове в версії 2.14.

Назва файла, який використовується для створення перекладів (може бути порожньою).

WL_FILE_FORMAT

Формат файла у поточному складнику.

WL_LANGUAGE

Мова поточного оброблюваного перекладу (недоступна для скриптів керування на рівні складника).

WL_PREVIOUS_HEAD

Попередній HEAD при оновленні (доступно, лише якщо запускається скрипт обробки після оновлення).

WL_COMPONENT_SLUG

Нове в версії 3.9.

Ідентифікатор складника, який використовується для побудови адреси.

WL_PROJECT_SLUG

Нове в версії 3.9.

Ідентифікатор проєкту, який використовується для побудови адреси.

WL_COMPONENT_NAME

Нове в версії 3.9.

Назва складника.

WL_PROJECT_NAME

Нове в версії 3.9.

Назва проєкту.

WL_COMPONENT_URL

Нове в версії 3.9.

Адреса складника.

WL_ENGAGE_URL

Нове в версії 3.9.

Адреса роботи з проєктом.

Дивись також:

Налаштування складників

Обробка репозиторію після оновлення

Можна скористатися для оновлення файлів перекладу, коли міняється джерело у основному репозиторії системи керування версіями. Слід пам'ятати, що Weblate «бачить» лише файли, які внесено до системи керування версіями, тому вам доведеться внести зміни до репозиторію у самому скрипті.

Наприклад, з Gulp ви можете зробити це за допомогою такого коду:

```
#!/bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Обробка перекладів перед внесенням до репозиторію

Скористайтеся скриптом подання даних (commit) для автоматичного внесення змін до перекладу перед тим, як його буде внесено до репозиторію.

Передається як один параметр, що складається з назви файла поточного перекладу.

2.15 Пам'ять перекладів

Нове в версії 2.20.

Weblate постачається із вбудованою пам'яттю перекладів, яка складається з таких даних:

- Вручну імпортована пам'ять перекладів (див. *Інтерфейс користувача*).
- Автоматично збережених перекладів, які було здійснено на Weblate (залежно від *Області дії пам'яті перекладів*).
- Автоматично імпортовані попередні переклади.

Вміст пам'яті перекладів може бути застосовано у один з двох способів:

- Вручну, панель *Автоматичні пропозиції* під час перекладу.
- Автоматично, перекладом рядків за допомогою *Автоматичний переклад*, або додатка *Автоматичний переклад*.

Для отримання підказок щодо встановлення, див. розділ *Пам'ять перекладів Weblate*, типово увімкнено.

2.15.1 Області дії пам'яті перекладів

Нове в версії 3.2: У попередніх версіях пам'ять перекладів можна було завантажувати лише з файла, що відповідає поточній області імпортованої пам'яті перекладів.

Області пам'яті перекладів використовуються для забезпечення конфіденційності та поширення перекладів, залежно від бажаної поведінки.

Імпортована пам'ять перекладів

Імпортування довільних даних пам'яті перекладів за допомогою команди `import_memory` робить вміст пам'яті перекладів доступним для всіх користувачів і проєктів.

Пам'ять перекладів окремого користувача

Зберігає усі переклади користувача автоматично до особистої пам'яті перекладів кожного користувача.

Пам'ять перекладів окремого проєкту

Усі переклади у проєкті автоматично зберігаються у пам'яті перекладів проєкту, яка буде доступною лише для цього проєкту.

Спільна пам'ять перекладів

Усі переклади у проєктах, для яких увімкнено спільну пам'ять перекладів, зберігаються у спільній пам'яті перекладів, яка доступна для всіх проєктів.

Будь ласка, ретельно зважте, чи слід вмикати цю можливість для екземплярів Weblate, які є спільними для різних проєктів, оскільки вмикання може призвести до суттєвих наслідків:

- Переклади може бути використано будь-ким.
- Поширення перекладів може призвести до розкриття таємних відомостей.

2.15.2 Керування пам'яттю перекладів

Інтерфейс користувача

Нове в версії 3.2.

За допомогою базового інтерфейсу користувача ви можете керувати записами пам'яті перекладів для окремого користувача або окремого проєкту. Цим інтерфейсом можна скористатися для отримання, очищення або імпортування пам'яті перекладів.

Підказка: До Weblate можна імпортувати пам'ять перекладів у форматі JSON. Обробку TMX передбачено для обміну даними із іншими інструментами.

Дивись також:

Схема пам'яті перекладів Weblate

Інтерфейс керування

Передбачено декілька команд керування для роботи із вмістом пам'яті перекладів. Ці команди працюють із усією пам'яттю перекладів, без фільтрування за областями (якщо це не вказано параметрами):

`dump_memory`

Експортує пам'ять до JSON

`import_memory`

Імпортує файли TMX або JSON до пам'яті перекладів

2.16 Налаштування

Усі параметри зберігаються у файлі `settings.py` (звична картина для Django).

Примітка: Після внесення змін до будь-яких з цих параметрів вам слід перезапустити Weblate — обидва процеси, WSGI і Celery.

У випадку, якщо Weblate запущено як `mod_wsgi`, вам слід перезапустити Apache для перезавантаження налаштувань.

Дивись також:

Будь ласка, також ознайомтеся із [документацією до Django](#), щоб дізнатися більше про параметри налаштування самого Django.

2.16.1 AKISMET_API_KEY

Weblate може використовувати Akismet для перевірки вхідних анонімних пропозицій на спам. Відвідайте akismet.com, щоб придбати ключ до програмного інтерфейсу і пов'яжіть його із сайтом.

2.16.2 ANONYMOUS_USER_NAME

Імена користувачів, які не увійшли до системи.

Дивись також:

Керування доступом

2.16.3 AUDITLOG_EXPIRY

Нове в версії 3.6.

Кількість днів, протягом яких Weblate має зберігати журнали аудиту, які містять дані щодо дій з облікових записів.

Типовим є значення у 180 днів.

2.16.4 AUTH_LOCK_ATTEMPTS

Нове в версії 2.14.

Максимальна кількість спроб невдалого розпізнавання, перш ніж буде застосовано обмеження швидкості.

У поточній версії застосовується у таких місцях:

- Вхід. Вилучає пароль до облікового запису, забороняючи користувачеві входити до системи без запиту щодо введення нового пароля.
- Скидання пароля. Забороняє надсилання нових повідомлень електронної пошти, не набридаючи користувачам надто великою кількістю повідомлень щодо скидання пароля.

Типове значення — 10.

Дивись також:

Обмеження частоти

2.16.5 AUTO_UPDATE

Нове в версії 3.2.

Змінено в версії 3.11: Початковий параметр зі станами «увімкнено» і «вимкнено» було змінено для розрізнення прийнятних рядків.

Оновлює усі репозиторію щоденно.

Підказка: Корисно, якщо ви не використовуєте *Обробники сповіщень* для автоматичного оновлення репозиторіїв Weblate.

Примітка: Окрім вибору рядка, передбачено значення «увімкнено» і «вимкнено» для зворотної сумісності.

Можливі варіанти:

"none"

Без щоденних оновлень.

"remote" також False

Лише оновити віддалені репозиторію.

"full" також True

Оновити віддалені репозиторію і злити із робочою копією.

Примітка: Це потребує працездатної *Фонові завдання з використанням Celery* і набуває чинності після перезапуску.

2.16.6 AVATAR_URL_PREFIX

Префікс для побудови адрес аватарів: `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. Перевірено працездатність таких служб:

Gravatar (типовий), як на <https://gravatar.com/>

```
AVATAR_URL_PREFIX = 'https://www.gravatar.com/'
```

Libravatar, як на <https://www.libravatar.org/>

```
AVATAR_URL_PREFIX = 'https://www.libravatar.org/'
```

Дивись також:

Кешування аватару, ENABLE_AVATARS, Аватари

2.16.7 AUTH_TOKEN_VALID

Нове в версії 2.14.

Визначає строк дії ключа розпізнавання та тимчасового пароля з повідомлень електронної пошти щодо скидання пароля. Визначається у секундах. Типове значення — 172800 (2 дні).

2.16.8 AUTH_PASSWORD_DAYS

Нове в версії 2.15.

Кількість днів, протягом яких Weblate відмовлятиметься повторно використовувати раніше використаний пароль для користувача.

Перевірку засновано на журналі ревізії, *AUDITLOG_EXPIRY* має бути принаймні таким самим великим, як це значення.

Примітка: Зміни пароля, виконані до Weblate 2.15 не братимуться до уваги у цьому правилі.

Типовим є значення у 180 днів.

2.16.9 AUTOFIX_LIST

Список автоматичних виправлено, які слід застосувати при збереженні рядка.

Примітка: Надайте повний шлях до класу Python, який реалізує інтерфейс автоматичного виправлення.

Доступні виправлення:

`weblate.trans.autofixes.whitespace.SameBookendingWhitespace`

Відповідає пробілу на початку і наприкінці початкового рядка.

`weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis`

Замінює кінцеві крапки (...), якщо у рядку початкового коду міститься відповідна багатокрапка (...).

`weblate.trans.autofixes.chars.RemoveZeroSpace`

Вилучає пробільні символи нульової ширини, якщо у початковому коді міститься ці символи.

`weblate.trans.autofixes.chars.RemoveControlChars`

Вилучає керівні символи, якщо у початковому коді міститься ці символи.

`weblate.trans.autofixes.html.BleachHTML`

Вилучає небезпечну розмітку HTML з рядків, які позначено як `safe-html` (див. *Небезпечний HTML-код*).

Ви можете вибрати, яким з них скористатися:

```
AUTOFIX_LIST = (  
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",  
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",  
)
```

Дивись також:

Автоматичні виправлення, Нетипові автоматичні виправлення

2.16.10 BACKGROUND_TASKS

Нове в версії 4.5.2.

Визначає частоту, із якою запускатимуться тривалі завдання для складника.

У поточній версії керує такими елементами:

- додатком *Автоматичний переклад*
- повторною обробкою *Перевірки і виправлення*

Можливі варіанти значень:

- `monthly` (щомісяця, типове значення)
- `weekly` (щотижня)
- `daily` (щодня)
- `never` (ніколи)

Примітка: Не рекомендуємо збільшувати частоту, якщо у Weblate тисячі складників.

2.16.11 BASIC_LANGUAGES

Нове в версії 4.4.

Список мов, які буде запропоновано користувачам при започаткуванні нового перекладу. Якщо не вказано, буде використано вбудований список, до якого включено найпоширеніші мови, але без діалектів певних країн.

Це заборонить додавати небажані мови лише непривілейованим користувачам. Адміністраторам проекту все одно буде надано повний набір мов, які визначено у Weblate.

Примітка: Це не визначає нових мов для Weblate, лише фільтрує наявні у базі даних.

Приклад:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Дивись також:

Визначення мов

2.16.12 BORG_EXTRA_ARGS

Нове в версії 4.9.

Ви можете передати додаткові аргументи **borg create**, якщо увімкнено вбудоване резервне копіювання.

Приклад:

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/"]
```

Дивись також:

Резервне копіювання і пересування Weblate, borg create

2.16.13 CACHE_DIR

Нове в версії 4.16.

Каталог, у якому Weblate зберігає файли кешу. Типовим є підкаталог `cache` у `DATA_DIR`.

Змініть це на локальну або тимчасову файлову систему, якщо `DATA_DIR` є каталогом мережевої файлової системи.

Контейнер Docker використовує для цього окремий том, див. *Томи контейнера Docker*.

2.16.14 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Коригування заголовка Content-Security-Policy у Weblate. Заголовок створюється автоматично на основі увімкнених інтеграцій зі сторонніми службами (Matomo, Google Analytics, Sentry, ...).

Для всіх цих змінних типове значення — це порожній список.

Приклад:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Дивись також:

Правила щодо безпеки даних, Content Security Policy (CSP)

2.16.15 CHECK_LIST

Список перевірок якості, які слід виконати над перекладом.

Примітка: Надайте повний шлях до класу Python, який реалізує інтерфейс перевірок.

Скоригувати список перевірок для включення потрібних вам.

Усі вбудовані *Перевірки якості* типово увімкнено. Ви можете змінити ці параметри. Типово, записи у *Зразок налаштувань* закоментовано, тому використовуються типові значення. Нові перевірки додаються із кожною новою версією Weblate.

Ви можете вимкнути усі перевірки:

```
CHECK_LIST = ()
```

Ви можете увімкнути лише декілька:

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

Примітка: Зміна значення цього параметра стосується лише змінених наново перекладів — наявні перевірки зберігатимуться у базі даних. Щоб зміни було застосовано до вже збережених перекладів, віддайте команду *updatechecks*.

Дивись також:

Перевірки якості, Налаштовування поведінки за допомогою прапорців

2.16.16 COMMENT_CLEANUP_DAYS

Нове в версії 3.6.

Вилучати коментарі за вказану кількість днів. Типове значення — None, тобто не вилучати коментарі.

2.16.17 COMMIT_PENDING_HOURS

Нове в версії 2.10.

Кількість годин, протягом яких слід надіслати до репозиторію зміни з черги разом із фоновим завданням.

Дивись також:

Налаштовування складників, Вік змін для подання, Запуск завдань щодо супроводу, commit_pending

2.16.18 CONTACT_FORM

Нове в версії 4.6.

Налаштовує спосіб надсилання повідомлення електронної пошти з форми зв'язку. Виберіть налаштування, які відповідають налаштуванням вашого поштового сервера.

"reply-to"

Запис відправника той самий, що і у *Reply-To*, це типова поведінка.

"from"

Запис відправника той самий, що і у *From*. На вашому поштовому сервері має бути дозволено надсилання таких повідомлень електронної пошти.

2.16.19 DATA_DIR

Тека, у якій Weblate зберігає дані. Тека містить посилання на репозиторію системи керування версіями, повнотекстовий показник та різноманітні файли налаштувань для зовнішніх інструментів.

Зазвичай, існують такі підкаталоги:

home

Домашній каталог, який використовується для виклику скриптів.

ssh

Ключі і налаштування SSH.

static

Типове місце для статичних файлів Django, визначається на основі `STATIC_ROOT`. Див. *Обслуговування статичних файлів*.

Контейнер Docker використовує для цього окремий том, див. *Томи контейнера Docker*.

media

Типове розташування мультимедійних файлів Django, визначається на основі `MEDIA_ROOT`. Містить вивантажені знімки вікон, див. *Візуальний контекст для рядків*.

vcs

Репозиторії систем керування версіями для перекладів.

backups

Дані щоденної резервної копії. Докладніше про це у розділі *Дампи даних для резервних копій*.

fonts:

Вивантажені користувачем шрифти, див. *Керування шрифтами*.

cache

Різнорічні кеші, можна розташувати деінде за допомогою `CACHE_DIR`.

Контейнер Docker використовує для цього окремий том, див. *Томи контейнера Docker*.

Примітка: Цей каталог має бути доступний до запису для Weblate. Якщо його запущено від імені uWSGI, користувач `www-data` повинен мати доступ до запису до нього.

Найпростішим способом досягти результату є надання користувачеві прав власника каталогу:

```
sudo chown www-data:www-data -R $DATA_DIR
```

Типове значення — `/home/weblate/data`, але варто налаштувати все відповідним чином.

Дивись також:

Права доступу у файлової системі, Резервне копіювання і пересування Weblate, `CACHE_DIR`

2.16.20 DATABASE_BACKUP

Нове в версії 3.1.

Чи мають резервні копії зберігатися у форматі звичайного тексту, стисненими чи пропущеними. Коректними значеннями є такі:

- "plain"
- "compressed"
- "none"

Дивись також:

Резервне копіювання і пересування Weblate

2.16.21 DEFAULT_ACCESS_CONTROL

Нове в версії 3.3.

Типовий параметр керування доступом для нових проєктів:

0	<i>Відкритий</i>
1	<i>Захищений</i>
100	<i>Приватний</i>
200	<i>Нетиповий</i>

Скористайтесь варіантом *Нетиповий*, якщо ви керуєте ACL вручну, тобто не покладаєтеся на внутрішні механізми керування Weblate.

Дивись також:

Керування доступом на рівні проєкту, Керування доступом

2.16.22 DEFAULT_AUTO_WATCH

Нове в версії 4.5.

Налаштовує, чи має бути увімкнено *Автоматичне спостереження за проєктами із внеском* для нових користувачів. Типове значення — True.

Дивись також:

Сповіщення

2.16.23 DEFAULT_RESTRICTED_COMPONENT

Нове в версії 4.1.

Типове значення для обмеження на доступ до складника.

Дивись також:

Обмежений доступ, Область команд

2.16.24 DEFAULT_ADD_MESSAGE, DEFAULT_COMMIT_MESSAGE, DEFAULT_MERGE_MESSAGE

DEFAULT_ADDON_MESSAGE, DEFAULT_DELETE_MESSAGE,

Типові повідомлення про внесок для різних дій. Докладніше про це у розділі *Налаштовування складників*.

Дивись також:

Розмітка шаблонів, Налаштовування складників, Повідомлення щодо внеску, додавання, вилучення, повідомлення додатків та запитів щодо об'єднання

2.16.25 DEFAULT_ADDONS

Типові додатки, які слід встановлювати для всіх створених складників.

Примітка: Цей параметр стосується лише новостворених складників.

Приклад:

```
DEFAULT_ADDONS = {
    # Add-on with no parameters
    "weblate.flags.target_edit": {},
    # Add-on with parameters
    "weblate.autotranslate.autotranslate": {
        "mode": "suggest",
        "filter_type": "todo",
        "auto_source": "mt",
        "component": "",
        "engines": ["weblate-translation-memory"],
        "threshold": "80",
    },
}
```

Дивись також:

install_addon, Додатки, WEBLATE_ADDONS

2.16.26 DEFAULT_COMMITER_EMAIL

Нове в версії 2.4.

Адреса електронної пошти автора внеску. Типове значення — `noreply@weblate.org`.

Дивись також:

DEFAULT_COMMITER_NAME

2.16.27 DEFAULT_COMMITER_NAME

Нове в версії 2.4.

Ім'я автора внеску. Типове значення — `Weblate`.

Дивись також:

DEFAULT_COMMITER_EMAIL

2.16.28 DEFAULT_LANGUAGE

Нове в версії 4.3.2.

Типова початкова мова, наприклад, у *Джерельна мова*.

Типовим значенням є *en*. Відповідний об'єкт мови має існувати у базі даних.

Дивись також:

Визначення мов, Джерельна мова

2.16.29 DEFAULT_MERGE_STYLE

Нове в версії 3.4.

Стиль злиття для всіх нових складників.

- *rebase* — типовий
- *merge*

Дивись також:

Налаштовування складників, Стиль злиття

2.16.30 DEFAULT_SHARED_TM

Нове в версії 3.2.

Налаштовує типове значення *Використовувати спільну пам'ять перекладів* і *Зробіть внесок у спільну пам'ять перекладів*.

2.16.31 DEFAULT_TRANSLATION_PROPAGATION

Нове в версії 2.5.

Типовий параметр для надсилання перекладу, типове значення — `True`.

Дивись також:

Налаштовування складників, Дозволити поширення перекладу

2.16.32 DEFAULT_PULL_MESSAGE

Налаштовує типовий заголовок та повідомлення для запитів щодо об'єднання.

2.16.33 ENABLE_AVATARS

Визначає, чи слід вмикати засновані на Gravatar аватари для користувачів. Типово увімкнено.

Зображення аватарів отримуються і кешуються на сервері — це знижує ризик витоку приватних даних і пришвидшує обробку даних для користувача.

Дивись також:

Кешування аватару, AVATAR_URL_PREFIX, Аватари

2.16.34 ENABLE_HOOKS

Чи слід вмикати доступу до анонімних віддалених скриптів обробки.

Дивись також:

Обробники сповіщень

2.16.35 ENABLE_HTTPS

Визначає, надсилати посилання Weblate як HTTPS або як HTTP. Цей параметр впливає на надіслані повідомлення електронної пошти і створення абсолютних адрес.

У типових налаштуваннях цей параметр також використовується для декількох параметрів Django, які пов'язано із HTTPS — вмикає безпечні куки, перемикає HSTS або вмикає переспрямовування на адресу HTTPS.

У певних випадках переспрямовування HTTPS може викликати проблеми — ви можете зіткнутися із проблемами циклічного переспрямовування, якщо ви використовуєте реверсивний проксі-сервер, який виконує обривання SSL і не передає належним чином заголовки протоколів до Django. Будь ласка, скоригуйте налаштування вашого реверсивного проксі-сервера так, щоб він надсилав заголовки X-Forwarded-Proto або Forwarded, або налаштуйте `SECURE_PROXY_SSL_HEADER`, щоб надати Django змогу правильно визначити стан SSL.

Дивись також:

`SESSION_COOKIE_SECURE,` `CSRF_COOKIE_SECURE,` `SECURE_SSL_REDIRECT,`
`SECURE_PROXY_SSL_HEADER` *Установіть належний домен сайту*

2.16.36 ENABLE_SHARING

Увімкнути або вимкнути меню *Поділитися*, щоб користувачі могли оприлюднювати поступ перекладу у соціальних мережах.

2.16.37 EXTRA_HTML_HEAD

Нове в версії 4.15.

Вставити додаткову розмітку у заголовок HTML. Можна скористатися, наприклад, для перевірки власності сайту:

```
EXTRA_HTML_HEAD = '<link href="https://fosstodon.org/@weblate" rel="me">'
```

Попередження: Обробка рядка не виконуватиметься, його буде просто вставлено до заголовка HTML.

2.16.38 GET_HELP_URL

Нове в версії 4.5.2.

Адреса, за якою можна отримати підтримку для вашого екземпляра Weblate.

2.16.39 GITEA_CREDENTIALS

Нове в версії 4.12.

Список реєстраційних даних для серверів Gitea.

```
GITEA_CREDENTIALS = {
  "try.gitea.io": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitea.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Дивись також:

Запити щодо злиття Gitea, Створення особистого жетона доступу Gitea

2.16.40 GITLAB_CREDENTIALS

Нове в версії 4.3.

Список реєстраційних даних для серверів GitLab.

```
GITLAB_CREDENTIALS = {
  "gitlab.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitlab.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Дивись також:

Запити щодо об'єднання GitLab, GitLab: особистий жетон доступу

2.16.41 GITHUB_CREDENTIALS

Нове в версії 4.3.

Список реєстраційних даних для серверів GitHub.

```
GITHUB_CREDENTIALS = {
  "api.github.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "github.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Дивись також:

Запити щодо злиття GitHub, створення особистого жетона доступу GitHub

2.16.42 BITBUCKETSERVER_CREDENTIALS

Нове в версії 4.16.

Список реєстраційних даних для серверів Bitbucket.

```
BITBUCKETSERVER_CREDENTIALS = {  
    "git.self-hosted.com": {  
        "username": "weblate",  
        "token": "http-access-token",  
    },  
}
```

Дивись також:

Запит щодо злиття для сервера Bitbucket, Bitbucket: жетон доступу HTTP

2.16.43 GOOGLE_ANALYTICS_ID

Ідентифікатор Google Analytics для вмикання спостереження за Weblate за допомогою Google Analytics.

2.16.44 HIDE_REPO_CREDENTIALS

Приховати реєстраційні дані репозиторію від показу у вебінтерфейсі. Якщо у адресі репозиторію міститься ім'я користувача і пароль, Weblate приховає їх при показі відповідних даних користувачам.

Наприклад, замість `https://користувач:пароль@git.example.com/repo.git` буде показано просто `https://git.example.com/repo.git`. Програма також намагатиметься вилучити повідомлення про помилки системи керування версіями у подібний же спосіб.

Примітка: Типово увімкнено.

2.16.45 HIDE_VERSION

Нове в версії 4.3.1.

Ховає відомості щодо версії від користувачів, які не пройшли розпізнавання. Це також призведе до того, що усі посилання на документацію вказуватимуть на документацію із найсвіжішої версії, замість документації відповідної встановленої версії.

Приховування версії є рекомендованим для посилення захисту у деяких корпораціях, але воно не захистить вас від спроб нападника визначити версію за зондуванням її поведінки.

Примітка: Типово вимкнено.

2.16.46 INTERLEDGER_PAYMENT_POINTERS

Нове в версії 4.12.1.

Список Interledger Payment Pointers (ILPs) для вебмонетизації.

Якщо вказано декілька, ймовірнісний розподіл прибутку буде виконано випадковим вибором якогось.

Докладніші відомості можна знайти на <<https://webmonetization.org/>>.

Підказка: Типове значення уможливорює фінансову підтримку користувачами самого Weblate.

2.16.47 IP_BEHIND_REVERSE_PROXY

Нове в версії 2.14.

Вказує, чи працює Weblate за реверсивним проксі-сервером.

Якщо встановлено значення True, Weblate отримуватиме IP-адресу з заголовка, який визначено `IP_PROXY_HEADER`.

Попередження: Переконайтеся, що зворотний проксі-сервер справді використовується, і що він встановлює цей заголовок, інакше користувачі зможуть підробляти IP-адресу.

Примітка: Типово не увімкнено.

Дивись також:

Робота за реверсивним проксі-сервером, Обмеження частоти, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`

2.16.48 IP_PROXY_HEADER

Нове в версії 2.14.

Вказує, з якого заголовка Weblate має отримати адресу, коли увімкнено `IP_BEHIND_REVERSE_PROXY`.

Типове значення — `HTTP_X_FORWARDED_FOR`.

Дивись також:

Робота за реверсивним проксі-сервером, Обмеження частоти, `SECURE_PROXY_SSL_HEADER`, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_OFFSET`

2.16.49 IP_PROXY_OFFSET

Нове в версії 2.14.

Вказує, яку частину `IP_PROXY_HEADER` має бути використано як клієнтську IP-адресу.

Залежно від вашої конфігурації, цей заголовок може складатися з декількох IP-адрес (наприклад, `X-Forwarded-For: a, b, client-ip`), і ви можете налаштувати, яку адресу з заголовка буде використано як клієнтську IP-адресу тут.

Попередження: Встановлення цього параметра впливає на безпеку вашої установки, вам слід налаштувати її лише на використання довірених проксі-серверів для визначення IP-адреси.

Типове значення — 0.

Дивись також:

Робота за реверсивним проксі-сервером, Обмеження частоти, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_HEADER

2.16.50 LEGAL_TOS_DATE

Нове в версії 4.15.

Примітка: Щоб мати змогу цим скористатися, вам слід встановити *Правові відомості*.

Дата останнього оновлення документів щодо умов надання послуг. Кожного разу, коли змінюється дата, користувачі мають погоджуватися із умовами надання послуг.

```
from datetime import date

LEGAL_TOS_DATE = date(2022, 2, 2)
```

2.16.51 LEGAL_URL

Нове в версії 3.5.

Адреса, за якою екземпляр Weblate показуватиме правові документи.

Підказка: Корисний, якщо ви зберігаєте ваші правові документи поза Weblate для вбудовування їх до Weblate. Докладніше про це у розділі *Правові відомості*.

Приклад:

```
LEGAL_URL = "https://weblate.org/terms/"
```

Дивись також:

PRIVACY_URL

2.16.52 LICENSE_EXTRA

Додаткові ліцензії, які слід включити до варіантів ліцензування.

Примітка: Кожне визначення ліцензії має бути кортежем його короткої назви, довгої назви і адреси.

Приклад:

```
LICENSE_EXTRA = [
    (
        "AGPL-3.0",
        "GNU Affero General Public License v3.0",
        "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
    ),
]
```

2.16.53 LICENSE_FILTER

Змінено в версії 4.3: Відтепер, якщо спорожнити це значення, попередження про ліцензію буде вимкнено. Фільтрувати перелік ліцензій для показу. Це також вимкне оповіщення про ліцензію, якщо спорожнити значення.

Примітка: У цьому фільтрі використовуються скорочені назви ліцензій.

Приклад:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

Це вимкне попередження про ліцензію:

```
LICENSE_FILTER = set()
```

Дивись також:

alerts

2.16.54 LICENSE_REQUIRED

Визначає, чи є обов'язковим атрибут ліцензування у записі *Налаштовування складників*.

Примітка: Типово вимкнено.

2.16.55 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Визначає, чи має бути обмежено довжину вказаного перекладу. Обмеження довжини — довжина початкового рядка × 10 символів.

Підказка: Установіть значення `False`, щоб уможливити довші переклади (аж до 10000 символів), незалежно від довжини початкового рядка.

Примітка: Типове значення — `True`.

2.16.56 LOCALIZE_CDN_URL і LOCALIZE_CDN_PATH

Ці параметри призначено для налаштування додатка *CDN локалізації JavaScript*. `LOCALIZE_CDN_URL` визначає кореневу адресу, за якою доступний CDN локалізації, а `LOCALIZE_CDN_PATH` визначає шлях, за яким Weblate має зберігати створені файли, які обслуговуватимуться за адресою `LOCALIZE_CDN_URL`.

Підказка: На базовому Weblate використано `https://weblate-cdn.com/`.

Дивись також:

CDN локалізації JavaScript

2.16.57 LOGIN_REQUIRED_URLS

Список адрес, для яких обов'язковим буде реєстрація. (Окрім стандартних правил, які вбудовано до Weblate).

Підказка: За допомогою цього параметра ви можете захистити паролем увесь екземпляр за допомогою:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

Підказка: Бажано заблокувати також і доступ до програмного інтерфейсу, як показано у наведеному вище прикладі.

Дивись також:

[REQUIRE_LOGIN](#)

2.16.58 LOGIN_REQUIRED_URLS_EXCEPTIONS

Список винятків для [LOGIN_REQUIRED_URLS](#). Якщо не вказано, користувачам буде відкрито доступу до сторінки входу до системи.

Деякі винятки, які вам варто включити:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r"/accounts/(.*)$", # Required for sign in
    r"/static/(.*)$", # Required for development mode
    r"/widgets/(.*)$", # Allowing public access to widgets
    r"/data/(.*)$", # Allowing public access to data exports
    r"/hooks/(.*)$", # Allowing public access to notification hooks
    r"/api/(.*)$", # Allowing access to API
    r"/js/i18n/$", # JavaScript localization
)
```

2.16.59 MATOMO_SITE_ID

Ідентифікатор сайта Matomo (раніше Piwik), за яким ви хочете стежити.

Примітка: У цьому варіанті інтеграції не передбачено підтримки керування мітками Matomo.

Дивись також:

[MATOMO_URL](#)

2.16.60 MATOMO_URL

Повна адреса (разом із кінцевою похилою рисою) встановленого Matomo (раніше Piwik), яким ви хочете скористатися для стеження за використанням Weblate. Будь ласка, ознайомтеся із <<https://matomo.org/>>, щоб дізнатися більше.

Підказка: У цьому варіанті інтеграції не передбачено підтримки керування мітками Matomo.

Приклад:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Дивись також:

MATOMO_SITE_ID

2.16.61 NEARBY_MESSAGES

Кількість показаних рядків, які розташовано поруч із поточним рядком перекладу. Це лише типове значення, користувачі можуть змінювати його за допомогою *Профіль користувача*.

2.16.62 DEFAULT_PAGE_LIMIT

Нове в версії 4.7.

Типова кількість показуваних елементів, коли увімкнено розбиття на сторінки.

2.16.63 PAGURE_CREDENTIALS

Нове в версії 4.3.2.

Список реєстраційних даних для серверів Pagure.

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

Дивись також:

Запити щодо об'єднання Pagure, програмний інтерфейс Pagure

2.16.64 PRIVACY_URL

Нове в версії 4.8.1.

Адреса, за якою екземпляр Weblate показуватиме правила конфіденційності.

Підказка: Корисний, якщо ви зберігаєте ваші правові документи поза Weblate для вбудовування їх до Weblate. Докладніше про це у розділі *Правові відомості*.

Приклад:

```
PRIVACY_URL = "https://weblate.org/terms/"
```

Дивись також:

LEGAL_URL

2.16.65 PRIVATE_COMMIT_EMAIL_OPT_IN

Нове в версії 4.15.

Налаштовує, чи є конфіденційність адреси електронної пошти при внеску такою, з якою погоджуються, чи такою, від якої відмовляються (типово, вважається, що від неї відмовляються).

Дивись також:

Профіль, PRIVATE_COMMIT_EMAIL_TEMPLATE

2.16.66 PRIVATE_COMMIT_EMAIL_TEMPLATE

Нове в версії 4.15.

Шаблон для створення конфіденційної адреси електронної пошти для користувача. Типовим є "{користувач}@users.noreply.{сайт_домен}".

Встановіть порожні рядок, щоб вимкнути.

Примітка: Використання іншої адреси електронної пошти для внесків є об'єктом погодження з боку користувача, якщо це не змінено за допомогою *PRIVATE_COMMIT_EMAIL_OPT_IN*. Користувачі можуть налаштувати адресу електронної пошти для внесків у *профільі*.

2.16.67 PROJECT_BACKUP_KEEP_COUNT

Нове в версії 4.14.

Визначає, скільки резервних копій на проєкт слід зберігати на сервері. Типовим значенням є 3.

Дивись також:

Резервне копіювання на рівні проєктів

2.16.68 PROJECT_BACKUP_KEEP_DAYS

Нове в версії 4.14.

Визначає, наскільки довго зберігатимуться резервні копії проєкту на сервері. Типовим значенням є 30 днів.

Дивись також:

Резервне копіювання на рівні проєктів

2.16.69 PROJECT_NAME_RESTRICT_RE

Нове в версії 4.15.

Визначає формальний вираз для обмеження назв проєктів. Усі назви, відповідність яких виразу буде встановлено, буде заборонено.

Дивись також:

Назва проєкту

2.16.70 PROJECT_WEB_RESTRICT_HOST

Нове в версії 4.16.2.

Заборонити використання певних хостів на сайті проєкту. Перевіряється будь-який піддомен, тому включення `example.com` також заблокує `test.example.com`. Список повинен містити тільки рядки в нижньому регістрі, аналіз домену перед зіставленням починається з нижнього регістру.

Типова конфігурація:

```
PROJECT_WEB_RESTRICT_HOST = {"localhost"}
```

Дивись також:

Вебсайт проєкту `PROJECT_WEB_RESTRICT_NUMERIC`, `PROJECT_WEB_RESTRICT_RE`,

2.16.71 PROJECT_WEB_RESTRICT_NUMERIC

Нове в версії 4.16.2.

Заборонити використання числової IP-адреси на сайті проєкту. Типово ввімкнено.

Дивись також:

Вебсайт проєкту `PROJECT_WEB_RESTRICT_HOST`, `PROJECT_WEB_RESTRICT_RE`,

2.16.72 PROJECT_WEB_RESTRICT_RE

Нове в версії 4.15.

Визначає формальний вираз для обмеження сайтів проєкту. Усі адреси, відповідність яких виразу буде встановлено, буде заборонено.

Дивись також:

Вебсайт проєкту `PROJECT_WEB_RESTRICT_HOST`, `PROJECT_WEB_RESTRICT_NUMERIC`

2.16.73 RATELIMIT_ATTEMPTS

Нове в версії 3.2.

Максимальна кількість спроб розпізнавання, перш ніж буде застосовано обмеження швидкості.

Типове значення — 5.

Дивись також:

Обмеження частоти, `RATELIMIT_WINDOW`, `RATELIMIT_LOCKOUT`

2.16.74 RATELIMIT_WINDOW

Нове в версії 3.2.

Тривалість періоду прийняття розпізнавання після застосування обмеження частоти.

Проміжок часу у секундах, типове значення — 300 (5 хвилин).

Дивись також:

Обмеження частоти, `RATELIMIT_ATTEMPTS`, `RATELIMIT_LOCKOUT`

2.16.75 RATELIMIT_LOCKOUT

Нове в версії 3.2.

Тривалість періоду блокування розпізнавання після застосування обмеження частоти.

Проміжок часу у секундах, типове значення — 600 (10 хвилин).

Дивись також:

Обмеження частоти, `RATELIMIT_ATTEMPTS`, `RATELIMIT_WINDOW`

2.16.76 REGISTRATION_ALLOW_BACKENDS

Нове в версії 4.1.

Список модулів розпізнавання, з яких уможливлено реєстрацію. Обмежує лише нові реєстрації. Користувачі зможуть проходити розпізнавання та додавати розпізнавання за допомогою всіх налаштованих модулів розпізнавання.

Рекомендуємо не вимикати `REGISTRATION_OPEN`, а обмежити роботу модулів реєстрації. Якщо ви цього не зробите, користувачі зможуть реєструватися, але Weblate не показуватиме посилань на реєстрацію в інтерфейсі користувача.

Приклад:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

Підказка: Назви модулів відповідають назвам, використаним у адресі для розпізнавання.

Дивись також:

`REGISTRATION_OPEN`, *Розпізнавання*

2.16.77 REGISTRATION_CAPTCHA

Значенням може бути `True` або `False`. Значення вказує, чи буде реєстрацію нових облікових записів захищено за допомогою CAPTCHA. Цей параметр є необов'язковим. Типовим значенням є `True`. Це значення буде використано, якщо ви не вкажете якогось іншого.

Якщо увімкнено, на усі сторінки, де користувачі вводять адреси електронної пошти, буде додано CAPTCHA:

- Реєстрація нового облікового запису.
- Відновлення пароля.
- Додавання адреси електронної пошти до облікового запису.
- Контактна форма для користувачів, які не увійшли до системи.

2.16.78 REGISTRATION_EMAIL_MATCH

Нове в версії 2.17.

Надає вам змогу фільтрувати, які адреси електронної пошти можна реєструвати.

Типове значення — `.*`, що надає змогу реєструвати будь-яку адресу електронної пошти.

Ви можете скористатися ним для обмеження реєстрації єдиним доменом електронної пошти:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.79 REGISTRATION_OPEN

Визначає, чи дозволено реєстрацію нових облікових записів. Це додатковий параметр може лишатися у типовому значенні `True` або може бути перемкнутим на `False`.

Цей параметр впливає на вбудовані засоби розпізнавання за адресою електронної пошти або Python Social Auth (ви можете додати певні модулі до «білого» списку за допомогою параметра `REGISTRATION_ALLOW_BACKENDS`).

Примітка: Якщо використано сторонні способи розпізнавання, зокрема *Розпізнавання за допомогою LDAP*, це просто призводить до приховування реєстраційної форми, але нові користувачі попри це зможуть входити до системи і створювати облікові записи.

Дивись також:

`REGISTRATION_ALLOW_BACKENDS`, `REGISTRATION_EMAIL_MATCH`, *Розпізнавання*

2.16.80 REGISTRATION_REBIND

Нове в версії 4.16.

Уможливорює повторну прив'язку модулів розпізнавання для наявних користувачів. Увімкніть при перенесенні між різними надавачами засобів для розпізнавання.

Примітка: Типово, вимкнено, щоб заборонити додавання інших модулів розпізнавання до наявного облікового запису. Повторна прив'язка може призвести до компрометації облікового запису при використанні додаткових сторонніх модулів розпізнавання.

2.16.81 REPOSITORY_ALERT_THRESHOLD

Нове в версії 4.0.2.

Порогове значення для вмикання попередження щодо застарілих репозиторіїв, у яких міститься надто багато змін. Типове значення — 25.

Дивись також:

alerts

2.16.82 REQUIRE_LOGIN

Нове в версії 4.1.

Це увімкне `LOGIN_REQUIRED_URLS` та налаштовує набір бібліотек REST вимагати розпізнавання для всіх кінцевих точок програмного інтерфейсу.

Примітка: Це реалізовано у *Зразок налаштувань*. Для Docker скористайтеся `WEBLATE_REQUIRE_LOGIN`.

2.16.83 SENTRY_DSN

Нове в версії 3.9.

Sentry DSN, яким слід скористатися для *Збирання звітів щодо помилок*.

Дивись також:

Інтеграція Django із Sentry

2.16.84 SESSION_COOKIE_AGE_AUTHENTICATED

Нове в версії 4.3.

Встановити термін дії сеансу для автентифікованих користувачів. Це доповнює: параметр: `django:SESSION_COOKIE_AGE`, який застосовується для неавтентифікованих користувачів.

Дивись також:

`SESSION_COOKIE_AGE`

2.16.85 SIMPLIFY_LANGUAGES

Використовувати прості коди мов для типових комбінацій мова-країна. Наприклад, для перекладу `fr_FR` буде використано код мови `fr`. Зазвичай, така поведінка є бажаною, оскільки спрощує побудову списку мов для цих типових комбінацій.

Вимкніть, якщо вам потрібні різні переклади для кожного з варіантів.

2.16.86 SITE_DOMAIN

Налаштовує домен сайта. Домен потрібен для створення правильних абсолютних посилань у багатьох областях застосування (наприклад, для повідомлень активації, сповіщень та стрічок RSS).

Якщо Weblate запущено на нестандартному порту, допишіть його до вказаної адреси.

Приклади:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

Примітка: Цей параметр має містити лише назву домену. Для налаштованого протоколу (увімкнутого і примусово встановленого HTTPS) скористайтеся [ENABLE_HTTPS](#), а для зміни адреси скористайтеся [URL_PREFIX](#).

Підказка: У контейнері Docker домен сайта можна налаштувати за допомогою [WEBLATE_ALLOWED_HOSTS](#).

Дивись також:

faq-site, [Налаштовування дозволених вузлів](#), [Належне налаштовування HTTPS](#) [ENABLE_HTTPS](#), changesite, The “sites” framework

2.16.87 SITE_TITLE

Назва сайта, яку буде використано для сайта і надсилання повідомлень електронною поштою.

2.16.88 SPECIAL_CHARS

Додаткові символи, які слід включити на візуальну клавіатуру, [Візуальна клавіатура](#).

Типове значення:

```
SPECIAL_CHARS = ("\t", "\n", "\u00a0", "...")
```

2.16.89 SINGLE_PROJECT

Нове в версії 3.8.

Переспрямовує користувачів безпосередньо до проекту або складника замість показу панелі приладів. Ви можете встановити значення `True` — у цьому випадку це працюватиме, лише якщо на Weblate є лише один проект. У протилежному випадку встановіть ідентифікатор проекту, і він безумовно переспрямовуватиме до цього проекту.

Змінено в версії 3.11: У поточній версії цього параметра також приймається ідентифікатор проекту для примусового показу лише цього проекту.

Приклад:

```
SINGLE_PROJECT = "test"
```

2.16.90 SSH_EXTRA_ARGS

Нове в версії 4.9.

Уможливорює додавання нетипових параметрів, якщо Weblate викликає SSH. Це корисно, при встановленні з'єднання із серверами з використанням застарілого шифрування або іншими нестандартними можливостями.

Наприклад, при встановленні з'єднання SSH у Weblate з'являється повідомлення про помилку *Unable to negotiate with legacyhost: no matching key exchange method found. Their offer: diffie-hellman-group1-sha1*, ви можете увімкнути потрібні можливості за допомогою такого:

```
SSH_EXTRA_ARGS = "-oKexAlgorithms=+diffie-hellman-group1-sha1"
```

Підказка: Рядок буде оброблено командною оболонкою, тому слід взяти у лапки усі пробіли і спеціальні символи.

Дивись також:

[Застарілі параметри OpenSSH](#)

2.16.91 STATUS_URL

Адреса, на яку екземпляр Weblate повідомляє про свій стан.

2.16.92 SUGGESTION_CLEANUP_DAYS

Нове в версії 3.2.1.

Автоматично вилучає пропозицій після вказаної кількості днів. Типовим є значення `None`, тобто пропозиції не вилучаються.

2.16.93 UPDATE_LANGUAGES

Нове в версії 4.3.2.

Керує тим, чи слід оновлювати базу даних мови під час перенесення бази даних. Типово увімкнено. Цей параметр не впливає на підсумки виклику *setuplang*.

Попередження: З цим параметром показ мов може стати неоднорідним. Визначення мов Weblate з часом розширюють, і код мови для визначених мов не буде показано.

Дивись також:

[Вбудовані визначення мов](#)

2.16.94 URL_PREFIX

За допомогою цього параметра ви можете запускати Weblate з певним значенням шляху (якщо параметр не вказано, програма покладатиметься на те, що її запущено з кореневого каталогу вебсервера).

Примітка: Щоб скористатися цим параметром, вам також слід налаштувати ваш сервер на вилучення цього префікса. Наприклад, для WSGI вилучення можна досягти встановленням параметра `WSGIScriptAlias`.

Підказка: Префікс має розпочинатися з `/`.

Приклад:

```
URL_PREFIX = "/translations"
```

Примітка: Цей параметр не працює із вбудованим сервером Django. Вам доведеться скоригувати `urls.py` так, щоб у ньому містився цей префікс.

2.16.95 VCS_API_DELAY

Нове в версії 4.15.1.

Налаштовує мінімальну затримку у секундах для викликів стороннього програмного інтерфейсу у *Запити щодо злиття GitHub*, *Запити щодо об'єднання GitLab*, *Запити щодо злиття Gitea* і *Запити щодо об'єднання Pagure*.

Це обмежує частоту викликів програмного інтерфейсу з Weblate до цих служб, щоб уникнути перевантаження.

Якщо вашу роботу обмежує вторинний засіб обмеження частоти на GitHub, може допомогти збільшення цього значення.

Типове значення: 10.

2.16.96 VCS_BACKENDS

Налаштовування доступних модулів систем керування версіями.

Примітка: Weblate намагається скористатися усіма підтримуваними модулями, для яких є відповідні інструменти.

Підказка: За допомогою цього ви можете обмежити варіанти і додавати нетипові модулі систем керування версіями.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

Дивись також:

Налаштування інтеграції із керуванням версіями

2.16.97 VCS_CLONE_DEPTH

Нове в версії 3.10.2.

Визначає, наскільки глибоким має бути клонування репозиторіїв у Weblate.

Примітка: У поточній версії підтримку передбачено лише у *Git*. Типово, Weblate створює поверхневі клони репозиторіїв, щоб пришвидшити клонування і заощадити місце на диску. Залежно від ваших підходів до користування (наприклад, використання нетипових *Додатки*), у вас може виникнути потреба у збільшенні глибини або вимиканні поверхневих клонів повністю встановленням значення 0.

Підказка: Якщо при записі з Weblate ви отримуєте повідомлення `fatal: protocol error: expected old/new/ref, got 'shallow <commit hash>'`, вимкніть поверхневі клонування повністю:

```
VCS_CLONE_DEPTH = 0
```

2.16.98 WEBLATE_ADDONS

Список додатків, якими можна скористатися. Щоб скористатися ними, їх слід увімкнути для певного складника перекладу. Типово, до списку буде включено всі вбудовані додатки. За розширення списку вам, імовірно, слід зберегти наявні додатки ввімкненими. Приклад:

```
WEBLATE_ADDONS = (
    # Built-in add-ons
    "weblate.addons.gettext.GenerateMoAddon",
    "weblate.addons.gettext.UpdateLinguasAddon",
    "weblate.addons.gettext.UpdateConfigureAddon",
    "weblate.addons.gettext.MsgmergeAddon",
    "weblate.addons.gettext.GettextCustomizeAddon",
    "weblate.addons.gettext.GettextAuthorComments",
    "weblate.addons.cleanup.CleanupAddon",
    "weblate.addons.consistency.LanguageConsistencyAddon",
    "weblate.addons.discovery.DiscoveryAddon",
    "weblate.addons.flags.SourceEditAddon",
    "weblate.addons.flags.TargetEditAddon",
    "weblate.addons.flags.SameEditAddon",
    "weblate.addons.flags.BulkEditAddon",
    "weblate.addons.generate.GenerateFileAddon",
    "weblate.addons.json.JSONCustomizeAddon",
    "weblate.addons.xml.XMLCustomizeAddon",
    "weblate.addons.properties.PropertiesSortAddon",
    "weblate.addons.git.GitSquashAddon",
    "weblate.addons.removal.RemoveComments",
    "weblate.addons.removal.RemoveSuggestions",
    "weblate.addons.resx.ResxUpdateAddon",
    "weblate.addons.autotranslate.AutoTranslateAddon",
    "weblate.addons.yaml.YAMLCustomizeAddon",
    "weblate.addons.cdn.CDNJSAddon",
    # Add-on you want to include
    "weblate.addons.example.ExampleAddon",
)
```

Примітка: Вилучення додатка зі списку не призводить до видалення його зі складників. У цьому випадку стається збій Weblate. Вилучте додаток з усіх складників, перш ніж вилучати його зі списку.

Дивись також:

Додатки, `DEFAULT_ADDONS`

2.16.99 WEBLATE_EXPORTERS

Нове в версії 4.2.

Список доступних засобів експортування, які надають змогу отримувати переклади або глосарії у різноманітних форматах.

Дивись також:

Підтримувані формати файлів

2.16.100 WEBLATE_FORMATS

Нове в версії 3.0.

Список форматів файлів, якими ви можете скористатися.

Примітка: У типовому списку вже є загальноновживані формати.

Дивись також:

Підтримувані формати файлів

2.16.101 WEBLATE_MACHINERY

Нове в версії 4.13.

Список служб комп'ютерного перекладу, якими можна скористатися.

Дивись також:

Налаштування автоматичних пропозицій

2.16.102 WEBLATE_GPG_IDENTITY

Нове в версії 3.1.

Профіль, який буде використано Weblate для підписування Git. Приклад:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

Програма виконає пошук у репозиторії ключів GPG Weblate відповідного ключа (`home/.gnupg` у `DATA_DIR`). Якщо ключа не буде знайдено, ключ буде створено. Будь ласка, ознайомтеся із розділом *Підписування внесків Git за допомогою GnuPG*, щоб дізнатися більше.

Дивись також:

Підписування внесків Git за допомогою GnuPG

2.16.103 WEBSITE_REQUIRED

Визначає, чи має бути вказано *Вебсайт проекту* при створенні проекту. Типово увімкнено, оскільки це відповідає налаштуванням загальнодоступних серверів.

2.17 Зразок налаштувань

Наведений нижче приклад постачається як `weblate/settings_example.py` разом із Weblate:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

import os
import platform
from logging.handlers import SysLogHandler

# Title of site to use
SITE_TITLE = "Weblate"

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

#
# Django settings for Weblate project.
#

DEBUG = True

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password.
        "PASSWORD": "",
        # Set to empty string for localhost.
        "HOST": "127.0.0.1",
        # Set to empty string for default.
        "PORT": "",
        # Customizations for databases.
        "OPTIONS": {
            # In case of using an older MySQL server,
            # which has MyISAM as a default storage
            # "init_command": "SET storage_engine=INNODB",

```

(continues on next page)

(continued from previous page)

```

        # Uncomment for MySQL older than 5.7:
        # "init_command": "SET sql_mode='STRICT_TRANS_TABLES'",
        # Set emoji capable charset for MySQL:
        # "charset": "utf8mb4",
        # Change connection timeout in case you get MySQL gone away error:
        # "connect_timeout": 28800,
    },
    # Persistent connections
    "CONN_MAX_AGE": 0,
    # Disable server-side cursors, might be needed with pgbouncer
    "DISABLE_SERVER_SIDE_CURSORS": False,
}
}

# Data directory, you can use following for the development purposes:
# os.path.join(os.path.dirname(os.path.dirname(os.path.abspath(__file__))), "data")
DATA_DIR = "/home/weblate/data"
CACHE_DIR = f"{DATA_DIR}/cache"

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name
# although not all choices may be available on all operating systems.
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = "UTC"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

LANGUAGES = (
    ("ar", "العربية"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@latin", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("da", "Dansk"),
    ("de", "Deutsch"),
    ("en", "English"),
    ("el", "Ελληνικά"),
    ("en-gb", "English (United Kingdom)"),
    ("es", "Español"),
    ("fi", "Suomi"),
    ("fr", "Français"),
    ("gl", "Galego"),
    ("he", "עברית"),
    ("hu", "Magyar"),
    ("hr", "Hrvatski"),
    ("id", "Indonesia"),
    ("is", "Íslenska"),
    ("it", "Italiano"),
    ("ja", "日本語"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),

```

(continues on next page)

(continued from previous page)

```

    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("th", "ไทย"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "中文"),
    ("zh-hant", "繁體中文"),
)

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
USE_L10N = True

# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True

# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(CACHE_DIR, "static")

# URL prefix for static files.
STATIC_URL = f"{URL_PREFIX}/static/"

# Additional locations of static files
STATICFILES_DIRS = (
    # Put strings here, like "/home/html/static" or "C:/www/django/static".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    "django.contrib.staticfiles.finders.FileSystemFinder",

```

(continues on next page)

(continued from previous page)

```

    "django.contrib.staticfiles.finders.AppDirectoriesFinder",
    "compressor.finders.CompressorFinder",
)

# Make this unique, and don't share it with anybody.
# You can generate it using weblate-generate-secret-key
SECRET_KEY = ""

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
        },
        "APP_DIRS": True,
    }
]

# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_CREDENTIALS = {}

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_CREDENTIALS = {}

# Bitbucket username and token for sending merge requests.
# Please see the documentation for more details.
BITBUCKETSERVER_CREDENTIALS = {}

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth2",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

SOCIAL_AUTH_GITHUB_ORG_KEY = ""

```

(continues on next page)

(continued from previous page)

```

SOCIAL_AUTH_GITHUB_ORG_SECRET = ""
SOCIAL_AUTH_GITHUB_ORG_NAME = ""

SOCIAL_AUTH_GITHUB_TEAM_KEY = ""
SOCIAL_AUTH_GITHUB_TEAM_SECRET = ""
SOCIAL_AUTH_GITHUB_TEAM_ID = ""

SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (
    "social_core.pipeline.social_auth.social_details",
    "social_core.pipeline.social_auth.social_uid",
    "social_core.pipeline.social_auth.auth_allowed",
    "social_core.pipeline.social_auth.social_user",
    "weblate.accounts.pipeline.store_params",
    "weblate.accounts.pipeline.verify_open",
    "social_core.pipeline.user.get_username",
    "weblate.accounts.pipeline.require_email",
    "social_core.pipeline.mail.mail_validation",
    "weblate.accounts.pipeline.revoke_mail_code",
    "weblate.accounts.pipeline.ensure_valid",
    "weblate.accounts.pipeline.remove_account",
    "social_core.pipeline.social_auth.associate_by_email",
    "weblate.accounts.pipeline.reauthenticate",
    "weblate.accounts.pipeline.verify_username",
    "social_core.pipeline.user.create_user",
    "social_core.pipeline.social_auth.associate_user",
    "social_core.pipeline.social_auth.load_extra_data",
    "weblate.accounts.pipeline.cleanup_next",
    "weblate.accounts.pipeline.user_full_name",
    "weblate.accounts.pipeline.store_email",
    "weblate.accounts.pipeline.notify_connect",
    "weblate.accounts.pipeline.password_reset",
)
SOCIAL_AUTH_DISCONNECT_PIPELINE = (
    "social_core.pipeline.disconnect.allowed_to_disconnect",
    "social_core.pipeline.disconnect.get_entries",
    "social_core.pipeline.disconnect.revoke_tokens",
    "weblate.accounts.pipeline.cycle_session",
    "weblate.accounts.pipeline.adjust_primary_mail",
    "weblate.accounts.pipeline.notify_disconnect",
    "social_core.pipeline.disconnect.disconnect",
    "weblate.accounts.pipeline.cleanup_next",
)

# Custom authentication strategy
SOCIAL_AUTH_STRATEGY = "weblate.accounts.strategy.WeblateStrategy"

# Raise exceptions so that we can handle them later
SOCIAL_AUTH_RAISE_EXCEPTIONS = True

```

(continues on next page)

(continued from previous page)

```

SOCIAL_AUTH_EMAIL_VALIDATION_FUNCTION = "weblate.accounts.pipeline.send_validation"
SOCIAL_AUTH_EMAIL_VALIDATION_URL = f"{URL_PREFIX}/accounts/email-sent/"
SOCIAL_AUTH_LOGIN_ERROR_URL = f"{URL_PREFIX}/accounts/login/"
SOCIAL_AUTH_EMAIL_FORM_URL = f"{URL_PREFIX}/accounts/email/"
SOCIAL_AUTH_NEW_ASSOCIATION_REDIRECT_URL = f"{URL_PREFIX}/accounts/profile/#account
↪"
SOCIAL_AUTH_PROTECTED_USER_FIELDS = ("email",)
SOCIAL_AUTH_SLUGIFY_USERNAMES = True
SOCIAL_AUTH_SLUGIFY_FUNCTION = "weblate.accounts.pipeline.slugify_username"

# Password validation configuration
AUTH_PASSWORD_VALIDATORS = [
    {
        "NAME": "django.contrib.auth.password_validation.
↪UserAttributeSimilarityValidator" # noqa: E501, pylint: disable=line-too-long
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.CharsPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.PastPasswordsValidator"},
    # Optional password strength validation by django-zxcvbn-password
    # {
    #     "NAME": "zxcvbn_password.ZXCVBNValidator",
    #     "OPTIONS": {
    #         "min_score": 3,
    #         "user_attributes": ("username", "email", "full_name")
    #     }
    # },
]

# Password hashing (prefer Argon)
PASSWORD_HASHERS = [
    "django.contrib.auth.hashers.Argon2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher",
    "django.contrib.auth.hashers.BCryptSHA256PasswordHasher",
]

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "corsheaders.middleware.CorsMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
    "social_django.middleware.SocialAuthExceptionMiddleware",

```

(continues on next page)

(continued from previous page)

```

    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
    "weblate.wladmin.middleware.ManageMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",
    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
    "weblate.gitexport",
    # Standard Django modules
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "django.contrib.admin.apps.SimpleAdminConfig",
    "django.contrib.admindocs",
    "django.contrib.sitemaps",
    "django.contrib.humanize",
    # Third party Django modules
    "social_django",
    "crispy_forms",
    "crispy_bootstrap3",
    "compressor",
    "rest_framework",
    "rest_framework.authtoken",
    "django_filters",
    "django_celery_beat",
    "corsheaders",
]

# Custom exception reporter to include some details
DEFAULT_EXCEPTION_REPORTER_FILTER = "weblate.trans.debug.
↳ WeblateExceptionReporterFilter"

# Default logging of Weblate messages
# - to syslog in production (if available)
# - otherwise to console

```

(continues on next page)

(continued from previous page)

```

# - you can also choose "logfile" to log into separate file
#   after configuring it below

# Detect if we can connect to syslog
HAVE_SYSLOG = False
if platform.system() != "Windows":
    try:
        handler = SysLogHandler(address="/dev/log", facility=SysLogHandler.LOG_
↪LOCAL2)
        handler.close()
        HAVE_SYSLOG = True
    except OSError:
        HAVE_SYSLOG = False

DEFAULT_LOG = "console" if DEBUG or not HAVE_SYSLOG else "syslog"
DEFAULT_LOGLEVEL = "DEBUG" if DEBUG else "INFO"

# A sample logging configuration. The only tangible logging
# performed by this configuration is to send an email to
# the site admins on every HTTP 500 error when DEBUG=False.
# See http://docs.djangoproject.com/en/stable/topics/logging for
# more details on how to customize your logging configuration.
LOGGING = {
    "version": 1,
    "disable_existing_loggers": True,
    "filters": {"require_debug_false": {"()": "django.utils.log.RequireDebugFalse"}
↪},
    "formatters": {
        "syslog": {"format": "weblate[%(process)d]: %(levelname)s %(message)s"},
        "simple": {"format": "[% (asctime)s: %(levelname)s/%(process)s] %(message)s
↪"},
        "logfile": {"format": "%(asctime)s %(levelname)s %(message)s"},
        "django.server": {
            "()": "django.utils.log.ServerFormatter",
            "format": "[% (server_time)s] %(message)s",
        },
    },
    "handlers": {
        "mail_admins": {
            "level": "ERROR",
            "filters": ["require_debug_false"],
            "class": "django.utils.log.AdminEmailHandler",
            "include_html": True,
        },
        "console": {
            "level": "DEBUG",
            "class": "logging.StreamHandler",
            "formatter": "simple",
        },
        "django.server": {
            "level": "INFO",
            "class": "logging.StreamHandler",
            "formatter": "django.server",
        },
        "syslog": {
            "level": "DEBUG",
            "class": "logging.handlers.SysLogHandler",
            "formatter": "syslog",
            "address": "/dev/log",
            "facility": SysLogHandler.LOG_LOCAL2,
        },
    },
}

```

(continues on next page)

(continued from previous page)

```

    # Logging to a file
    # "logfile": {
    #     "level": "DEBUG",
    #     "class": "logging.handlers.RotatingFileHandler",
    #     "filename": "/var/log/weblate/weblate.log",
    #     "maxBytes": 100000,
    #     "backupCount": 3,
    #     "formatter": "logfile",
    # },
},
"loggers": {
    "django.request": {
        "handlers": ["mail_admins", DEFAULT_LOG],
        "level": "ERROR",
        "propagate": True,
    },
    "django.server": {
        "handlers": ["django.server"],
        "level": "INFO",
        "propagate": False,
    },
    # Logging database queries
    # "django.db.backends": {
    #     "handlers": [DEFAULT_LOG],
    #     "level": "DEBUG",
    # },
    "weblate": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Logging VCS operations
    "weblate.vcs": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Python Social Auth
    "social": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Django Authentication Using LDAP
    "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # SAML IdP
    "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
},
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    # "weblate.machinery.apertium.ApertiumAPYTranslation",
    # "weblate.machinery.baidu.BaiduTranslation",
    # "weblate.machinery.deepl.DeepLTranslation",
    # "weblate.machinery.glosbe.GlosbeTranslation",
    # "weblate.machinery.google.GoogleTranslation",
    # "weblate.machinery.googlev3.GoogleV3Translation",
    # "weblate.machinery.libretranslate.LibreTranslateTranslation",
    # "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    # "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    # "weblate.machinery.modernmt.ModernMTTranslation",
    # "weblate.machinery.mymemory.MyMemoryTranslation",
    # "weblate.machinery.netease.NeteaseSightTranslation",
    # "weblate.machinery.tmsserver.AmagamaTranslation",
    # "weblate.machinery.tmsserver.TMServerTranslation",
    # "weblate.machinery.yandex.YandexTranslation",
    # "weblate.machinery.saptranslationhub.SAPTranslationHub",
    # "weblate.machinery.youdao.YoudaoTranslation",

```

(continues on next page)

(continued from previous page)

```

    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APy server
MT_APERTIUM_APY = None

# DeepL API key
MT_DEEPL_KEY = None

# LibreTranslate
MT_LIBRETRANSLATE_API_URL = None
MT_LIBRETRANSLATE_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

# ModernMT
MT_MODERNMT_KEY = None

# MyMemory identification email, see
# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
MT_GOOGLE_CREDENTIALS = None
MT_GOOGLE_PROJECT = None

# Baidu app key and secret
MT_BAIDU_ID = None
MT_BAIDU_SECRET = None

# Youdao Zhiyun app key and secret
MT_YOUDAO_ID = None
MT_YOUDAO_SECRET = None

# Netease Sight (Jianwai) app key and secret
MT_NETEASE_KEY = None
MT_NETEASE_SECRET = None

# API key for Yandex Translate API
MT_YANDEX_KEY = None

# tmserver URL
MT_TMSERVER = None

# SAP Translation Hub
MT_SAP_BASE_URL = None
MT_SAP_SANDBOX_APIKEY = None
MT_SAP_USERNAME = None

```

(continues on next page)

(continued from previous page)

```

MT_SAP_PASSWORD = None
MT_SAP_USE_MT = True

# Use HTTPS when creating redirect URLs for social authentication, see
# documentation for more details:
# https://python-social-auth-docs.readthedocs.io/en/latest/configuration/settings.
# ↪html#processing-redirects-and-urlopen
SOCIAL_AUTH_REDIRECT_IS_HTTPS = ENABLE_HTTPS

# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view
CSRF_FAILURE_VIEW = "weblate.trans.views.error.csrf_failure"
SESSION_COOKIE_SECURE = ENABLE_HTTPS
SESSION_COOKIE_HTTPONLY = True
# SSL redirect
SECURE_SSL_REDIRECT = ENABLE_HTTPS
SECURE_SSL_HOST = SITE_DOMAIN
# Sent referrrrer only for same origin links
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",) # Allowing HTTP access to health check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
SESSION_COOKIE_SAMESITE = "Lax"
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000
# Allow more fields for case with a lot of subscriptions in profile
DATA_UPLOAD_MAX_NUMBER_FIELDS = 2000

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login

```

(continues on next page)

(continued from previous page)

```

LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer
# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
SIMPLIFY_LANGUAGES = True

# Render forms using bootstrap
CRISPY_ALLOWED_TEMPLATE_PACKS = "bootstrap3"
CRISPY_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
# CHECK_LIST = (
#     "weblate.checks.same.SameCheck",
#     "weblate.checks.chars.BeginNewlineCheck",
#     "weblate.checks.chars.EndNewlineCheck",
#     "weblate.checks.chars.BeginSpaceCheck",
#     "weblate.checks.chars.EndSpaceCheck",
#     "weblate.checks.chars.DoubleSpaceCheck",
#     "weblate.checks.chars.EndStopCheck",
#     "weblate.checks.chars.EndColonCheck",
#     "weblate.checks.chars.EndQuestionCheck",
#     "weblate.checks.chars.EndExclamationCheck",
#     "weblate.checks.chars.EndEllipsisCheck",
#     "weblate.checks.chars.EndSemicolonCheck",
#     "weblate.checks.chars.MaxLengthCheck",
#     "weblate.checks.chars.KashidaCheck",
#     "weblate.checks.chars.PunctuationSpacingCheck",
#     "weblate.checks.format.PythonFormatCheck",
#     "weblate.checks.format.PythonBraceFormatCheck",
#     "weblate.checks.format.PHPFormatCheck",
#     "weblate.checks.format.CFormatCheck",
#     "weblate.checks.format.PperlFormatCheck",
#     "weblate.checks.format.JavaScriptFormatCheck",
#     "weblate.checks.format.LuaFormatCheck",
#     "weblate.checks.format.ObjectPascalFormatCheck",
#     "weblate.checks.format.SchemeFormatCheck",
#     "weblate.checks.format.CSharpFormatCheck",
#     "weblate.checks.format.JavaFormatCheck",
#     "weblate.checks.format.JavaMessageFormatCheck",
#     "weblate.checks.format.PercentPlaceholdersCheck",

```

(continues on next page)

(continued from previous page)

```

# "weblate.checks.format.VueFormattingCheck",
# "weblate.checks.format.I18NextInterpolationCheck",
# "weblate.checks.format.ESTemplateLiteralsCheck",
# "weblate.checks.angularjs.AngularJSInterpolationCheck",
# "weblate.checks.icu.ICUMessageFormatCheck",
# "weblate.checks.icu.ICUSourceCheck",
# "weblate.checks.qt.QtFormatCheck",
# "weblate.checks.qt.QtPluralCheck",
# "weblate.checks.ruby.RubyFormatCheck",
# "weblate.checks.consistency.PluralsCheck",
# "weblate.checks.consistency.SamePluralsCheck",
# "weblate.checks.consistency.ConsistencyCheck",
# "weblate.checks.consistency.TranslatedCheck",
# "weblate.checks.chars.EscapedNewlineCountingCheck",
# "weblate.checks.chars.NewLineCountCheck",
# "weblate.checks.markup.BBCodeCheck",
# "weblate.checks.chars.ZeroWidthSpaceCheck",
# "weblate.checks.render.MaxSizeCheck",
# "weblate.checks.markup.XMLValidityCheck",
# "weblate.checks.markup.XMLTagsCheck",
# "weblate.checks.markup.MarkdownRefLinkCheck",
# "weblate.checks.markup.MarkdownLinkCheck",
# "weblate.checks.markup.MarkdownSyntaxCheck",
# "weblate.checks.markup.URLCheck",
# "weblate.checks.markup.SafeHTMLCheck",
# "weblate.checks.placeholders.PlaceholderCheck",
# "weblate.checks.placeholders.RegexCheck",
# "weblate.checks.duplicate.DuplicateCheck",
# "weblate.checks.source.OptionalPluralCheck",
# "weblate.checks.source.EllipsisCheck",
# "weblate.checks.source.MultipleFailingCheck",
# "weblate.checks.source.LongUntranslatedCheck",
# "weblate.checks.format.MultipleUnnamedFormatsCheck",
# "weblate.checks.glossary.GlossaryCheck",
# )

# List of automatic fixups
# AUTOFIX_LIST = (
#     "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
#     "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
#     "weblate.trans.autofixes.chars.RemoveZeroSpace",
#     "weblate.trans.autofixes.chars.RemoveControlChars",
# )

# List of enabled addons
# WEBLATE_ADDONS = (
#     "weblate.addons.gettext.GenerateMoAddon",
#     "weblate.addons.gettext.UpdateLinguasAddon",
#     "weblate.addons.gettext.UpdateConfigureAddon",
#     "weblate.addons.gettext.MsgmergeAddon",
#     "weblate.addons.gettext.GettextCustomizeAddon",
#     "weblate.addons.gettext.GettextAuthorComments",
#     "weblate.addons.cleanup.CleanupAddon",
#     "weblate.addons.cleanup.RemoveBlankAddon",
#     "weblate.addons.consistency.LanguaugeConsistencyAddon",
#     "weblate.addons.discovery.DiscoveryAddon",
#     "weblate.addons.autotranslate.AutoTranslateAddon",
#     "weblate.addons.flags.SourceEditAddon",
#     "weblate.addons.flags.TargetEditAddon",
#     "weblate.addons.flags.SameEditAddon",
#     "weblate.addons.flags.BulkEditAddon",

```

(continues on next page)

(continued from previous page)

```

# "weblate.addons.generate.GenerateFileAddon",
# "weblate.addons.generate.PseudolocaleAddon",
# "weblate.addons.generate.PrefillAddon",
# "weblate.addons.json.JSONCustomizeAddon",
# "weblate.addons.xml.XMLCustomizeAddon",
# "weblate.addons.properties.PropertiesSortAddon",
# "weblate.addons.git.GitSquashAddon",
# "weblate.addons.removal.RemoveComments",
# "weblate.addons.removal.RemoveSuggestions",
# "weblate.addons.resx.ResxUpdateAddon",
# "weblate.addons.yaml.YAMLCustomizeAddon",
# "weblate.addons.cdn.CDNJSAddon",
# )

# E-mail address that error messages come from.
SERVER_EMAIL = "noreply@example.com"

# Default email address to use for various automated correspondence from
# the site managers. Used for registration emails.
DEFAULT_FROM_EMAIL = "noreply@example.com"

# List of URLs your site is supposed to serve
ALLOWED_HOSTS = ["*"]

# Configuration for caching
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/1",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # "LOCATION": "unix:///var/run/redis/redis.sock?db=1",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
            # If you set password here, adjust CELERY_BROKER_URL as well
            "PASSWORD": None,
            "CONNECTION_POOL_KWARGS": {},
        },
        "KEY_PREFIX": "weblate",
        "TIMEOUT": 3600,
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(CACHE_DIR, "avatar"),
        "TIMEOUT": 86400,
        "OPTIONS": {"MAX_ENTRIES": 1000},
    },
}

# Store sessions in cache
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
# Store messages in session
MESSAGE_STORAGE = "django.contrib.messages.storage.session.SessionStorage"

# REST framework settings for API
REST_FRAMEWORK = {
    # Use Django's standard `django.contrib.auth` permissions,
    # or allow read-only access for unauthenticated users.
    "DEFAULT_PERMISSION_CLASSES": [
        # Require authentication for login required sites

```

(continues on next page)

(continued from previous page)

```

        "rest_framework.permissions.IsAuthenticated"
        if REQUIRE_LOGIN
        else "rest_framework.permissions.IsAuthenticatedOrReadOnly"
    ],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework.authentication.TokenAuthentication",
        "weblate.api.authentication.BearerAuthentication",
        "rest_framework.authentication.SessionAuthentication",
    ),
    "DEFAULT_THROTTLE_CLASSES": (
        "weblate.api.throttling.UserRateThrottle",
        "weblate.api.throttling.AnonRateThrottle",
    ),
    "DEFAULT_THROTTLE_RATES": {"anon": "100/day", "user": "5000/hour"},
    "DEFAULT_PAGINATION_CLASS": "weblate.api.pagination.StandardPagination",
    "PAGE_SIZE": 50,
    "VIEW_DESCRIPTION_FUNCTION": "weblate.api.views.get_view_description",
    "UNAUTHENTICATED_USER": "weblate.auth.models.get_anonymous",
}

# Fonts CDN URL
FONTS_CDN_URL = None

# Django compressor offline mode
COMPRESS_OFFLINE = False
COMPRESS_OFFLINE_CONTEXT = [
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI": ↪
↪ True},
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI": ↪
↪ False},
]

# Require login for all URLs
if REQUIRE_LOGIN:
    LOGIN_REQUIRED_URLS = (r"/(.*)$",)

# In such case you will want to include some of the exceptions
# LOGIN_REQUIRED_URLS_EXCEPTIONS = (
#     rf"{URL_PREFIX}/accounts/(.*)$", # Required for login
#     rf"{URL_PREFIX}/admin/login/(.*)$", # Required for admin login
#     rf"{URL_PREFIX}/static/(.*)$", # Required for development mode
#     rf"{URL_PREFIX}/widgets/(.*)$", # Allowing public access to widgets
#     rf"{URL_PREFIX}/data/(.*)$", # Allowing public access to data exports
#     rf"{URL_PREFIX}/hooks/(.*)$", # Allowing public access to notification hooks
#     rf"{URL_PREFIX}/healthz/$", # Allowing public access to health check
#     rf"{URL_PREFIX}/api/(.*)$", # Allowing access to API
#     rf"{URL_PREFIX}/js/i18n/$", # JavaScript localization
#     rf"{URL_PREFIX}/contact/$", # Optional for contact form
#     rf"{URL_PREFIX}/legal/(.*)$", # Optional for legal app
#     rf"{URL_PREFIX}/avatar/(.*)$", # Optional for avatars
# )

# Silence some of the Django system checks
SILENCED_SYSTEM_CHECKS = [
    # We have modified django.contrib.auth.middleware.AuthenticationMiddleware
    # as weblate.accounts.middleware.AuthenticationMiddleware
    "admin.E408"
]

# Celery worker configuration for testing
# CELERY_TASK_ALWAYS_EAGER = True

```

(continues on next page)

(continued from previous page)

```

# CELERY_BROKER_URL = "memory://"
# CELERY_TASK_EAGER_PROPAGATES = True
# Celery worker configuration for production
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

# Celery settings, it is not recommended to change these
CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_SCHEDULER = "django_celery_beat.schedulers:DatabaseScheduler"
CELERY_TASK_ROUTES = {
    "weblate.trans.tasks.auto_translate*": {"queue": "translate"},
    "weblate.accounts.tasks.notify*": {"queue": "notify"},
    "weblate.accounts.tasks.send_mails": {"queue": "notify"},
    "weblate.utils.tasks.settings_backup": {"queue": "backup"},
    "weblate.utils.tasks.database_backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup_service": {"queue": "backup"},
    "weblate.memory.tasks.*": {"queue": "memory"},
}

# CORS allowed origins
CORS_ALLOWED_ORIGINS = []
CORS_URLS_REGEX = r"^/api/.*$"

# Enable plain database backups
DATABASE_BACKUP = "plain"

# Enable auto updating
AUTO_UPDATE = False

# PGP commits signing
WEBLATE_GPG_IDENTITY = None

# Third party services integration
MATOMO_SITE_ID = None
MATOMO_URL = None
GOOGLE_ANALYTICS_ID = None
SENTRY_DSN = None
SENTRY_ENVIRONMENT = SITE_DOMAIN
AKISMET_API_KEY = None

```

2.18 Команди керування

Примітка: Запуск команд керування від імені користувача, відмінного від користувача, від імені якого запущено вебсервер, може призвести до встановлення помилкових прав доступу до файлів. Будь ласка, ознайомтеся із розділом *Права доступу у файловій системі*, щоб дізнатися більше.

Тут наведено базові команди керування (доступні як `./manage.py` у початкових кодах Django або як розширений набір у скрипті із назвою **weblate**, який встановлюється над Weblate).

2.18.1 Виклик команд керування

Як ми вже згадували раніше, виклик залежить від того, як убуло встановлено Weblate.

Якщо використовувати для Weblate віртуальне середовище, ви можете або вказати повний шлях до **weblate**, або активувати віртуальне середовище до виклику команди:

```
# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

Якщо ви використовуєте початковий код безпосередньо (з розпакованого архіву або з копії репозиторію Git), скриптом керування буде файл `./manage.py` з початкових кодів Weblate. Щоб запустити його, виконайте такі дії:

```
python ./manage.py list_versions
```

Якщо ви встановили Weblate за допомогою засобу встановлення `pip` або за допомогою скрипту `./setup.py`, програму **weblate** встановлено до основного набору шляхів вашої системи (або шляху віртуального середовища). Ви можете скористатися нею для керування Weblate:

```
weblate list_versions
```

Для випадку образу `Docker` скрипт буде встановлено подібно до інших варіантів і ви зможете запустити його за допомогою **docker exec**:

```
docker exec --user weblate <container> weblate list_versions
```

Для **docker-compose** процедура є подібною, вам лише треба скористатися **docker-compose exec**:

```
docker-compose exec --user weblate weblate weblate list_versions
```

Якщо вам треба передати програмі файл, ви можете тимчасово додати том:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /tmp/
↪users.json
```

Дивись також:

Установлення за допомогою Docker, Установлення на Debian і Ubuntu, Установлення у SUSE та openSUSE, Установлення у RedHat, Fedora і CentOS, Установлення з початкового коду

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

Нове в версії 2.5.

Імпортує переклад з файла для використання його як пропозиції для вказаного перекладу. Дублікати перекладів буде пропущено, додано буде лише різні переклади.

--author USER@EXAMPLE.COM

Адреса електронної пошти автора пропозиції. Запис цього користувача має вже створено до імпортування (ви можете створити його за допомогою адміністративного інтерфейсу, якщо потрібно).

Приклад:

```
weblate --author michal@cihar.com add_suggestions weblate application cs /tmp/
↪ suggestions-cs.po
```

2.18.3 auto_translate

weblate auto_translate <project> <component> <language>

Нове в версії 2.5.

Змінено в версії 4.6: Додано параметр для режиму перекладу.

Виконує автоматичний переклад на основі перекладів інших складників.

--source PROJECT/COMPONENT

Вказує складник, який слід використати як доступне джерело для перекладу. Якщо не вказано, буде використано усі складники проекту.

--user USERNAME

Визначає ім'я користувача, яке буде внесено до списку авторів перекладів. Якщо не вказано, буде використано запис «Anonymous user».

--overwrite

Чи переписувати наявні переклади.

--inconsistent

Чи переписувати наявні переклади, які є несумісними із попередніми (див. [Неузгодженість](#)).

--add

Автоматично додавати мову, якщо вказаного перекладу не існує.

--mt MT

Використовувати комп'ютерний переклад замість інших складників комп'ютерних перекладів.

--threshold THRESHOLD

Поріг подібності для комп'ютерного перекладу. Типовим значенням є 80.

--mode MODE

Вказати режим перекладу. Типовим є `translate`, але можна скористатися `fuzzy` або `suggest`.

Приклад:

```
weblate auto_translate --user nijel --inconsistent --source weblate/application_
↪ weblate website cs
```

Дивись також:

[Автоматичний переклад](#)

2.18.4 celery_queues

weblate celery_queues

Нове в версії 3.7.

Показує довжину черг завдань Celery.

2.18.5 checkgit

weblate checkgit <project|project/component>

Виводить поточний стан репозиторію модуля Git.

Ви можете або визначити проєкт або складник для оновлення (наприклад, weblate/application), або скористатися `--all` для оновлення усіх наявних складників.

2.18.6 commitgit

weblate commitgit <project|project/component>

Надсилає усі можливі зміни з черги до репозиторію Git модуля обробки.

Ви можете або визначити проєкт або складник для оновлення (наприклад, weblate/application), або скористатися `--all` для оновлення усіх наявних складників або скористайтеся `--file-format` для фільтрування на основі формату файлів.

2.18.7 commit_pending

weblate commit_pending <project|project/component>

Вносить зміни з черги, вік яких перевищує вказаний.

Ви можете або визначити проєкт або складник для оновлення (наприклад, weblate/application), або скористатися `--all` для оновлення усіх наявних складників.

--age HOURS

Вік у годинах для внесків. Якщо не вказано, буде використано значення, яке налаштовано у *Налаштування складників*.

Примітка: Це завдання автоматично виконується Weblate у фоновому режимі, отже, немає сенсу викликати його вручну, окрім примусового запису внеску, який передуватиме визначеному *складником*.

Дивись також:

Запуск завдань щодо супроводу, COMMIT_PENDING_HOURS

2.18.8 cleanuptrans

weblate cleanuptrans

Очищує список осиротілих перевірок та пропозицій перекладів. Потреби у такому очищенні вручну, зазвичай, немає, оскільки очищення відбувається автоматично у фоновому режимі.

Дивись також:

Запуск завдань щодо супроводу

2.18.9 cleanup_ssh_keys

weblate cleanup_ssh_keys

Нове в версії 4.9.1.

Виконує очищення збережених SSH-ключів хосту:

- Вимикає застарілі ключі RSA до GitHub. Такі ключі можуть спричиняти проблеми під час встановлення з'єднання із GitHub.
- Вилучає повторювані записи в ключах хосту.

Дивись також:

Репозиторії із доступом за SSH

2.18.10 createadmin

weblate createadmin

Створює обліковий запис `admin` із випадковим паролем, якщо пароль не вказано явним чином.

--password PASSWORD

Надає пароль у командному рядку, щоб не створювати його випадковим чином.

--no-password

Не встановлювати пароль. Може бути корисним із поєднанням `-update`.

--username USERNAME

Використовувати вказане ім'я замість `admin`.

--email USER@EXAMPLE.COM

Вказати адресу електронної пошти адміністратора.

--name

Вказати ім'я адміністратора (видиме).

--update

Оновити запис наявного користувача (ви можете скористатися цим для зміни паролів).

Змінено в версії 2.9: Додані параметри `--username`, `--email`, `--name` і `--update`.

2.18.11 dump_memory

weblate dump_memory

Нове в версії 2.20.

Експортувати файл JSON, який міститиме вміст пам'яті перекладів Weblate.

Дивись також:

Пам'ять перекладів, Схема пам'яті перекладів Weblate

2.18.12 dumpuserdata

```
weblate dumpuserdata <file.json>
```

Створити дамپ даних користувачів у файлі для наступного використання у *importuserdata*.

Підказка: Ця можливість стане у пригоді, якщо ви вирішите перенести або злити екземпляри Weblate.

2.18.13 import_demo

```
weblate import_demo
```

Нове в версії 4.1.

Створює демонстраційний проєкт зі складниками на основі <<https://github.com/WeblateOrg/demo>>. Перш ніж віддавати цю команду, переконайтеся, що завдання celery запущено.

Ця можливість може бути корисною для розробників Weblate.

2.18.14 import_json

```
weblate import_json <json-file>
```

Нове в версії 2.7.

Пакетне імпортування складників на основі даних JSON.

Структура імпортованого файла JSON дуже подібна до об'єкта складника (див. *GET /api/components/(string:project)/(string:component)/*). Вам слід включити поля name і filemask.

--project PROJECT

Вказує, звідки буде імпортовано складники.

--main-component COMPONENT

Використовувати вказаний репозиторій системи керування версіями з цього складника для всіх складників.

--ignore

Пропустити (вже) імпортовані складники.

--update

Оновити (вже) імпортовані складники.

Змінено в версії 2.9: Параметри **--ignore** і **--update** тут для обробки вже імпортованих складників.

Приклад файла JSON:

```
[
  {
    "slug": "po",
    "name": "Gettext PO",
    "file_format": "po",
    "filemask": "po/*.po",
    "new_lang": "none"
  },
  {
    "name": "Android",
    "filemask": "android/values-*/strings.xml",
    "template": "android/values/strings.xml",
```

(continues on next page)

(continued from previous page)

```

    "repo": "weblate://test/test",
    "file_format": "aresource"
  }
]

```

Дивись також:*import_memory*

2.18.15 import_memory

weblate import_memory <file>

Нове в версії 2.20.

Імпортує файл TMX або JSON до пам'яті перекладів Weblate.

--language-map LANGMAP

Надає змогу встановлювати прив'язки мов у TMX до пам'яті перекладів Weblate. Прив'язка кодів мов встановлюється після нормалізації, яку зазвичай виконує Weblate.

Наприклад, `--language-map en_US:en` імпортує усі рядки `en_US` як рядки `en`.

Це може бути корисним у випадку, якщо ваші локалі файлів TMX не збігаються із локалями, які використовуються у Weblate.

Дивись також:*Пам'ять перекладів, Схема пам'яті перекладів Weblate*

2.18.16 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

Змінено в версії 3.0: Роботу команди `import_project` у поточній версії засновано на додатку *Виявлення складників*. Це призвело до деяких змін у поведінці і параметрах, які приймає команда.

Виконує пакетне імпортування складників до проекту на основі маски файлів.

`<project>` є назвою наявного проекту, до якого буде імпортовано складники.

`<gitrepo>` визначає адресу репозиторію Git, якою слід скористатися, а `<branch>` вказує на гілку Git. Щоб імпортувати додаткові складники перекладу з наявного складника Weblate, скористайтеся адресою `weblate://<project>/<component>` для `<gitrepo>`.

`<filemask>` визначає спосіб пошуку файлів для репозиторію. Можна скористатися простими символами-замінниками або використати усю потужність формальних виразів.

При простому встановленні відповідності для назви складника використовується `**`, для мови — `*`. Приклад: `**/*.*po`

У формальному виразі мають міститися групи із назвою *component* і *language*. Приклад: `(?P<language>[^\/*]) / (?P<component>[^\/*]) \.po`

Засіб імпортування встановлює відповідність наявних складників на основі файлів і додає складники, яких не існує. Він не вносить змін до наявних складників.

--name-template TEMPLATE

Налаштування назви складника за допомогою синтаксису шаблонів Django.

Приклад: `Documentation: {{ component }}`

--base-file-template TEMPLATE

Налаштувати базовий файл для одномовних перекладів.

Приклад: `{{ component }}/res/values/string.xml`

--new-base-template TEMPLATE

Налаштування базового файла для додавання нових перекладів.

Приклад: `{{ component }}/ts/en.ts`

--file-format FORMAT

Ви також можете вказати формат файлів, який слід використати (див. [Підтримувані формати файлів](#)). Типовим є автоматичне визначення формату.

--language-regex REGEX

За допомогою цього параметра ви можете вказати фільтрування мов (див. [Налаштовування складників](#)). Значенням параметра має бути коректний формальний вираз.

--main-component

Ви можете вказати, який складник буде вибрано як основний — той, який насправді містить репозиторій системи керування версіями.

--license NAME

Визначення загальних умов ліцензування, умов ліцензування проєкту або умов ліцензування складника.

--license-url URL

Визначення адреси, за якою можна знайти умови ліцензування перекладу.

--vcs NAME

Якщо вам потрібно вказати, яку систему керування версіями слід використати, ви можете вказати її тут. Типовою системою керування версіями є `git`.

Для прикладу, давайте спробуємо імпортувати два проєкти.

Спочатку, розгляньмо переклади «The Debian Handbook», де для кожної мови передбачено окрему теку із перекладами кожної з глав книги:

```
weblate import_project \
  debian-handbook \
  git://anonscm.debian.org/debian-handbook/debian-handbook.git \
  squeeze/master \
  '*/**.po'
```

Далі, розгляньмо інструмент Tanaguru, де слід вказати формат файла, а також шаблон базового файла і те, як усі складники і переклади розташовано у одній теці:

```
weblate import_project \
  --file-format=properties \
  --base-file-template=web-app/tgol-web-app/src/main/resources/i18n/%s-I18N.
  →properties \
  tanaguru \
  https://github.com/Tanaguru/Tanaguru \
  master \
  web-app/tgol-web-app/src/main/resources/i18n/**-I18N_*.properties
```

Складніший приклад обробки назв файлів для отримання належних записів складників та мови з назви файла, зокрема `src/security/Numerous_security_holes_in_0.10.1.de.po`:

```
weblate import_project \
  tails \
  git://git.tails.boum.org/tails master \
  'wiki/src/security/(?P<component>.*).\.(?P<language>[^\.]*)\.po$'
```

Фільтрування з метою отримання лише перекладів вибраною мовою:

```
./manage import_project \  
  --language-regex '^ (cs|sk)$' \  
  weblate \  
  https://github.com/WeblateOrg/weblate.git \  
  'weblate/locale/*/LC_MESSAGES/**/*.po'
```

Імпортування документації Sphinx, поділеної на декілька файлів:

```
$ weblate import_project --name-template 'Documentation: %s' \  
  --file-format po \  
  project https://github.com/project/docs.git master \  
  'docs/locale/*/LC_MESSAGES/**/*.po'
```

Імпортування документації Sphinx, поділеної на декілька файлів і каталогів:

```
$ weblate import_project --name-template 'Directory 1: %s' \  
  --file-format po \  
  project https://github.com/project/docs.git master \  
  'docs/locale/*/LC_MESSAGES/dir1/**/*.po'  
$ weblate import_project --name-template 'Directory 2: %s' \  
  --file-format po \  
  project https://github.com/project/docs.git master \  
  'docs/locale/*/LC_MESSAGES/dir2/**/*.po'
```

Дивись також:

Докладніші приклади можна знайти у главі `starting`. Крім того, ви можете скористатися `import_json`.

2.18.17 importuserdata

weblate importuserdata <file.json>

Imports user data from a file created by `dumpuserdata`.

2.18.18 importusers

weblate importusers --check <file.json>

Імпортує записи користувачів з дампу JSON бази даних `auth_users` Django.

--check

Із цим параметром програма спочатку перевірить, чи може бути імпортовано вказаний файл, і повідомить про можливі конфлікти, які пов'язано із іменами користувачів або адресами електронної пошти.

Ви можете створити дамп записів користувачів на основі наявного встановленого екземпляра Django за допомогою такої команди:

```
weblate dumpdata auth.User > users.json
```


2.18.19 install_addon

Нове в версії 3.2.

weblate install_addon --addon ADDON <project|project/component>

Установлює додаток для набору складників.

--addon ADDON

Назва додатка, який слід встановити. Приклад: `weblate.gettext.customize`.

--configuration CONFIG

Закодовані у JSON налаштування додатка.

--update

Оновити наявні налаштування додатків.

Ви можете або визначити проєкт або складник для встановлення додатка (наприклад, `weblate/application`), або скористатися `--all` для включення усіх наявних складників.

Встановлення *Налаштувати виведення gettext* для всіх складників:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --
↪update --all
```

Дивись також:

Додатки

2.18.20 list_languages

weblate list_languages <locale>

Виводить список підтримуваних мов у форматі MediaWiki — коди мов, назви мов англійською та локалізовані назви мов.

Використовується для створення <https://wiki.l10n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF>.

2.18.21 list_translators

weblate list_translators <project|project/component>

Виводить список перекладачів за вказаною мовою для заданого проєкту:

```
[French]
Jean Dupont <jean.dupont@example.com>
[English]
John Doe <jd@example.com>
```

--language-code

Виводить список імен за кодом мов, а не за назвами мов.

Ви можете або визначити проєкт або складник для використання (наприклад, `weblate/application`), або скористатися `--all` для виведення списку перекладачів усіх наявних складників.

2.18.22 list_versions

weblate list_versions

Виводить список усіх залежностей Weblate та їхніх версій.

2.18.23 loadpo

weblate loadpo <project|project/component>

Перезавантажує переклади з диска (наприклад, у випадку, якщо ви виконували певні оновлення у репозиторії системи керування версіями).

--force

Примусове оновлення, навіть якщо файли мають бути актуальними.

--lang LANGUAGE

Обмежити обробку єдиною мовою.

Ви можете або визначити проєкт або складник для оновлення (наприклад, `weblate/application`), або скористатися `--all` для оновлення усіх наявних складників.

Примітка: Потреба у виклику цієї команди виникатиме нечасто — Weblate автоматично завантажуватиме змінені файли для будь-якого оновлення у системі керування версіями. Потреба у команді виникає тоді, коли ви вручну змінює підлеглий репозиторій системи керування версіями Weblate, або в особливих випадках після оновлення Weblate.

2.18.24 lock_translation

weblate lock_translation <project|project/component>

Забороняє подальший переклад складника.

Підказка: Корисна, якщо ви хочете виконати певні дії з супроводу над підлеглим репозиторієм.

Ви можете або визначити проєкт або складник для оновлення (наприклад, `weblate/application`), або скористатися `--all` для оновлення усіх наявних складників.

Дивись також:

`unlock_translation`

2.18.25 move_language

weblate move_language source target

Нове в версії 3.0.

Надає вам змогу об'єднувати дані мов. Корисна при оновленні до нової версії, яка містить альтернативні мови для раніше невідомих мов, записи яких було створено із суфіксом назви (*generated*). Переносить усі дані з мови *source* до мови *target*.

Приклад:

```
weblate move_language cze cs
```

Після перенесення даних вам слід перевірити, чи не залишилося чогось (таке трапляється через конкурен-тність при одночасному внесенню оновлень до репозиторію сторонньою особою), і вилучити мови із суфіксом (*generated*).

2.18.26 pushgit

weblate pushgit <project|project/component>

Записує внесені зміни до основного репозиторію системи керування версіями.

--force-commit

Примусово вносить до репозиторію усі зміни з черги до надсилання змін до репозиторію.

Ви можете або визначити проєкт або складник для оновлення (наприклад, weblate/application), або скористатися `--all` для оновлення усіх наявних складників.

Примітка: Weblate записує зміни до репозиторію автоматично, якщо позначено пункт *Відправляти при поданні* у налаштуваннях *складника* (типові налаштування).

2.18.27 unlock_translation

weblate unlock_translation <project|project/component>

Розблоковує вказаний складник, роблячи його доступним для перекладу.

Підказка: Корисна, якщо ви хочете виконати певні дії з супроводу над підлеглим репозиторієм.

Ви можете або визначити проєкт або складник для оновлення (наприклад, weblate/application), або скористатися `--all` для оновлення усіх наявних складників.

Дивись також:

lock_translation

2.18.28 setupgroups

weblate setupgroups

Налаштовує типові групи і, якщо потрібно, пов'язує усіх користувачів до цієї типової групи.

--no-privs-update

Вимикає автоматичне оновлення наявних груп (лише додає нові групи).

--no-projects-update

Запобігає автоматичним оновленням груп для наявних проєктів. Це уможливорює додавання новостворених груп до наявних проєктів, див. *Керування доступом на рівні проєкту*.

Дивись також:

Список привілеїв і вбудовані ролі

2.18.29 setuplang

weblate setuplang

Оновлює список визначених мов у Weblate.

--no-update

Вимикає автоматичні оновлення наявних мов (лише додає нові мови).

2.18.30 updatechecks

weblate updatechecks <project|project/component>

Оновлює всі перевірки для всіх рядків.

Підказка: Корисна для оновлень, у яких вносяться значні зміни до перевірок.

Ви можете або визначити проєкт або складник для оновлення (наприклад, `weblate/application`), або скористатися `--all` для оновлення усіх наявних складників.

2.18.31 updategit

weblate updategit <project|project/component>

Отримує дані з віддалених репозиторіїв системи керування версіями і оновлює вбудований кеш.

Ви можете або визначити проєкт або складник для оновлення (наприклад, `weblate/application`), або скористатися `--all` для оновлення усіх наявних складників.

Примітка: Зазвичай, краще налаштувати сторожові скрипт у репозиторії, щоб вмикати *Обробники сповіщень*, а не виконувати регулярне опитування за допомогою `updategit`.

2.19 Оголошення

Змінено в версії 4.0: У попередніх випусках ця можливість мала назву «повідомлення білої дошки».

Надайте відомості вашим перекладачам за допомогою створення оголошень — на рівні усього сайту, проєкту, складника або мови.

Оголошуйте про мету, контрольні дати, стан або вказуйте цілі для перекладу.

Користувачі отримають сповіщення при появі оголошень у проєктах, за якими вони спостерігають (якщо таке отримання не було вимкнено явним чином).

Це може бути корисним для різних речей — від оголошення мети на рівні сайту до визначення цілей для перекладу.

Оголошення можна розміщувати на будь-якому рівні меню *Управління* за допомогою *Повідомлення оголошення*:

Webate

Dashboard

Projects

Languages

Checks

+

WebateOrg

translated 90%

Translations will be used only if they reach 60%.

Components

Languages

Info

Search

Insights

Files

Tools

Manage

Share

Not watching

Post announcement

Message

You can use Markdown and mention users by @username.

Category

Info (light blue)

Category defines color used for the message.

Expiry date

mm/dd/yyyy

The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ Notify users

The message is shown for all translations within the project, until its given expiry, or permanently until it is deleted.

Add

Powered by Weblate 4.16

About Weblate

Legal

Contact

Documentation

Donate to Weblate

Додавання також можливе за допомогою адміністративного інтерфейсу:

Weblate administration
WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Home · Weblate translations · Announcements · Add Announcement




Add Announcement




Required fields are marked in bold.




Message:


Translations will be used only if they reach 60%


You can use Markdown and mention users by @username.

Project: WeblateOrg   

Component:   

Language:   

Category: Info (light blue) 
Category defines color used for the message.

Expiry date: Today 
The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ Notify users

Save and add another
Save and continue editing
SAVE

Далі, оголошення буде показано відповідно до вказаного контексту:

Контекст не вказано

Показано на панелі приладів (початковій сторінці).

Вказано проєкт

Показано у проєкті, включно із усіма його складниками і перекладами.

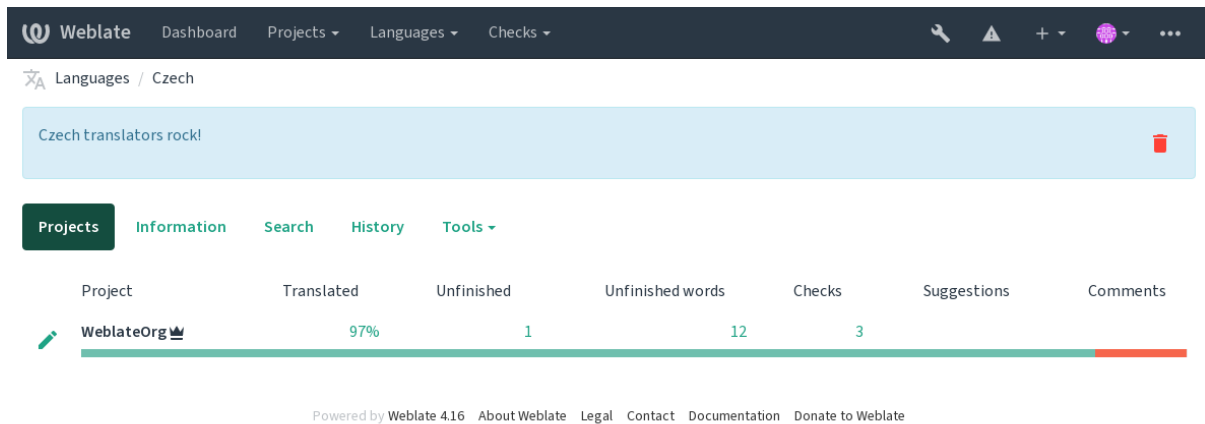
Вказано складник

Показано для вказаного складника й усіх його перекладів.

Вказано мову

Показано для огляду мови й усіх перекладів відповідною мовою.

Ось як це виглядає на сторінці огляду мови:



2.20 Списки складників

Вкажіть декілька списків складників, щоб їх було показано як пункти на панелі приладів користувача, з якої користувачі зможуть вибрати потрібний як типовий для перегляду. Див. [Домівка](#), щоб дізнатися більше.

Змінено в версії 2.20: Стан буде показано для кожного списку складників на панелі приладів.

Назви і вміст списків складників можна вказати у адміністративному інтерфейсі у розділі *Списки складників*. Кожен зі списків складників повинен мати назву, яку буде показано користувачеві, та ідентифікатор, який відповідатиме списку у адресі URL.

Змінено в версії 2.13: Змініть параметри панелі приладів для анонімних користувачів з адміністративного інтерфейсу, змінивши те, що буде показано на панелі приладів для користувачів, які не увійшли до системи.

2.20.1 Автоматичні списки складників

Нове в версії 2.13.

Додайте складники до списку автоматично на основі їхнього ідентифікатора створенням правил *Автоматичне призначення списку складників*.

- Корисно для супроводу списків складників для великих встановлень, або якщо вам потрібен один список складників з усіма складниками у вашому встановленому екземплярі Weblate.

Підказка: Створіть список складників, що міститиме усі складники встановленого вами Weblate.

1. Define *Automatic component list assignment* with `^.*$` as regular expression in both the project and the component fields, as shown on this image:

Weblate administration
WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Home · Weblate translations · Component lists · Add Component list

Add Component list

Required fields are marked in bold.

Component list name: Display name

URL slug: Name used in URLs and filenames.

☒ Show on dashboard
When enabled this component list will be shown as a tab on the dashboard

Components:

Available components ⓘ

- WeblateOrg/Django
- WeblateOrg/Language names
- WeblateOrg/WeblateOrg

Chosen components ⓘ

Choose all ⓘ Remove all ⓘ

Hold down "Control", or "Command" on a Mac, to select more than one.

AUTOMATIC COMPONENT LIST ASSIGNMENTS

PROJECT REGULAR EXPRESSION ⓘ	COMPONENT REGULAR EXPRESSION ⓘ	DELETE? ⓘ
<input type="text" value="^.*\$"/>	<input type="text" value="^.*\$"/>	<input type="button" value="X"/>

[+ Add another Automatic component list assignment](#)

2.21 Додаткові модулі Weblate

Для вашої конфігурації доступні декілька додаткових модулів.

2.21.1 Засіб експортування Git

Нове в версії 2.10.

Надає вам доступ лише до читання підлеглого репозиторію Git за допомогою HTTP(S).

Установлення

1. Додайте `weblate.gitexport` до запису встановлених застосунків у `settings.py`:

```
INSTALLED_APPS += ("weblate.gitexport",)
```

2. Експортування наявних репозиторіїв перенесенням вашої бази даних після встановлення:

```
weblate migrate
```

Користування

Модуль автоматично обробляє дані Weblate і встановлює адресу експортованого репозиторію у *складниці*. Доступ до репозиторіїв можна отримати за частиною `/git/` адреси Weblate, наприклад `https://example.org/git/weblate/main/`.

Репозиторії для загальнодоступних проєктів можна клонувати без розпізнавання:

```
git clone 'https://example.org/git/weblate/main/'
```

Доступ до навігації репозиторіями із обмеженнями доступу (з *керуванням доступом Private* або вмикання `REQUIRE_LOGIN`) потребує жетона програмного інтерфейсу, який можна отримати у вашому *профілі користувача*:

```
git clone 'https://user:KEY@example.org/git/weblate/main/'
```

Підказка: Типово учасники або групи *Користувач* та анонімний користувач мають доступ до репозиторіїв для загальнодоступних проєктів за допомогою ролей *Доступ до репозиторію* та *Досвідчений користувач*.

2.21.2 Оплата

Нове в версії 2.4.

Це використовується на [Hosted Weblate](#) для визначення тарифних планів, створення рахунків та встановлення обмежень у користуванні.

Установлення

1. Add `weblate.billing` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.billing",)
```

2. Запустіть перенесення бази даних, щоб додатково встановити структури бази даних для модуля:

```
weblate migrate
```

Користування

Після встановлення ви можете керувати рахунками за допомогою адміністративного інтерфейсу. Користувачі із увімкненим обліком побачать нову вкладку, *Оплата*, у їхньому *профіль користувача*.

Крім того, модуль обліку надає змогу адміністраторам проекту створювати проекти і складники без отримання прав надкористувача (див. *Додавання проектів і складників перекладу*). Це можливо при задоволенні таких умов:

- Рахунок перебуває у налаштованих межах (будь-яке перевищення обмежень призведе до блокування створення проектів і складників) із сплачено (якщо тарифний план не є нульовим)
- Користувач є адміністратором наявного проекту із обліком або користувач є власником рахунка (останнє є необхідним при створенні нового рахунку для користувачів, щоб мати змогу імпортувати нові проекти).

Під час створення проекту користувач може вибрати, до якого рахунка слід включати проект, якщо користувач має доступ до декількох з них.

2.21.3 Правові відомості

Нове в версії 2.15.

Це використовується на [Hosted Weblate](#) для надання обов'язкових юридичних документів. Типово використовуються порожні документи. Ви маєте заповнити такі шаблони у документах:

legal/documents/tos.html

Документ щодо умов надання послуг

legal/documents/privacy.html

Документ щодо правил конфіденційності

legal/documents/summary.html

Короткий огляд умов надання послуг та правил конфіденційності

При змін документів щодо умов обслуговування, будь ласка, скоригуйте `LEGAL_TOS_DATE`, щоб користувачам було запропоновано погодитися із оновленими документами.

Примітка: Юридичні документи для служби Hosted Weblate доступні у цьому репозиторії Git: <https://github.com/WeblateOrg/wllegal/tree/main/wllegal/templates/legal/documents>.

Найімовірніше, ці документи не будуть безпосередньо придатними для використання вами, але можуть бути корисними як база, яку ви можете скоригувати до ваших потреб.

Установлення

1. Add `weblate.legal` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.legal",)

# Optional:

# Social auth pipeline to confirm TOS upon registration/subsequent sign in
SOCIAL_AUTH_PIPELINE += ("weblate.legal.pipeline.tos_confirm",)

# Middleware to enforce TOS confirmation of signed in users
MIDDLEWARE += [
    "weblate.legal.middleware.RequireTOSMiddleware",
]
```

2. Запустіть перенесення бази даних, щоб додатково встановити структури бази даних для модуля:

```
weblate migrate
```

- Змініть юридичні документи у теці `weblate/legal/templates/legal/` так, щоб вони відповідали вашій службі.

Користування

Після встановлення і редагування юридичні документи буде показано у інтерфейсі користувача Weblate.

2.21.4 Аватари

Аватари завантажуються і кешуються на боці сервера для зменшення витоку відомостей для сайтів, які їх типово обслуговують. Вбудовану підтримку для отримання аватарів з адрес електронної пошти, які для них налаштовано, можна вимкнути за допомогою `ENABLE_AVATARS`.

У поточній версії Weblate передбачено підтримку таких служб:

- Gravatar
- Libravatar

Дивись також:

Кешування аватару, AVATAR_URL_PREFIX, ENABLE_AVATARS

2.21.5 Захист від спаму

Ви можете захистити систему від спаму користувачів за допомогою служби Akismet.

- Установіть модуль Python `akismet` (його вже включено в офіційний образ Docker).
- Отримайте ключ до програмного інтерфейсу Akismet.
- Зберігайте його як `AKISMET_API_KEY` або `WEBLATE_AKISMET_API_KEY` у Docker.

Цей вміст надіслано на перевірку Akismet:

- Пропозиції користувачів, які не пройшли розпізнавання
- Описи та посилання проєктів і складників

Примітка: Це (окрім інших речей) залежить від даних IP-адреси клієнта. Будь ласка, див. *Робота за реверсивним проксі-сервером*, щоб дізнатися більше про належне налаштування.

Дивись також:

Робота за реверсивним проксі-сервером, AKISMET_API_KEY, WEBLATE_AKISMET_API_KEY

2.21.6 Підписування внесків Git за допомогою GnuPG


Нове в версії 3.1.

Усі внески можна підписувати за допомогою ключа GnuPG екземпляра Weblate.

- Turn on `WEBLATE_GPG_IDENTITY`. (Weblate will generate a GnuPG key when needed and will use it to sign all translation commits.)

Для користування цією можливістю слід встановити GnuPG 2.1 або новішу версію.

Ключ зберігається у каталозі `DATA_DIR`, а відкритий ключ буде показано на сторінці «Про Weblate»:


[Weblate](#)
[Dashboard](#)
[Projects](#)
[Languages](#)
[Checks](#)
[Register](#)
[Sign in](#)

[About Weblate](#) / [Weblate keys](#)

[About Weblate](#)
[Statistics](#)
[Keys](#)

SSH key

Weblate uses SSH key to access remote repositories. The corresponding public key is found below, you can use it to grant Weblate access to a repository.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNZ8fKPCbs6hiYpnovamGbWdxygRSjmBGWjV0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhLwpluYlW1v
Weblate
```

Commit signing

All commits made with Weblate are signed with the GPG key 56E3C078521F2FFAB015106E7105A114D5061D6A, for which the corresponding public key is found below.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNBGP/WzoBDADL1oNK7oXL4YPx0wZIC99aTB0D5z8gA8Pdlcpk4mkdYBveDrLA
D/IW7z/mfgSqj0Yx3uZ8gkNMLM9mvAA0zURvQCpvZZN/Cudepua7ADtVYeVXaZ
PvCG9M8256PF2ZyO+PUI67HpLEJzHrhFjtjQcRoImKWMWfVfqHsL99lyq2lLeI7q
BaP/RKsbm09ziix/TGNMGPOq4l3JWtBfuApD5yp18235BjuabR++esiy4CtjRe0J
ncpNXA7xfyOt09IU3gUgmBRE3x8oH3B4fkjNAqK/rRobclWruW/3lsmMr9kt9wjh
lMLevg8aMnd0i+wq48sUpVXoo6nZ5gPoOzIR9EtvEuNJBLoMgQzdg3pgsPhhhH9
SaNbr2+V7BD10nyfqYuZRUec6GcbVXzdXFu8IzSjGktZzfjx14DvprEIL+HsAu2c
9kR08KS12KTUQHQdyJnEjIN5Q2tdabkiNgZwSOgHgjaN9kucnWOYeqDqdyIjklR
fM9xhcf08w6uRZ0AEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J
Ac4EEwEKADgWlQRW48B4U8v+rAVEG5xBaEU1QYdagUCY/9bOglbAwULCQgHAgYV
CgklCwIEFgIDAQIeAQIXgAAKCRBxBaEU1QYdapDkDkDp0vyAXKzFMNq0hoiyW
IHMk/zGH55mM04V1DyCvMnbuHkMfGu2y8mfq2QsrgelTUBikf0lyLu037+I/fJkO
-----
```

Powered by Weblate 4.16
 [About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking `gpg`.

Дивись також:

`WEBLATE_GPG_IDENTITY`

2.21.7 Обмеження частоти

Змінено в версії 3.2: У поточній версії засіб обмеження частоти можна налаштувати точніше.

Змінено в версії 4.6: Обмеження на частоту більше не стосується надкористувачів.

Частоту деяких дій у Weblate можна обмежити. Дозволено виконувати `RATELIMIT_ATTEMPTS` спроб дії протягом `RATELIMIT_WINDOW` секунд. При перевищенні обмеження користувача буде заблоковано на значення `RATELIMIT_LOCKOUT`. Також передбачено специфічні параметри для областей, наприклад `RATELIMIT_CONTACT_ATTEMPTS` або `RATELIMIT_TRANSLATE_ATTEMPTS`. У розташованій нижче таблиці наведено повний список доступних областей.

Обмеженню частоти підлягають такі дії:

Назва	Область	Дозволені спроби	Вікно обмеження частоти	Період блокування
Реєстрація	REGISTRATION	5	300	600
Надсилання повідомлення адміністраторам	MESSAGE	2	300	600
Розпізнавання за паролем при вході	LOGIN	5	300	600
Загальний пошук за сайтом	SEARCH	6	60	60
Переклад	TRANSLATE	30	60	600
Додавання до глосарія	GLOSSARY	30	60	600
Започаткування перекладу новою мовою	LANGUAGE	2	300	600
Створення нового проєкту	PROJECT	5	600	600

Якщо користувач не зможе увійти до системи `AUTH_LOCK_ATTEMPTS` разів, на обліковому записі буде вимкнено розпізнавання за паролем, аж доки не буде виконано процедуру скидання пароля облікового запису.

Крім того, параметри може бути застосовано у контейнері Docker додаванням префікса `WEBLATE_` до назви параметра. Приклад: `RATELIMIT_ATTEMPTS` слід замінити на `WEBLATE_RATELIMIT_ATTEMPTS`.

У програмному інтерфейсі є окремі параметри обмеження частоти, див. *Обмеження частоти у програмному інтерфейсі*.

Дивись також:

Обмеження частоти, Робота за реверсивним проксі-сервером, Обмеження частоти у програмному інтерфейсі

2.21.8 Інтеграція Fedora Messaging

Fedora Messaging — заснований на AMQP засіб публікації усіх змін, які відбуваються у Weblate. За її допомогою ви можете будувати додаткові служби на основі змін, які відбуваються у Weblate.

Інтеграція з Fedora Messaging відбувається за допомогою окремого модуля Python, `weblate-fedora-messaging`. Настанови з налаштування можна знайти тут: https://github.com/WeblateOrg/fedora_messaging/.

2.22 Налаштовування Weblate

Розширюйте можливості та налаштовуйте систему за допомогою Django і Python. Надсилайте ваші зміни до основної гілки розробки, щоб ними могли скористатися усі. Це зменшує витрати на супровід системи; код у Weblate важливий при зміні внутрішніх інтерфейсів або переплануванні коду.

Попередження: Ні внутрішні інтерфейси, ні шаблони не вважаються стабільним програмним інтерфейсом. Будь ласка, переглядайте ваші налаштування при кожному оновленні — інтерфейси або їхня семантика можуть змінитися без попередження.

Дивись також:

Участь у розробці Weblate

2.22.1 Створення модуля Python

Якщо ви не є професіоналом у Python, вам варто ознайомитися із книгою [Python For Beginners](#), де наведено пояснення щодо основ мови та посилання на інші підручники.

Для написання файл із нетиповим кодом Python (який називається модулем), потрібний репозиторій для зберігання його даних — або у загальносистемному каталозі (зазвичай, щось подібне до `/usr/lib/python3.9/site-packages/`), або у каталозі Weblate, який також додається до шляху пошуку інтерпретатора мови.

Нове в версії 3.8-5: Якщо *використано Docker*, ви можете розташувати модулі Python у `/app/data/python/` (див. *Тому контейнера Docker*) для завантаження їх Weblate, наприклад, з *файла перевизначення параметрів*.

Ще краще — перетворіть ваші налаштування на належний пакунок Python:

1. Створіть теку для вашого пакунка (ми використовуємо `weblate_customization`).
2. У теці створіть файл `setup.py` для опису пакунка:

```
from setuptools import setup

setup(
    name="weblate_customization",
    version="0.0.1",
    author="Your name",
    author_email="yourname@example.com",
    description="Sample Custom check for Weblate.",
    license="GPLv3+",
    keywords="Weblate check example",
    packages=["weblate_customization"],
)
```

3. Створіть теку для модуля Python (також називається `weblate_customization`) для коду налаштування.
4. У ній створіть файл `__init__.py`, щоб переконатися, що Python може імпортувати модуль.
5. Тепер цей пакунок можна встановити за допомогою команди `pip install -e`. Докладніший опис можна знайти у розділі [Editable installs](#).
6. Після встановлення модулем можна скористатися у налаштуваннях Weblate (наприклад `weblate_customization.checks.FooCheck`).

Ваша структура пакунка має виглядати десь так:

```
weblate_customization
├─ setup.py
└─ weblate_customization
    ├─ __init__.py
    ├─ addons.py
    └─ checks.py
```

Приклад налаштування Weblate можна знайти тут: <https://github.com/WeblateOrg/customize-example>. У цьому прикладі реалізовано усі речі, які описано нижче.

2.22.2 Зміна логотипа

1. Створіть просту програму Django, яка міститиме статичні файли, які ви хочете перезаписати (див. *Створення модуля Python*).

Логотип буде використано у таких файлах:

icons/weblate.svg

Логотип, який буде показано на смужці навігації.

logo-*.png

Вебпиктограми залежатимуть від роздільності екрана та браузера.

favicon.ico

Вебпиктограма для використання у застарілих браузерах.

weblate-*.png

Аватари для ботів або анонімних користувачів. У деяких браузерах ці піктограми буде використано як мініатюри.

email-logo.png

Використовується у повідомленнях електронної пошти зі сповіщеннями.

2. Додайте її до `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    # Add your customization as first
    "weblate_customization",
    # Weblate apps are here...
)
```

3. Віддайте команду `weblate collectstatic --noinput`, щоб зібрати статичні файли, які надаються клієнтам.

Дивись також:

[How to manage static files \(e.g. images, JavaScript, CSS\)](#), *Обслуговування статичних файлів*

2.22.3 Нетипові перевірки якості, додатки та автоматичні виправлення

Щоб встановити ваш код для *Нетипові автоматичні виправлення*, *Написання власних перевірок* або *Написання додатка* у Weblate, виконайте такі дії:

1. Розташуйте належним чином файли у вашому модулі Python, які містять налаштування Weblate (див. *Створення модуля Python*).
2. Додайте його шлях повністю до класу Python у відповідних параметрах (`WEBLATE_ADDONS`, `CHECK_LIST` або `AUTOFIX_LIST`):

```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

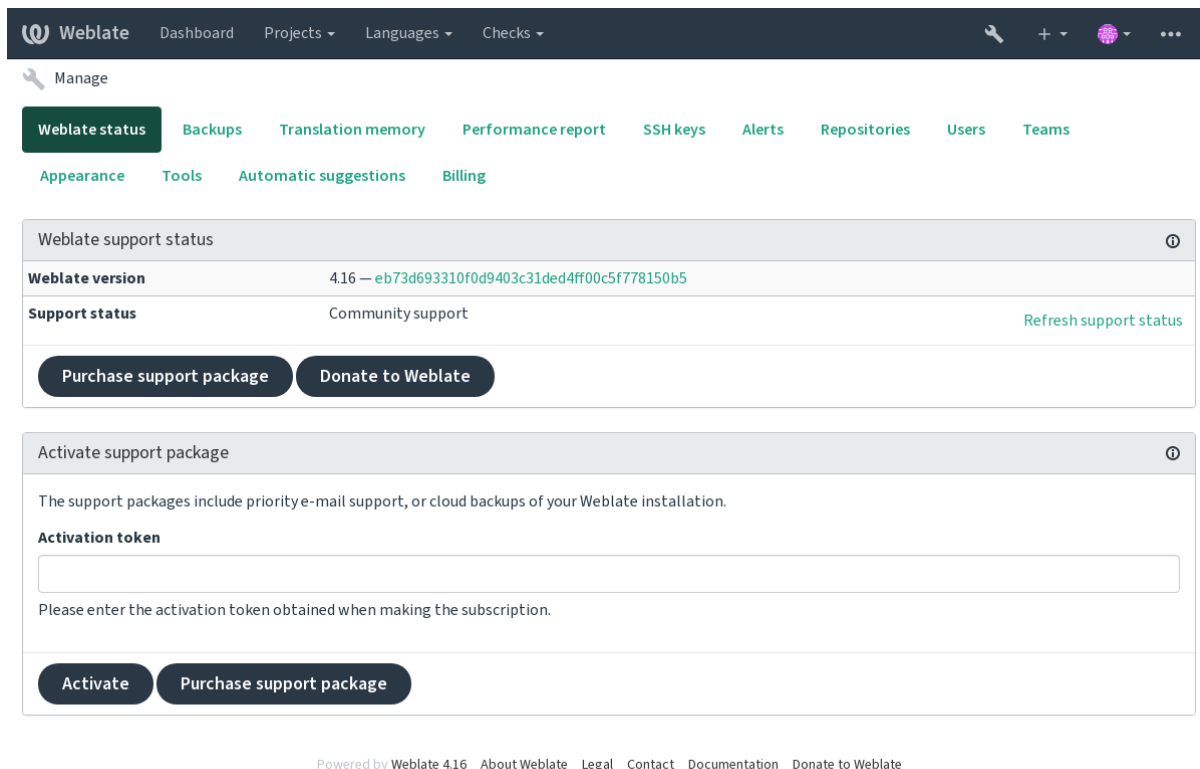
# Add-ons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Дивись також:

Нетипові автоматичні виправлення, *Написання власних перевірок*, *Написання додатка*, *Виконання скриптів з додатка*

2.23 Інтерфейс керування

Адміністративні параметри інтерфейсу керування зберігаються у каталозі `/manage/`. Доступ до інтерфейсу можуть отримати користувачі, які увійшли до системи із правами доступу адміністратора. Кнопку доступу із зображенням гайкового ключа розташовано у верхній правій частині сторінки:



Сюди включено базовий огляд вашого Weblate:

- Стан підтримки, див. *Отримання підтримки щодо Weblate*
- Резервні копії, див. *Резервне копіювання і пересування Weblate*
- Спільна пам'ять перекладів, див. *Пам'ять перекладів*
- Звіт щодо швидкодії для визначення стану Weblate та довжини черг Celery
- Керування ключами SSH, див. *Резерваторії із доступом за SSH*
- Огляд сповіщень для всіх складників, див. `alerts`

2.23.1 Адміністративний інтерфейс Django

Попередження: Будьте обережні з користуванням, оскільки це низькорівневий інтерфейс. Здебільшого, потреби у його використанні немає, оскільки доступ до більшості речей можна комфортно здійснювати за допомогою інтерфейсу користувача або програмного інтерфейсу Weblate.

Тут ви можете керувати об'єктами, які зберігаються у базі даних, зокрема записами користувачів, перекладів та іншими параметрами:

Webplate administration

WELCOME WEBLATE TEST RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT

Site administration

REPORTS

Webplate support status

Status of repositories

SSH keys

Performance report

Translation memory

ACCOUNTS

Audit log entries

+ Add

Change

User profiles

+ Add

Change

Verified e-mails

+ Add

Change

AUTH TOKEN

Tokens

+ Add

Change

AUTHENTICATION

Groups

+ Add

Change

Roles

+ Add

Change

Users

+ Add

Change

BILLING

Billing plans

+ Add

Change

Customer billings

+ Add

Change

Invoices

+ Add

Change

FONTS

Font groups

+ Add

Change

Fonts

+ Add

Change

LEGAL

TOS agreements

+ Add

Change

PERIODIC TASKS

Clocked

+ Add

Change

Crontabs

+ Add

Change

Intervals

+ Add

Change

Periodic tasks

+ Add

Change

Solar events

+ Add

Change

PYTHON SOCIAL AUTH

Associations

+ Add

Change

Nonces

+ Add

Change

User social auths

+ Add

Change

SCREENSHOTS

Screenshots

+ Add

Change

TRANSLATION MEMORY

Translation memory entries

+ Add

Change

WEBLATE CONFIGURATION

Settings

+ Add

Change

WEBLATE LANGUAGES

Languages

+ Add

Change

WEBLATE TRANSLATIONS

Announcements

+ Add

Change

Component lists

+ Add

Change

Components

+ Add

Change

Contributor agreements

+ Add

Change

Projects

+ Add

Change

Recent actions

My actions

None available

У розділі *Звіти* ви можете перевірити стан вашого сайту, скоригувати його для *Промислові налаштування* або керувати ключами SSH, які використовуються для доступу до *Доступ до репозиторіїв*.

Ви можете керувати об'єктами бази даних у будь-якому із описаних нижче розділів. Ймовірно, найцікавішим буде розділ *Переклади Weblate*, де ви можете керувати придатними до перекладу проєктами, див. *Налаштування проєкту* і *Налаштування складників*.

Мови Weblate містять визначення мов, докладніший опис у розділі *Визначення мов*.

Додавання проєкту

Додавання проєкту створює контейнер для всіх складників. Зазвичай ви створюєте один проєкт для однієї частини програмного забезпечення або для однієї книги (див. *Налаштування проєкту*, щоб дізнатися більше про окремі параметри):

Weblate administration

WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)[Home](#) · [Weblate translations](#) · [Projects](#) · [Add Project](#)

Add Project

Required fields are marked in bold.

Project name:	<input type="text" value="WeblateOrg"/>
	Display name

URL slug:	<input type="text" value="weblateorg"/>
	Name used in URLs and filenames.

Project website:	<input type="text" value="https://weblate.org/"/>
	Main website of translated project.

Translation instructions:	<div><input type="text" value="https://weblate.org/contribute/"/></div>
	You can use Markdown and mention users by @username.

☒ **Set "Language-Team" header**
Lets Weblate update the "Language-Team" file header of your project.

☒ **Use shared translation memory**
Uses the pool of shared translations between projects.

☒ **Contribute to shared translation memory**
Contributes to the pool of shared translations between projects.

Access control:	<div>Protected ▾</div>
	How to restrict access to this project is detailed in the documentation.

☐ **Enable reviews**
Requires dedicated reviewers to approve translations.

☐ **Enable source reviews**
Requires dedicated reviewers to approve source strings.

☒ **Enable hooks**
Whether to allow updating this repository by remote hooks.

Language aliases:	<div></div>
	Comma-separated list of language code mappings, for example: en_GB:en,en_US:en

Machinery settings:	<div>⌀</div>
---------------------	--------------

Дивись також:*Налаштування проекту*

Двомовні складники

Після додавання проєкту до нього можна додати складники перекладу. (Див. [Налаштування складників](#), щоб дізнатися більше про окремі параметри):

[illegible]

Дивись також:

Налаштування складників, Двомовні і одномовні формати

Одномовні складники

Для спрощення перекладу вкажіть файл шаблону, який міститиме прив'язку ідентифікаторів повідомлень до відповідної початкової мови (зазвичай, англійської). (Див. *Налаштування складників*, щоб дізнатися більше про окремі параметри):

Дивись також:

Налаштовування складників, Двововні і однововні формати

2.24 Отримання підтримки щодо Weblate

Weblate є вільним програмним забезпеченням, модифікація якого можлива лише за умов вільного її поширення, із супроводом, який забезпечується спільнотою користувачів. Передплатники отримують пріоритетну підтримку без додаткових сплат. Передплачені пакунки допомоги доступні для всіх. Докладніше про поточні пропозиції супроводу можна дізнатися за допомогою сторінки <<https://weblate.org/support/>>.

2.24.1 Інтегрування підтримки

Нове в версії 3.8.

Придбані пакунки підтримки може бути, якщо хочете, інтегровано до інтерфейсу керування передплатами вашого Weblate, де буде розміщено відповідні посилання. У цьому випадку до Weblate надсилатимуться дані щодо базових параметрів установленого вами екземпляра.

The screenshot shows the Weblate dashboard interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a 'Manage' section with various tabs: 'Weblate status' (active), 'Backups', 'Translation memory', 'Performance report', 'SSH keys', 'Alerts', 'Repositories', 'Users', 'Teams', 'Appearance', 'Tools', 'Automatic suggestions', and 'Billing'. The 'Weblate status' section displays the following information:

- Weblate support status** (with an info icon)
- Weblate version**: 4.16 — eb73d693310f0d9403c31ded4ff00c5f778150b5
- Support status**: Community support (with a 'Refresh support status' link)
- Buttons: 'Purchase support package' and 'Donate to Weblate'

Below this, the 'Activate support package' section (also with an info icon) contains:

- A description: 'The support packages include priority e-mail support, or cloud backups of your Weblate installation.'
- Activation token**: A text input field.
- A note: 'Please enter the activation token obtained when making the subscription.'
- Buttons: 'Activate' and 'Purchase support package'

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

2.24.2 Дані, які подаються до Weblate

- Адреса, за якою налаштовано ваш екземпляр Weblate
- Назва вашого сайта
- Версія Weblate, якою ви користуєтесь
- Бирки деяких об'єктів у вашій базі даних Weblate (проектів, складників, записів мов, початкових рядків та користувачів)
- Відкритий ключ SSH вашого екземпляра

Крім того, якщо увімкнено *Виявлення Weblate*:

- Список загальнодоступних проєктів (назва, адреса і сайт)

Інші дані не подаються.

2.24.3 Служби інтеграції

- Переконайтеся, що ваш пакунок підтримки є чинним
- *Передбачене у Weblate сховище резервних копій*
- *Виявлення Weblate*

Підказка: Придбані пакунки підтримки є активованими з моменту придбання — ними можна користуватися без інтеграції.

2.24.4 Виявлення Weblate

Нове в версії 4.5.2.

Примітка: Ця можливість у поточній версії перебуває на ранньому етапі тестування.

«Виявлення Weblate» є додатковою службою, яка спрощує для користувачів пошук серверів та спільнот Weblate. Користувачі можуть здійснювати навігацію зареєстрованими службами на <<https://weblate.org/discover/>> і знаходити там проєкти для участі в них.

Потрапляння до списку

Підказка: Участь у Discover Weblate дозволяє Weblate надсилати певні відомості щодо вашого сервера. Будь ласка, див. *Дані, які подаються до Weblate*.

Щоб ваш сервер потрапив до списку із активною передплатою підтримки (див. *Інтегрування підтримки*) у Discover Weblate, вам достатньо увімкнути можливість на панелі керування:

Weblate
 Dashboard Projects Languages Checks

Manage

Weblate status

Backups

Translation memory

Performance report

SSH keys

Alerts

Repositories

Users

Teams

Appearance

Tools

Automatic suggestions

Billing

Weblate support status

Weblate version

4.16 — eb73d693310f0d9403c31ded4ff00c5f778150b5

Support status

Community support

Refresh support status

Discover Weblate

Your Weblate is not listed on weblate.org

Enable discovery

Browse discovery

Manage support package

Purchase support package

Donate to Weblate

Activate support package

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate

Purchase support package

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Додавання вашого сервера без передплати підтримки до Discover Weblate:

1. Зареєструйтеся на <<https://weblate.org/user/>>
2. Зареєструйте ваш сервер Weblate у базі даних виявлення на сторінці <<https://weblate.org/subscription/discovery/>>
3. Підтвердьте активацію служби у вашому Weblate і увімкніть список виявлення на вашій сторінці керування Weblate за допомогою кнопки *Увімкнути виявлення*:

430

Розділ 2. Документація для адміністраторів

Weblate
 Dashboard Projects Languages Checks

Manage

Weblate status
 Backups
 Translation memory
 Performance report
 SSH keys
 Alerts
 Repositories
 Users
 Teams

Appearance
 Tools
 Automatic suggestions
 Billing

Weblate support status ⓘ

Weblate version 4.16 — eb73d693310f0d9403c31ded4ff00c5f778150b5

Support status Community support Refresh support status

Discover Weblate Your Weblate is not listed on weblate.org Browse discovery

Enable discovery

Manage support package Purchase support package Donate to Weblate

Activate support package ⓘ

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate Purchase support package

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Налаштовування пункту у списку

Ви можете налаштувати ваш пункт списку, надавши текст і зображення (570 x 260 пікселів) на сторінці <https://weblate.org/user/>.

2.25 Правові документи

Примітка: Тут ви знайдете різноманітні юридичні відомості, які можуть знадобитися вам для роботи з Weblate у певних юрисдикціях. Їх надано для орієнтування, без жодних гарантій точності або правильності. Уся відповідальність щодо забезпечення повної відповідності законодавству користування вами Weblate покладається на вас.

2.25.1 Сумісність ліцензування

Weblate постачається зі специфікацією сумісності ліцензування [REUSE 3.0](#).

2.25.2 ITAR та інші обмеження експортування

Weblate може працювати у вашому власному датацентрі або віртуальній приватній хмарі. Як такий, Weblate можна використовувати для зберігання ITAR або інших відомостей, які підлягають експортному контролю. Втім, за забезпечення виконання відповідних умов відповідальність несуть кінцеві користувачі.

Аудит на відповідність ITAR та іншим умовам експортного контролю для Hosted Weblate не виконувався. У поточній версії цей сайт не надає можливості обмежувати доступ до перекладів за країною.

2.25.3 Умови керування шифруванням США

У Weblate немає специфічного коду для шифрування, але Weblate може підлягати експортному контролю, оскільки використовує сторонні складники, які використовують шифрування для розпізнавання користувачів, забезпечення цілісності та конфіденційності даних.

Найімовірніше, Weblate слід класифікувати як ECCN 5D002 або 5D992 і, як загальнодоступне вільне програмне забезпечення, Weblate не повинен підлягати EAR (див. [Об'єкти шифрування, які НЕ підлягають EAR](#)).

Програмні складники, які використовуються Weblate (у списку наведено лише складники, які пов'язано із криптографією):

Python

Див. https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

GnuPG

Може використовуватися Weblate

Git

Може використовуватися Weblate

curl

Використовується Git

OpenSSL

Використовується Python і cURL

Стійкість ключів шифрування залежить від налаштувань Weblate і сторонніх складників, з якими взаємодіє Weblate, але у будь-якій вартій уваги конфігурації Weblate включатиме усі криптографічні функції із експортними обмеженнями:

- Із перевищенням 56-бітів для симетричного алгоритму
- Розклад на множники цілих чисел з перевищенням 512 бітів для асиметричного алгоритму
- Обчислення дискретних логарифмів у мультиплікативній групі скінченного поля із розміром, що перевищує 512 бітів, для асиметричного алгоритму
- Дискретні логарифми у групі, відмінній від вказаної вище, із перевищенням 112 бітів для асиметричного алгоритму

У Weblate ще не передбачено можливості якось активувати криптографію, але Weblate можна налаштувати так, що не використовуватиметься ніякий криптографічний код. Серед криптографічних можливостей Weblate можна відзначити такі:

- Доступ до віддалених серверів за допомогою захищених протоколів (HTTPS)
- Створення підписів до внесків до коду (PGP)

Дивись також:

[Обмеження щодо експорту \(EAR\) для вільного відкритого програмного забезпечення](#)

3.1 Участь у розробці Weblate

Існує десятки варіантів взяти участь у поліпшенні Weblate. Ви можете вибрати той з них, який вам подобається найбільше — програмування, графічний дизайн, документація, безпосередня фінансова підтримка чи ідеї:

- *Надсилання звітів щодо вад у Weblate*
- *Перші кроки у програмуванні коду Weblate*
- *Участь у розробці модулів Weblate*
- *Переклад Weblate*
- *Посприяйте документуванню Weblate*
- *Обговорення у Weblate*
- *Фінансування розробки Weblate*

3.1.1 Переклад Weblate

Weblate постійно [перекладається](#) за допомогою самого Weblate. Не вагайтеся і долучайтесь до зусиль з забезпечення доступності Weblate якомога більшою кількістю мов. Переклад робить Weblate ближчим до користувачів!

Якщо вами виявлено можливу помилку у початковому рядку, ви можете позначити її коментарем у редакторі Weblate. У такий спосіб її може бути обговорено і виправлено. Якщо ви певні щодо помилки, ви також можете натиснути посилання у розділі *Позиція джерельного рядка* і створити запит щодо об'єднання із вашим виправленням.

3.1.2 Посприяйте документуванню Weblate

Будемо раді вашій участі в удосконаленні будь-яких сторінок документації. Зробити це просто: натисніть кнопку *Редагувати на GitHub* у верхньому правому куті вибраної сторінки.

Будь ласка, дотримуйтеся таких настанов під час написання тексту:

1. Не вилучайте частини документації, якщо вона є чинною.
2. Користуйтеся чіткою і зрозумілою мовою. Ви пишете технічну документацію, не поему. Не усі читачі документації володіють англійською, будьте уважні.
3. Не бійтеся спитати, якщо щось не зрозуміло. Якщо ви запитали про якусь можливість під час редагування, не змінюйте документацію щодо неї, доки не отримаєте відповіді. Це означає: або змінійте, або питайте. Не виконуйте обидві дії одночасно.
4. Перевіряйте ваші зміни, виконуючи описані дії відповідно до документації.
5. Надсилайте запити щодо об'єднання малими фрагментами, щоб спростити і пришвидшити їхнє рецензування та злиття до репозиторію.
6. Якщо вам хочеться переписати або змінити структуру великого розділу, робіть це у два кроки:
 1. Переписування
 2. Щойно переписування буде рецензовано, виправлено та злито до репозиторію, змініть структуру абзаців в іншому запиті щодо об'єднання.

Підказка: Ви можете [перекласти документацію](#).

3.1.3 Розширення вбудованих визначень мов

Визначення мов зберігаються у [репозиторії weblate-language-data](#).

Ви можете додати відсутні визначення мови до `languages.csv` з цього файлу генеруються інші файли.

3.1.4 Обговорення у Weblate

Якщо у вас є ідея і ви не певні, чи варта вона створення повідомлення про ваду, не переймайтеся. Ви можете долучитися до спільноти у [обговореннях GitHub](#).

3.1.5 Фінансування розробки Weblate

Ви можете сприяти подальшій розробці Weblate за допомогою [сторінки допомоги](#). Зібрані кошти буде використано для фінансування безплатного обслуговування проєктів вільного програмного забезпечення і подальшої розробки Weblate. Будь ласка, ознайомтеся зі [сторінкою допомоги](#), щоб дізнатися більше, зокрема про цілі фінансування та відзнаки, які ви можете отримати як меценат.

Спонсори, які фінансували Weblate

Список спонсорів Weblate:

- Yashiro Ccs
- Cheng-Chia Tseng
- Timon Reinhard
- Cassidy James
- Loic Dachary
- Marozed
- <https://freedombox.org/>
- GNU Solidario (GNU Health)
- BallotReady
- Richard Nespithal
- MyExpenses.Mobi
- Michael Totschnig

Хочете потрапити до списку? Будь ласка, ознайомтеся із вмістом сторінки [Пожертви Weblate](#).

3.2 Перші кроки у програмуванні коду Weblate

Щоб краще зрозуміти початковий код Weblate, будь ласка, спочатку зверніться до розділів *Початковий код Weblate*, *Оболонка Weblate* та *Нутрощі Weblate*.

3.2.1 Перші кроки на основі базового програмного коду

Якщо ви шукаєте якісь вади, які допоможуть ознайомитися із початковим кодом Weblate, зверніться до тих з них, які позначено міткою `good first issue`.

Ми будемо раді, якщо ви почнете роботу над цими проблемами, навіть нічого не питаючи. Просто повідомте про початок роботи у гілці звіту щодо вади, щоб усі знали, що над вирішенням проблеми уже хтось працює.

3.2.2 Локальний запуск Weblate

Найкомфортнішим способом розпочати розробку Weblate є виконання настанов розділу `../admin/install/source`. Виконання цих настанов надасть вам змогу отримати віртуальне середовище із придатними до редагування початковими кодами Weblate.

1. Клонувати початковий код Weblate:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Створіть віртуальне середовище:

```
virtualenv .venv
.venv/bin/activate
```

3. Установіть Weblate (це потребуватиме декількох залежностей у системі, див. `../admin/install/source`):

```
pip install -e .
```

3. Встановити всі корисні для розробки залежності:

```
pip install -r requirements-dev.txt
```

4. Запустити сервер розробки:

```
weblate runserver
```

5. Залежно від ваших налаштувань, вам може знадобитися запуск обробників Celery:

```
./weblate/examples/celery start
```

6. Для запуску тестування (див. *Локальне тестування*, щоб дізнатися більше):

```
. scripts/test-database.sh  
./manage.py test
```

Дивись також:

Установлення з початкового коду

3.2.3 Запуск Weblate локально у Docker

Якщо у вас встановлено Docker і docker-compose, ви можете запустити середовище розробки простими командами:

```
./rundev.sh
```

У результаті буде створено і запущено образ Docker для розробки. Weblate працюватиме за адресою `<http://127.0.0.1:8080/>`, і ви зможете увійти від імені користувача `admin` із паролем `admin`. Новий встановлений екземпляр буде порожнім, тому ви можете продовжити з *Додавання проєктів і складників перекладу*.

Файли `Dockerfile` і `docker-compose.yml` для цього екземпляра зберігатимуться у каталозі `dev-docker`.

Крім того, скрипт приймає декілька параметрів. Щоб виконати перевірки, запустіть його з параметром `test`, а потім вкажіть будь-які параметри `test`. Приклад запуску тестів лише у модулі `weblate.machine`:

```
./rundev.sh test --failfast weblate.machine
```

Примітка: Переконайтеся, що ваші контейнери Docker запущено, і вони працюють, перш ніж запускати тести. Зробити це можна за допомогою команди `docker ps`.

Для перегляду журналів:

```
./rundev.sh logs
```

Щоб зупинити контейнери, які працюють у фоновому режимі, віддайте таку команду:

```
./rundev.sh stop
```

Запуск скрипту без аргументів призведе до повторного створення контейнера Docker і його перезапуску.

Примітка: Ця конфігурація є непридатною для промислових варіантів, до неї включено декілька модифікацій, які не є безпечними, але пришвидшують розробку.

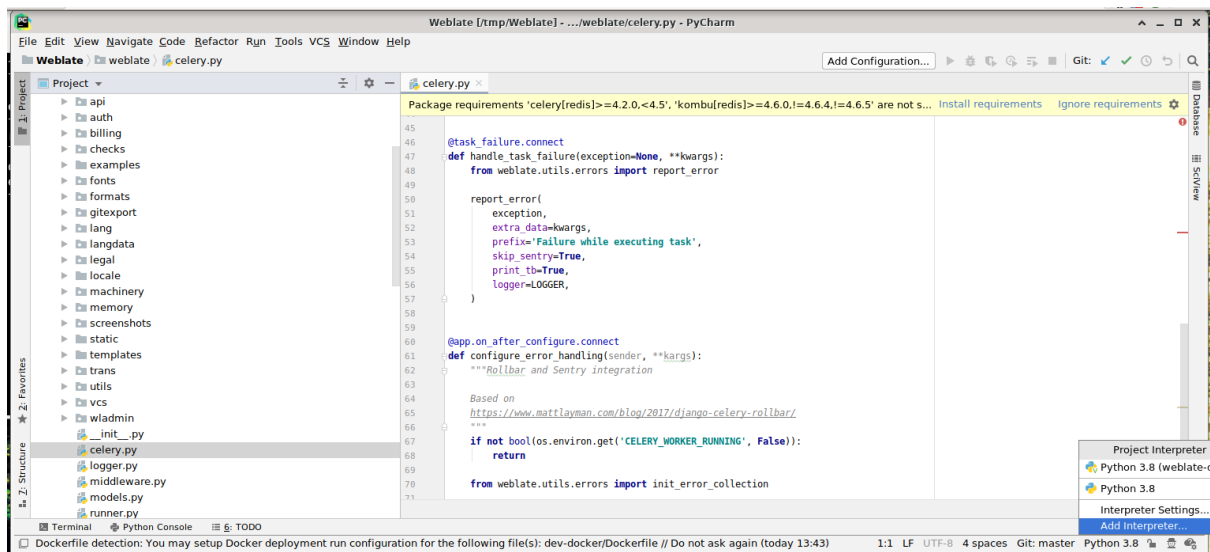
3.2.4 Самопобудова вашого екземпляра для розробки

Ймовірно, вам варто скористатися `import_demo` для створення демонстраційних перекладів і `createadmin`, щоб створити запис користувача-адміністратора.

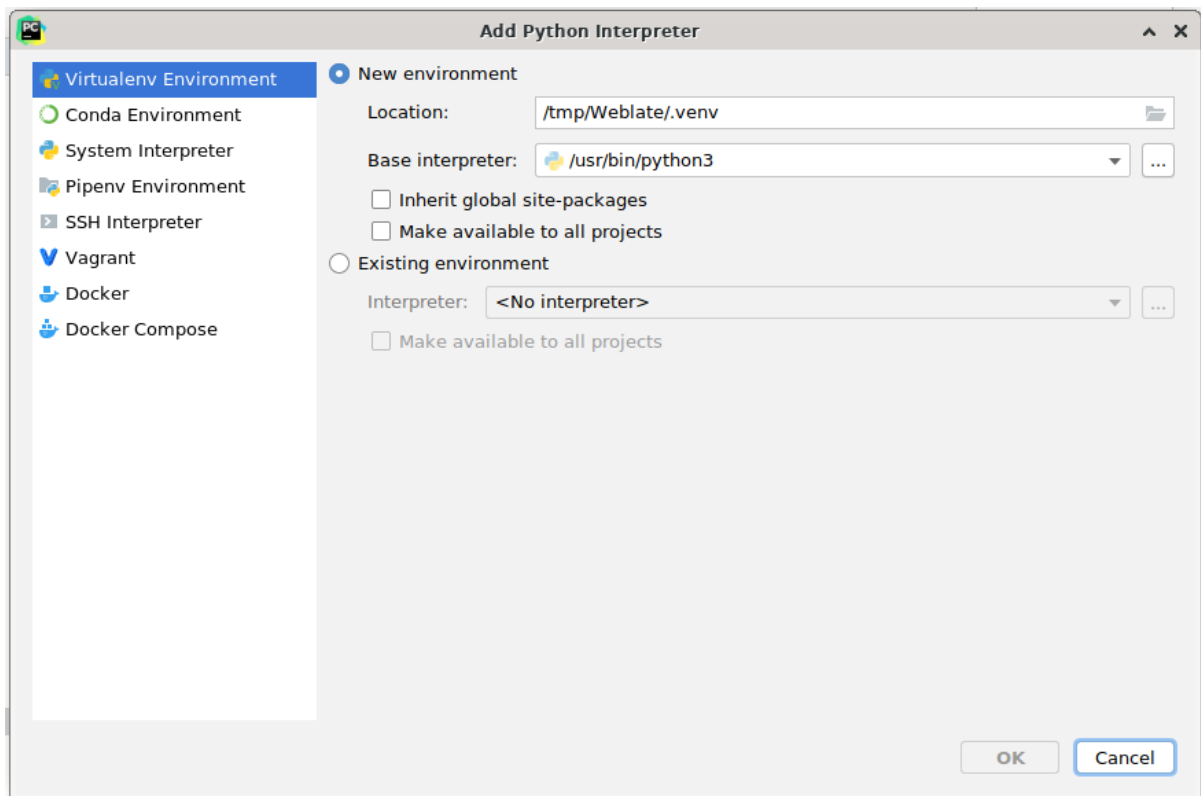
3.2.5 Програмування Weblate з використанням PyCharm

PyCharm є відомим комплексним середовищем розробки для Python. У цьому розділі наведено декілька порад, які допоможуть вам налаштувати ваш проєкт Weblate.

Припустімо, що ви вже створили клон репозиторію GitHub і відкрили теку, до якої було клоновано код, у PyCharm. Після відкриття вікна середовища розробки першим кроком є визначення потрібного вам інтерпретатора:

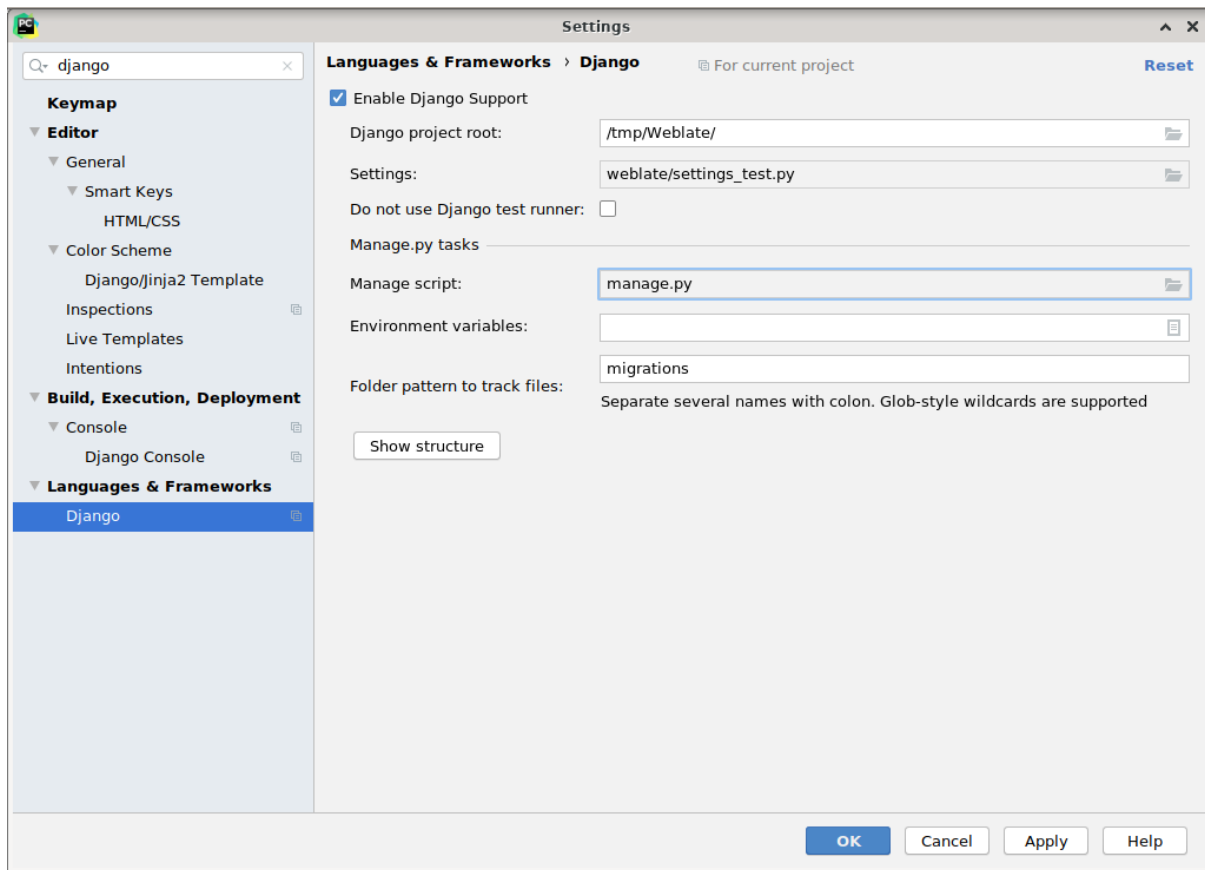


Ви можете або дозволити PyCharm створити для вас віртуальне середовище, або вибрати наявне середовище:



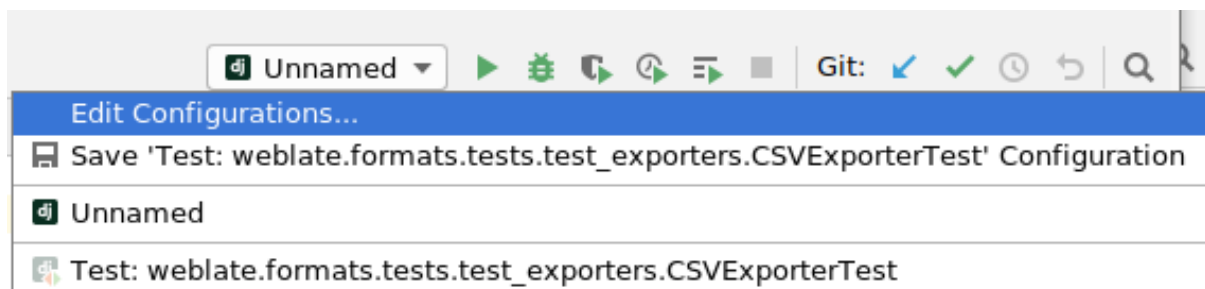
Не забудьте встановити залежності після встановлення інтерпретатора. Зробити це можна або за допомогою консолі (вбудована консоль типово безпосередньо використовуватиме ваше віртуальне середовище), або за допомогою графічного інтерфейсу, коли ви отримаєте попередження щодо невстановлених залежностей.

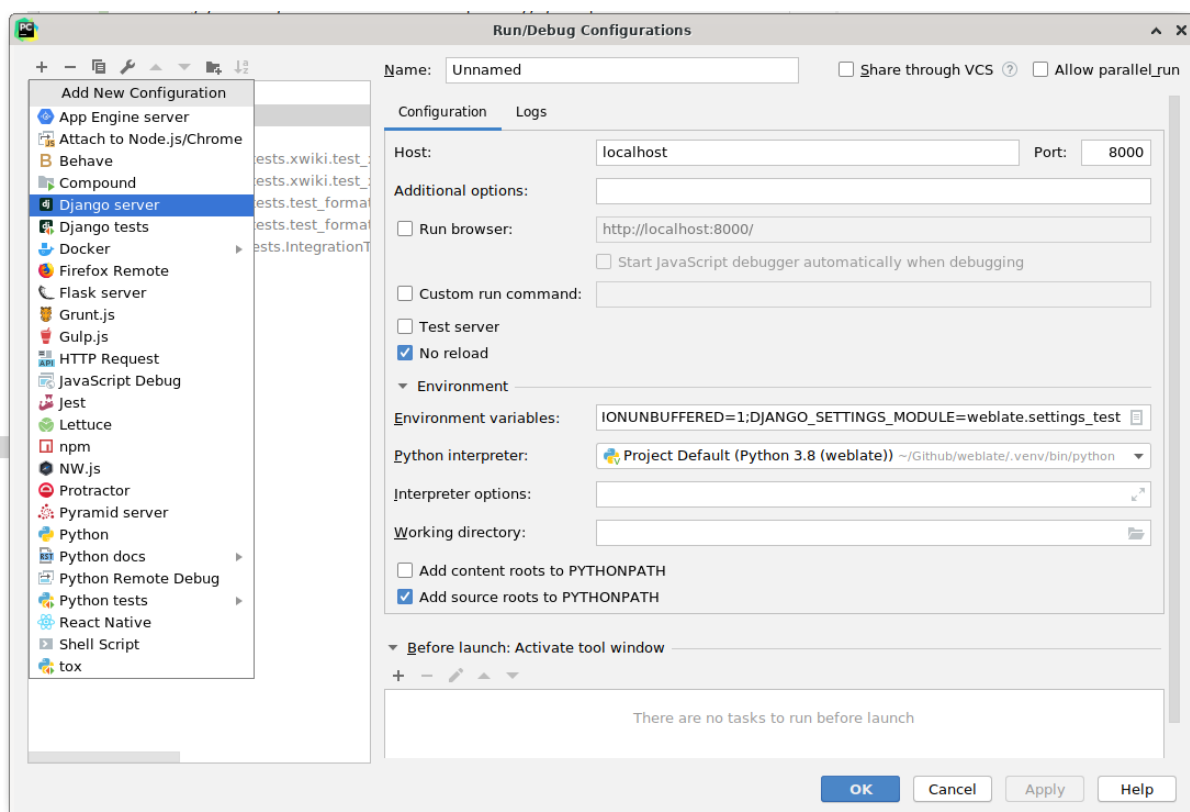
Другим кроком є належне налаштування для використання Django у PyCharm: основна проблема полягає в уможливленні негайного запуску тестів модулів у комплексному середовищі для розробки. Для цього вам слід вказати кореневий шлях до проєкту Django та шлях до його параметрів:



Будьте обережні, *корінь проекту Django* є кореневою текою репозиторію, а не підкаталогом Weblate. Щодо параметрів, ви можете скористатися *weblate/settings_test.py* з репозиторію коду, але ви можете створити власні параметри і вказати їх у відповідному місці.

Останнім кроком є уможливлення запуску сервера і розставлення точок зупинки у коді, щоб уможливити його діагностування. Це можна зробити створенням нових налаштувань *Django Server*:





Підказка: Будьте обережні із властивістю із назвою *Не перезавантажувати*: якщо ви позначите цей пункт, під час внесення змін до файлів не відбуватиметься автоматичного перезавантаження сервера. Це уможливіло збереження поточних діагностичних точок зупинення для зневадження, які відкидаються у результаті перезавантаження.

3.3 Початковий код Weblate

Розробка Weblate ведеться на [GitHub](#). Ми будемо раді, якщо ви створите відгалуження коду і надсилатимете запити щодо злиття. Будемо раді будь-яким латкам у будь-якій формі.

Дивись також:

Ознайомтеся із розділом [Нумпроуці Weblate](#), щоб дізнатися про те, як Weblate виглядає з середини.

3.3.1 Настанови щодо програмного коду

Будь-який код для Weblate має бути написано із використанням [Security by Design Principles](#) (принципів безпеки за компонуванням).

Будь-який програмний код має супроводжуватися документацією, яка пояснює його роботу. Не забувайте документувати методи, складні блоки коду або видимі користувачам можливості.

У коді має бути використано підказки щодо типів з [PEP 484](#). Ми ще не перевіряємо їх у системі неперервної інтеграції, оскільки у наявному коді їх ще немає.

3.3.2 Стандарт програмування та шліфування коду

У коді має бути виконано вимоги щодо коду PEP-8, його має бути форматовано засобом форматування **black**.

Для перевірки якості коду ви можете скористатися програмою **flake8**. Перелік рекомендованих додатків можна знайти у `.pre-commit-config.yaml`, а налаштування — у `setup.cfg`.

Найпростішим підходом для запровадження примусового виконання усіх умов є встановлення репозиторію **pre-commit**. Weblate, у якому містяться налаштування для перевірки внесених файлів на відповідність умовам. Після встановлення репозиторію (його вже включено у `requirements-lint.txt`) увімкніть його за допомогою команди `pre-commit install` у клоні репозиторію Weblate. У такий спосіб ви зможете автоматично перевіряти усі внесені вами зміни.

Ви також можете запустити перевірку вручну. Щоб перевірити усі файли:

```
pre-commit run --all
```

3.4 Діагностика у Weblate

Помилки можуть проявлятися як аварійні завершення роботи застосунку або помилкова поведінка. Ми будемо раді, якщо ви будете збирати будь-які дані про помилки та надасте їх у [нашій системі стеження за помилками](#).

3.4.1 Режим зневадження

Вмикання діагностичного режиму призведе до показу винятків у браузері. Це корисно для діагностування проблем у вебінтерфейсі, але неприйнятно для промислового середовища, оскільки призводить до погіршення швидкодії та може призвести до витоку конфіденційних даних.

У промисловому середовищі скористайтеся [ADMINS](#) для отримання повідомлень електронної пошти зі звітами щодо помилок або налаштування збирання помилок за допомогою сторонньої служби.

Дивись також:

Вимкнути режим діагностики, Належне налаштування записів адміністраторів, Збирання звітів щодо помилок

3.4.2 Журнали Weblate

Weblate може створювати докладні журнали із записами того, що відбувається за лаштунками. За типових налаштувань програма використовує журнал системи, і те, з чого складається журнал записується або до `/var/log/messages`, або до `/var/log/syslog` (залежно від налаштувань фонові служби системного журналу у вашій системі).

Процес Celery (див. [Фонові завдання з використанням Celery](#)) зазвичай створює власний журнал. У прикладі загальносистемного налаштування журнал записується до декількох файлів у каталозі `/var/log/celery/`.

Контейнери Docker записують виведені дані (це звичайна практика у царині Docker), тож ви можете ознайомитися із журналом за допомогою команди `docker-compose logs`.

Дивись також:

Зразок налаштувань містить налаштування `LOGGING`.

3.4.3 Фонові завдання не обробляються

Значна частина даних обробляється у фоновому режимі обробниками Celery. Якщо не працює надсилання повідомлень електронної пошти або вилучення складників, можливо, проблема полягає саме у роботі обробників.

Речі, які слід перевірити у цьому випадку:

- Перевірте, чи запущено процес Celery, див. *Фонові завдання з використанням Celery*
- Перевірте стан черги Celery у *Інтерфейс керування* або за допомогою `celery_queues`
- Пошукайте повідомлення про помилки у журналі Celery (див. *Журнали Weblate*)

3.4.4 Не надходять повідомлення електронної пошти від Weblate

Ви можете перевірити, чи працює вихідна пошта належним чином, за допомогою команди керування `sendtestemail` (див. *Виклик команд керування*, щоб ознайомитися із настановами щодо того, як викликати її у різних середовищах) або за допомогою *інтерфейсу керування* на вкладці *Знаряддя*.

Ці засоби надсилають повідомлення електронної пошти безпосередньо, тому це підтверджує правильність ваших налаштувань SMTP (див. *Налаштовування вихідної електронної пошти*). Втім, більша частина повідомлень від Weblate надсилається у фоновому режимі, тому проблеми можуть бути пов'язані з Celery. Опис діагностики Celery наведено у розділі *Фонові завдання не обробляються*.

3.4.5 Аналіз аварійних завершень застосунку

Якщо програма завершує роботу в аварійному режимі, корисно зібрати якомога більше відомостей щодо збою. Це можна зробити за допомогою сторонніх служб, які можуть збирати потрібні дані в автоматичному режимі. Знайти дані про налаштування системи можна у розділі *Збирання звітів щодо помилок*.

3.4.6 Помилки без повідомлень

Багато завдань покладаються на Celery для забезпечення фонові обробки. Дані аварійних завершень роботи таких завдань не буде показано в інтерфейсі користувача, а буде записано до журналу Celery. Налаштовування *збирання помилок* спростить діагностику таких помилок.

3.4.7 Проблеми із швидкодією

Якщо Weblate працює неналежним чином у певних випадках, будь ласка, зберіть відповідні дані журналів, які демонструють помилку, і усі дані, які можуть допомогти у визначенні того, як може бути удосконалено код, щоб запобігти неналежній поведінці.

Якщо певні запити виконуються надто довго без додаткової індикації, вам варто встановити `dogslow` і *Збирання звітів щодо помилок* і отримати точні та докладні трасування за допомогою засобу збирання даних помилок.

Якщо уповільнення роботи пов'язано із базою даних, ви можете увімкнути журналювання усіх запитів до бази даних за допомогою таких налаштувань після вмикання `DEBUG`:

```
LOGGING["loggers"]["django.db.backends"] = {"handlers": ["console"], "level":
↪ "DEBUG" }
```

3.5 Нутрощі Weblate

Примітка: У цьому розділі наведено базовий огляд нутрощів Weblate.

Структура більшої частини коду Weblate походить від *Django* і засновано на структурі коду Django.

3.5.1 Структура каталогів

Короткий огляд структури каталогів основного репозиторію Weblate:

docs

Початковий код цієї документації, який можна зібрати за допомогою [Sphinx](#).

dev-docker

Код Docker для запуску сервера розробки, див. [Запуск Weblate локально у Docker](#).

weblate

Вихідний код Weblate як застосунку Django <<https://www.djangoproject.com/>>, див. [Нутрощі Weblate](#).

weblate/static

Клієнтські файли (CSS, Javascript і зображення), див. [Оболонка Weblate](#).

3.5.2 Модулі

Weblate складається з декількох програм Django (деякі є необов'язковими, див. [Додаткові модулі Weblate](#)):

accounts

Обліковий запис користувача, профілі та сповіщення.

addons

Додатки для коригування поведінки Weblate, див. [Додатки](#).

api

Програмний інтерфейс на основі Django REST framework.

auth

Розпізнавання і права доступу.

billing

Додатковий модуль [Оплата](#).

checks

Модуль [Перевірки якості](#) для рядків перекладів.

fonts

Модуль перевірок обробки шрифтів.

formats

Шар абстракції форматів файлів на основі translate-toolkit.

gitexport

Додатковий модуль [Засіб експортування Git](#).

lang

Модуль визначення мов і моделей форм множини.

legal

Додатковий модуль *Правові відомості*.

machinery

Інтеграція служб комп'ютерного перекладу.

memory

Вбудована пам'ять перекладів, див. *Пам'ять перекладів*.

screenshots

Керування знінками вікон та модуль оптичного розпізнавання тексту.

trans

Головний модуль для обробки перекладів.

utils

Різнманітні допоміжні інструменти.

vcs

Абстракція системи керування версіями.

wladmin

Налаштовування адміністративного інтерфейсу Django.

3.6 Розробка додатків

За допомогою можливості *Додатки* ви можете налаштувати процедуру локалізації у Weblate.

class weblate.addons.base.BaseAddon (*storage=None*)

Базовий клас для додатків Weblate.

classmethod can_install (*component, user*)

Перевірка, чи сумісний додаток із вказаним складником.

configure (*settings*)

Збереження налаштувань.

daily (*component*)

Скрипт обробки, який вмикається щодня.

classmethod get_add_form (*user, component, **kwargs*)

Повернення форми налаштування для додавання нового додатка.

get_settings_form (*user, **kwargs*)

Повернення форми налаштування для цього додатка.

post_add (*translation*)

Скрипт обробки, який вмикається після додавання нового перекладу.

post_commit (*component*)

Скрипт, який вмикається після того, як зміни буде внесено до репозиторію.

post_push (*component*)

Скрипт, який вмикається після запису до репозиторію основної гілки розробки.

post_update (*component, previous_head: str, skip_push: bool*)

Скрипт обробки, який вмикається після оновлення репозиторію у основній гілці розробки.

Параметри

- **previous_head**(*str*) – HEAD репозиторію до оновлення, може бути порожнім для початкового клону.
- **skip_push**(*bool*) – Визначає, чи слід пропускати при обробці додатком запис до основного репозиторію коду. Зазвичай, ви можете передати цей параметр підлеглим методам як `commit_and_push` або `commit_pending`.

pre_commit(*translation, author*)

Сценарій, який вмикається, перш ніж зміни буде внесено до репозиторію.

pre_push(*component*)

Скрипт, який вмикається перед записом із репозиторію до основної гілки розробки.

pre_update(*component*)

Скрипт, який вмикається перед оновленням репозиторію з основної гілки розробки.

save_state()

Збереження відомостей щодо стану додатка.

store_post_load(*translation, store*)

Скрипт обробки, який вмикається після обробки файла.

Отримує як аргумент екземпляр класу форматування файлів.

Корисний для внесення змін до класу форматування файлів, наприклад, для коригування способу зберігання файла.

unit_pre_create(*unit*)

Скрипт обробки, який вмикається до створення нового модуля.

Ось приклад додатка:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

from django.utils.translation import gettext_lazy as _

from weblate.addons.base import BaseAddon
from weblate.addons.events import EVENT_PRE_COMMIT

class ExampleAddon(BaseAddon):
    # Filter for compatible components, every key is
    # matched against property of component
    compat = {"file_format": {"po", "po-mono"}}
    # List of events add-on should receive
    events = (EVENT_PRE_COMMIT,)
    # Add-on unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example add-on")
    # Detailed add-on description
    description = _("This add-on does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

3.7 Оболонка Weblate

Поточну версію оболонки побудовано з використанням Bootstrap, jQuery і декількох сторонніх бібліотек.

3.7.1 Підтримувані браузери

У Weblate передбачено підтримку найсвіжіших стабільних випусків усіх основних браузерів та програмних платформ.

Підтримки альтернативних браузерів, які використовують найсвіжішу версію WebKit, Blink або Gecko безпосередньо або за допомогою програмного інтерфейсу перегляду даних інтернету платформи, явним чином не передбачено. Втім, Weblate має (у більшості випадків) показувати дані і працювати належним чином і у цих браузерах.

Застарілі браузери можуть працювати, але працездатність деяких можливостей у них може бути обмежено.

3.7.2 Керування залежностями

Менеджер пакетів yarn використовується для оновлення сторонніх бібліотек. Конфігурації зберігаються в `scripts/yarn`і є скрипт обгортки :file:`scripts/yarn-update для оновлення бібліотек, збирання їх і копіювання до належних каталогів у weblate/static/vendor, де зберігається увесь код сторонніх бібліотек. Специфічний для Weblate код слід розмістити безпосередньо у weblate/static або специфічних підкаталогах (наприклад weblate/static/editor).`

Процедура додавання нової сторонньої бібліотеки, типово, складається з таких кроків:

```
# Add a yarn package
yarn --cwd scripts/yarn add PACKAGE
# Edit the script to copy package to the static folder
edit scripts/yarn-update
# Run the update script
./scripts/yarn-update
# Add files to git
git add .
```

3.7.3 Стиль програмного коду

Weblate використовує для форматування коду у файлах JavaScript і CSS [Prettier](#).

Ми також використовуємо [ESLint](#) для перевірки коду JavaScript.

3.7.4 Локалізація

Якщо вам потрібен будь-який видимий користувачеві текст у коді бібліотеки, цей текст потрібно локалізувати. У більшості випадків вам достатньо огорнути ваш текст у функцію `gettext`, але доступні й складніші можливості:

```
document.write(gettext('this is to be translated'));

var object_count = 1 // or 0, or 2, or 3, ...
s = gettext('literal for the singular case',
    'literal for the plural case', object_count);

fmts = gettext('There is %s object. Remaining: %s',
    'There are %s objects. Remaining: %s', 11);
s = interpolate(fmts, [11, 20]);
// s is 'There are 11 objects. Remaining: 20'
```

Дивись також:

Тема перекладу у документації Django

3.7.5 Піктограми

Поточна версія Weblate використовує піктограми material design. Якщо вам потрібна якась нова піктограма, скористайтесь [Material Design Icons](#) або [Material Design Resources](#).

Крім того, можна скористатися `scripts/optimize-svg` для зменшення розміру SVG, оскільки більшість піктограм вбудовуються до HTML для уможливлення застосування стилю до контурів.

3.8 Надсилання звітів щодо вад у Weblate

Систему стеження за вадами у Weblate розташовано на GitHub.

Будемо раді звітам про будь-які вади або пропозиціям щодо поліпшень у Weblate. Нами приготовано різноманітні шаблони для зручного створення звітів щодо вад.

Якщо вами було виявлено проблему із захистом у Weblate, будь ласка, ознайомтеся із розділом *Негаразди з безпекою* нижче.

Якщо ви не певні щодо вашого звіту щодо вади або запиту щодо можливості, ви можете спробувати *Обговорення у Weblate*.

3.8.1 Негаразди з безпекою

Щоб надати спільноті час для усунення вади і оновлення, наполегливо рекомендуємо вам повідомляти про вади захисту приватно. Для роботи із вадами захисту ми використовуємо HackerOne. Про них можна повідомляти безпосередньо на [HackerOne](#). Після надсилання повідомлення у спільноті буде обмежений, але достатній час для усунення проблеми.

Крім того, можете написати повідомлення на security@weblate.org. Це повідомлення буде продубльовано на HackerOne.

Якщо з якоїсь причини ви не хочете користуватися HackerOne, ви можете надіслати звіт електронною поштою до michal@weblate.org. Ви можете зашифрувати повідомлення за допомогою цього ключа PGP: `3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D`. Ви також можете отримати ключ PGP з [Keybase](#).

Примітка: Weblate залежить від сторонніх складників для виконання багатьох завдань. Якщо вами буде виявлено вразливість, яка стосується одного з цих складників загалом, будь ласка, повідомте про неї розробникам відповідного проекту.

Ось деякі з цих складників:

- [Django](#)
 - [Бібліотека REST Django](#)
 - [Python Social Auth](#)
-

3.9 Комплекс тестування Weblate і неперервна інтеграція

Комплекси тестування існують для більшої частини поточного коду. Збільште покриття додаванням тестувань для будь-яких нових функціональних можливостей і перевірте, чи працює тестування.

3.9.1 Неперервна інтеграція

Із поточними результатами тестування можна ознайомитися на сторінці [GitHub Actions](#) і покриття зі звіту [Codecov](#).

Передбачено декілька завдань для перевірки різних аспектів:

- Перевірка модулів
- Збирання документації та зовнішні посилання
- Тестування перенесення для всіх підтримуваних випусків
- Аналіз коду
- Перевірка конфігурації (забезпечує наявність у створених файлах усіх даних та можливість їх тестування)

Налаштування для системи неперервної інтеграції зберігаються у каталозі `.github/workflows`. Система неперервної інтеграції значною частиною використовує допоміжні скрипти, які зберігаються у каталозі `ci`. Ці скрипти може бути виконано вручну, але для їхньої роботи потрібні декілька змінних середовища, які, здебільшого, визначають файл параметрів Django, яким слід скористатися, та параметр з'єднання із базою даних. Приклад визначення цих змінних можна знайти у файлі `scripts/test-database.sh`:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

# Simple way to configure test database from environment

# shellcheck shell=sh

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE="${1:-postgresql}"

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

Просте виконання може виглядати так:

```
. scripts/test-database.sh
./ci/run-migrate
./ci/run-test
./ci/run-docs
```

3.9.2 Локальне тестування

Щоб запустити комплекс тестів локально, скористайтеся такими командами:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

Підказка: Для тестів вам знадобиться сервер бази даних (PostgreSQL). Типово, Django створює окрему базу даних для запуску тестів. У назві цієї бази даних міститься префікс `test_`. Отже, якщо параметри налаштовано на використання `weblate`, для тестування буде використано базу даних `test_weblate`. Див. *Налаштування бази даних для Weblate*, щоб дізнатися більше про налаштування.

Файл `weblate/settings_test.py` також використовується у середовищі неперервної інтеграції (див. *Неперервна інтеграція*). Його роботу можна скоригувати за допомогою змінних середовища:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

# Simple way to configure test database from environment

# shellcheck shell=sh

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE="${1:-postgresql}"

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

Перш ніж запускати тести, вам слід зібрати статичні файли, оскільки робота деяких тестів залежить від наявності цих файлів:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

Ви також можете вказати окремі тести, які слід виконати:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

Підказка: Тести також можна запускати у межах контейнера `docker` розробника, див. *Запуск Weblate локально у Docker*.

Дивись також:

Див. *Testing in Django*, щоб дізнатися більше про запуск і написання тестів для Django.

3.10 Схеми даних

Weblate використовує *схему JSON* для визначення компонування системи зовнішніх файлів JSON.

3.10.1 Схеми пам'яті перекладів Weblate

https://weblate.org/schemas/weblate-memory.schema.json	
тип	<i>array</i>
записи	<i>Запис пам'яті перекладів</i>
тип	<i>object</i>
властивості	<ul style="list-style-type: none"> category <i>Категорія рядків</i> 1 — загальний, 2 — спільний, 10000000+ — специфічний для проєкту, 20000000+ — специфічний для користувача тип <i>integer</i> приклади 1 мінімум 0 типове значення 1 origin <i>Походження рядка</i> Назва файлу або складника тип <i>рядок</i> приклади test.tmx проєкт/складник типове значення source <i>Початковий рядок</i> тип <i>рядок</i> приклади Hello minLength 1 типове значення source_language <i>Початкова мова</i> ISO 639-1 / ISO 639-2 / IETF BCP 47 тип <i>рядок</i> приклади en взірець <code>^[^]+\$</code> типове значення target <i>Рядок перекладу</i> тип <i>рядок</i> приклади Ahoj minLength 1 типове значення target_language <i>Мова перекладу</i> ISO 639-1 / ISO 639-2 / IETF BCP 47 тип <i>рядок</i> приклади cs взірець <code>^[^]+\$</code> типове значення
additionalProperties	False
визначення	

Дивись також:

Пам'ять перекладів, *dump_memory*, *import_memory*

3.10.2 Експорт даних користувача Weblate

https://weblate.org/schemas/weblate-userdata.schema.json	
тип	<i>object</i>
властивості	
• basic	<i>Основа</i>
тип	<i>object</i>
властивості	
• username	<i>Ім'я користувача</i>
тип	<i>рядок</i>
приклади	<code>admin</code>
типове значення	
• full_name	<i>Повне ім'я</i>
тип	<i>рядок</i>
приклади	<code>Адміністратор Weblate</code>
типове значення	
• email	<i>Електронна пошта</i>
тип	<i>рядок</i>
приклади	<code>noreply@example.com</code>
типове значення	
format	<code>email</code>
• date_joined	<i>Дата приєднання</i>
тип	<i>рядок</i>
приклади	<code>2019-11-18T18:53:54.862Z</code>
типове значення	
format	<code>date-time</code>
additionalProperties	<code>False</code>
• profile	<i>Профіль</i>
тип	<i>object</i>
властивості	
• language	<i>Мова</i>
тип	<i>рядок</i>
приклади	<code>cs</code>
взірець	<code>^[^]*\$</code>
типове значення	
• suggested	<i>Кількість рядків-пропозицій</i>
тип	<i>integer</i>
приклади	<code>1</code>
типове значення	<code>0</code>
• translated	<i>Кількість перекладених рядків</i>
тип	<i>integer</i>
приклади	<code>24</code>
типове значення	<code>0</code>
• uploaded	<i>Кількість вивантажених знімків вікон</i>
тип	<i>integer</i>
приклади	<code>1</code>
типове значення	<code>0</code>
• hide_completed_translations	<i>Приховати завершені переклади у домівці</i>
тип	<i>boolean</i>
приклади	<code>False</code>
типове значення	<code>True</code>
• show_untranslated_strings	<i>Показувати переклади іншими мовами в режимі Дзен</i>
тип	<i>boolean</i>
приклади	<code>True</code>
типове значення	<code>True</code>

continues on next page

Table 2 – continued from previous page

• hide_source_s	<i>Приховати джерельний текст, коли є переклад будь-якою іншою мовою</i>		
	тип	<i>boolean</i>	
	приклади	False	
	типове значення	True	
• editor_link	<i>Посилання на редактор</i>		
	тип	<i>рядок</i>	
	приклади		
	взірець	<i>^.*\$</i>	
• translate_mod	<i>Режим редактора перекладу</i>		
	тип	<i>integer</i>	
	приклади	0	
	типове значення	0	
• zen_mode	<i>Режим Дзен-редактора</i>		
	тип	<i>integer</i>	
	приклади	0	
	типове значення	0	
• special_chars	<i>Спеціальні символи</i>		
	тип	<i>рядок</i>	
	приклади		
	взірець	<i>^.*\$</i>	
• dashboard_vie	<i>Типовий вигляд домівки</i>		
	тип	<i>integer</i>	
	приклади	1	
	типове значення	0	
• dashboard_co	<i>Типовий список складників</i>		
	типове значення	null	
	anyOf	тип	<i>null</i>
		тип	<i>integer</i>
• languages	<i>Перекладено мовами</i>		
	тип	<i>array</i>	
	типове значення		
	записи	<i>Код мови</i>	
		тип	<i>рядок</i>
		приклади	cs
• secondary_lan	<i>Інші мови</i>		
	тип	<i>array</i>	
	типове значення		
	записи	<i>Код мови</i>	
		тип	<i>рядок</i>
		приклади	sk
• watched	<i>Відстежувані проекти</i>		
	тип	<i>array</i>	
	типове значення		
	записи	<i>Ідентифікатор проекту</i>	
		тип	<i>рядок</i>
		приклади	weblate
		взірець	<i>^.*\$</i>
		типове значення	
additionalProperties	False		

continues on next page

Table 2 – continued from previous page

• auditlog	<i>Часопис перевірок</i>	
	тип	<i>array</i>
	типове значення	
	записи	<i>Елементи</i>
	тип	<i>object</i>
	властивості	
	• address	<i>IP-адреса</i>
	тип	<i>рядок</i>
	приклади	127.0.0.1
	взірець	<i>^.*\$</i>
	типове значення	
	• user_agent	<i>User agent</i>
	тип	<i>рядок</i>
	приклади	PC / Linux / Firefox 70.0
	взірець	<i>^.*\$</i>
	типове значення	
	• timestamp	<i>Часова позначка</i>
	тип	<i>рядок</i>
	приклади	2019-11- 18T18:58:30.845Z
	типове значення	
	format	date-time
	• activity	<i>Діяльність</i>
	тип	<i>рядок</i>
	приклади	login
	взірець	<i>^.*\$</i>
	типове значення	
	additionalProperties	False
визначення		

Дивись також:*Профіль користувача, `dumpuserdata`*

3.11 Створення випусків Weblate

3.11.1 Планування випусків

У Weblate передбачено двомісячний цикл для випусків (x.y). За основними випусками, зазвичай, слідують випуски із виправленнями вад, які мають додаткову нумерацію (x.y.z).

Зміна основної версії означає, що у процесі оновлення не можна пропустити проміжну версію — вам завжди слід спочатку оновитись до версії x.0, а вже потім оновлювати їх до новіших випусків x.y.

Дивись також:*Оновлення Weblate*

3.11.2 Планування випусків

Можливості нових випусків збираються у етапах (milestones) GitHub. Ознайомитися із нашим списком змін можна тут: <<https://github.com/WeblateOrg/weblate/milestones>>.

3.11.3 Процедура випусків

Речі, які слід перевірити перед випуском:

1. Пошукати нові мови перекладу за допомогою команди `./scripts/list-translated-languages`.
2. Встановити остаточну версію за допомогою команди `./scripts/prepare-release`.
3. Переконаватися, що знімки вікон є актуальними за допомогою команди `make -j 12 -C docs update-screenshots`.
4. Злити усі можливі переклади з черги `wlc push; git remote update; git merge origin/weblate`

Створити випуск:

5. Створити випуск за допомогою команди `./scripts/create-release --tag` (нижче наведено вимоги).

Після випуску виконати вручну:

6. Оновити образ Docker.
7. Закрити етап на GitHub.
8. Щойно образ Docker буде перевірено, додати мітку і записати його до репозиторію.
9. Оновити до нової версії діаграму Helm.
10. Включити нову версію до `.github/workflows/migrations.yml`, щоб її було оброблено засобами тестування перенесення.
11. Збільште номер версії у посиланнях на отримання пакунків на сайті.
12. Збільшити номер версії у репозиторії за допомогою команди `./scripts/set-version`.
13. Перевірте, чи зібрано `readthedocs.org` усі переклади документації, за допомогою `./scripts/rtd-projects`.

Для створення міток за допомогою скрипту `./scripts/create-release` вам знадобиться:

- GnuPG із закритим ключем, який використовується для підписування випусків
- Доступ до запису змін до репозиторіїв git Weblate (скрипт записує мітки)
- Налаштована програма **hub** для доступу і створення випусків у репозиторії Weblate
- SSH-доступ до сервера отримання пакунків Weblate (туди буде скопійовано пакунки Website)

3.12 Безпека і конфіденційність

Порада: У Weblate ми серйозно ставимося до безпеки і прагнемо зберегти середовище, яке цінує безпеку і конфіденційність наших користувачів.

Процес розробки Weblate відповідає [найкращим практикам ініціативи базової інфраструктури Linux Foundation](#).

Дивись також:

Негаразди з безпекою

3.12.1 Оновлення захисту

Оновлення захисту гарантовано отримуватиме лише найсвіжіший випуск.

3.12.2 Стеження за вразливостями у залежностях

Ми стежимо за вадами у захисті наших залежностей за допомогою [Dependabot](#). Перевірки стосуються бібліотек Python і JavaScript, а найсвіжіший стабільний випуск має містити кориговані залежності з метою уникнення вразливостей.

Підказка: Можуть траплятися вразливості у сторонніх бібліотеках, які не впливають на роботу Weblate. Ми не намагаємося окремо усунути ці вразливості у випусках із виправленнями вад Weblate.

3.12.3 Захист контейнерів Docker

Сканування контейнерів Docker виконується регулярно за допомогою сканерів безпеки [Anchore](#) і [Trivy](#).

Це надає нам змогу виявити вразливості на ранньому етапі і швидко випустити удосконалену версію.

Результати сканувань розміщено на [GitHub](#) — вони зберігаються як запис у нашій системі неперервної інтеграції у форматі Static Analysis Results Interchange Format (SARIF).

Дивись також:

Неперервна інтеграція

3.13 Участь у розробці модулів Weblate

Окрім основного сховища, Weblate складається з декількох модулів Python. У всіх цих модулях використано однакову структуру, а ця документація стосується їх усіх.

Це, наприклад, стосується таких компонентів:

- [wlc](#), клієнтська бібліотека Python, див. *Клієнт Weblate*
- [translation-finder](#), використано для виявлення придатних до перекладу файлів у сховищі
- [language-data](#), визначення мови для Weblate, див. *Визначення мов*

3.13.1 Настанови щодо програмного коду

Будь-який код для Weblate має бути написано із використанням [Security by Design Principles](#) (принципів безпеки за компонуванням).

Будь-який програмний код має супроводжуватися документацією, яка пояснює його роботу. Не забувайте документувати методи, складні блоки коду або видимі користувачам можливості.

У коді має бути використано підказки щодо типів з [PEP 484](#). Ми ще не перевіряємо їх у системі неперервної інтеграції, оскільки у наявному коді їх ще немає.

3.13.2 Виконання перевірок

Перевірки буде виконано за допомогою **py.test**. Спершу, вам слід встановити обов'язкові складники для тестування:

```
pip install -r requirements-test.txt
```

Далі, ви можете виконати комплекс перевірок для локальної копії сховища:

```
py.test
```

Дивись також:

Інтеграція із системою неперервної інтеграції дуже подібна до *Комплекс тестування Weblate і неперервна інтеграція*.

3.13.3 Стандарт програмування та шліфування коду

У коді має бути виконано вимоги щодо коду PEP-8, його має бути сформатовано засобом форматування **black**.

Для перевірки якості коду ви можете скористатися програмою **flake8**. Перелік рекомендованих додатків можна знайти у `.pre-commit-config.yaml`, а налаштування — у `setup.cfg`.

Найпростішим підходом для запровадження примусового виконання усіх умов є встановлення репозиторію **pre-commit**. Weblate, у якому містяться налаштування для перевірки внесених файлів на відповідність умовам. Після встановлення репозиторію (його вже включено у `requirements-lint.txt`) увімкніть його за допомогою команди `pre-commit install` у клоні репозиторію Weblate. У такий спосіб ви зможете автоматично перевіряти усі внесені вами зміни.

Ви також можете запустити перевірку вручну. Щоб перевірити усі файли:

```
pre-commit run --all
```

Дивись також:

Початковий код Weblate

3.14 Про Weblate

3.14.1 Цілі проєкту

Заснований на інтернет-технологіях інструмент локалізації із тісною прив'язкою до *систем керування версіями*, підтримкою широкого діапазону *форматів файлів*. Спрощує доступ до перекладів для перекладачів.

3.14.2 Назва проєкту

«Weblate» є словом-гібридом від слів «web» («мережа») та «translate» («перекладати»).

3.14.3 Вебсайт проекту

Основною сторінкою інструмента є <https://weblate.org/>, а власну «хмарну» службу розташовано за адресою <https://hosted.weblate.org/>. Відповідну документацію розміщено на сайті <https://docs.weblate.org/>.

3.14.4 Логотипи проекту

Логотипи проекту та інші графічні роботи можна знайти у репозиторії <https://github.com/WeblateOrg/graphics>.

3.14.5 Лідерство

Супровід цього проекту виконується Міхалом Чихаржем (Michal Čihař) michal@weblate.org.

3.14.6 Автори

Розробку Weblate було розпочато Міхалом Чихаржем (Michal Čihař). З моменту свого створення у 2012 році, тисячі людей зробили свій внесок.

3.15 Ліцензія

Докладніші відомості щодо ліцензування наведено у коді Weblate, вони відповідають [специфікації REUSE 3.0](#).

© Michal Čihař michal@weblate.org

Ця програма є вільною. Ви можете поширювати та змінювати її на умовах, викладених у GNU General Public License, що видана Free Software Foundation; або версії 3 ліцензії, або (на ваш розсуд) довільної старшої версії.

Ця програма розповсюджується з надією, що буде корисною, але БЕЗ БУДЬ-ЯКИХ ГАРАНТІЙ; навіть без явної гарантії КОМЕРЦІЙНОЇ СПРОМОЖНОСТІ або ВІДПОВІДНОСТІ ЗАДАНИЙ МЕТІ. Докладніше дивіться General Public License.

Ви мали отримати копію GNU General Public License разом з %s. Якщо цього не сталося, дивіться [<http://www.gnu.org/licenses/>](http://www.gnu.org/licenses/).

4.1 Weblate 4.16.4

Released on March 16th 2023.

- Dependencies updates.
- Improved background tasks scheduling.

[All changes in detail.](#)

4.2 Weblate 4.16.3

Released on March 15th 2023.

- Improved session handling with project backups.
- Dependencies updates.
- Оновлення локалізації.
- Удосконалення у документації.

[All changes in detail.](#)

4.3 Weblate 4.16.2

Випущено 8 березня 2023 року.

- Виправлено пошук у пам'яті перекладів.
- Виправлено автоматичний переклад для більшої кількості служб.
- Удосконалено показ збігів термінів глосарія, що перекриваються.
- Фіксований аналіз множини для неанглійської початкової мови в деяких форматах.
- Додано підтримку формату файлів JSON go-i18n.

[Докладний опис усіх змін.](#)

4.4 Weblate 4.16.1

Випущено 1 березня 2023 року.

- Виправлено помилку testsuite.

[Докладний опис усіх змін.](#)

4.5 Weblate 4.16

Випущено 1 березня 2023 року.

- У новій версії перевірки форматування рядків виявляють дубльовані формати.
- Удосконалено швидкодію пошуку для деяких спеціально форматуваних рядків.
- У новій версії биття Celery зберігає план завдань у базі даних.
- Додано підтримку перекладача IBM Watson.
- Відкинуто підтримку параметрів інтеграції з системами керування версіями, яку визнано застарілою у 4.14.
- Додано підтримку запитів щодо злиття для сервера Bitbucket.
- Удосконалено обробку конфліктів у файлах PO gettext.
- Додано підтримку визначення стану рядків при додаванні за допомогою програмного інтерфейсу.
- Додано підтримку налаштування дозволених джерел CORS.
- Додано підтримку форм множини для автоматичних пропозицій.

[Докладний опис усіх змін.](#)

4.6 Weblate 4.15.2

Випущено 25 січня 2023 року.

- Уможливлено формати gotext JSON і i18next v4 у типовій конфігурації.
- Виправлено аварійне завершення роботи при вивантаженні пошкоджених файлів.
- Реалізовано показ незмінних каталогів у стані сховища Git.

[Докладний опис усіх змін.](#)

4.7 Weblate 4.15.1

Випущено 19 січня 2023 року.

- Виправлено пропозиції автоматичного перекладу.
- Виправлено аварійне завершення роботи сторінки додатків у деяких рідкісних випадках.
- Виправлено неможливість перекладу шаблону для нових перекладів у деяких випадках.
- Документовано ліцензування з використанням [REUSE 3.0](#).
- Виправлено поділ списку користувачів при керуванні командами.
- Удосконалено швидкодію створення і збереження проєктів.
- Додано підтримку формату файлів JSON gotext.

- Додано підтримку файлів i18next v4.
- У новій версії можна налаштувати поділ на сторінки за допомогою програмного інтерфейсу.

[Докладний опис усіх змін.](#)

4.8 Weblate 4.15

Випущено 16 грудня 2022 року.

- Додано підтримку навігації змінами для окремого рядка.
- Виправлено обробку форм множини при автоматичному перекладі з інших складників.
- Додано клавіатурне скорочення Alt+Enter для надсилання рядка як пропозиції.
- Додано підтримку заміників у форматі файлів Fluent.
- Удосконалено швидкодію пам'яті перекладу.
- Реалізовано автоматичне створення посилань навігації у сховищі коду для відомих служб зберігання коду.
- Поліпшено швидкодію для декількох режимів перегляду.
- Удосконалено показ списків рядків із формами множини.
- Додано підтримку додавання нетипової розмітки у заголовки HTML.
- Виправлено створення файлів MO у додатку — реалізовано включення лише перекладених файлів.
- Виправлено показ прапорців формальних виразів.
- Удосконалено поведінку перевірки на заміник у формах множини.
- Додано підтримку іменування файлів перекладу, яку використовують на Google Play.
- Додано підтримку міток у програмному інтерфейсі.
- Додано підтримку вибору іншої адреси електронної пошти, ніж адреса для сповіщень.
- У образі для Docker більше не встановлюється типовий режим діагностики.
- Реалізовано упорядкування термінів глосарія на основі пріоритетності складника глосарія.
- Додано адміністраторів команд, які можуть додавати учасників до команди або вилучати їх з неї.
- Додано контекстну панель підтвердження при вилученні записів користувачів.
- Додано додаток для налаштування виведення XML.

[Докладний опис усіх змін.](#)

4.9 Weblate 4.14.2

Випущено 5 листопада 2022 року.

- Додано підтримку вилучення записів з пам'яті перекладів проєкту.
- Удосконалено аналіз при попередженні про дублювання мов.
- Удосконалено точність перевірки на послідовне дублювання слів.
- Удосконалено масштабування за одночасного надсилання багатьох сповіщень.
- Удосконалено обробку станів рядків для перекладу субтитрів.
- Визнано застарілими небезпечні налаштування служб керування версіями за допомогою ключів програмного інтерфейсу та `_TOKEN/_USERNAME` замість `_CREDENTIALS` list.

- Виправлено обробку деяких вивантажених файлів CSV.
- Удосконалено обробку змін у пробілах під час показу розбіжностей.
- На сторінки керування додано посилання на керування автоматичними пропозиціями.
- Реалізовано вилучення/видалення коментарів у журналі.
- Виправлено відновлення проєктів з пов'язаними складниками з резервних копій.
- Виправлено введення CAPTCHA при невдалій реєстрації.
- Удосконалено обробку мов у DeepL.
- Удосконалено сумісність скриптів спостереження зі сховищами із розпізнаванням.
- Додано підтримку Python 3.11.

[Докладний опис усіх змін.](#)

4.10 Weblate 4.14.1

Випущено 15 вересня 2022 року.

- Виправлено генерування резервних копій проєкту в деяких ситуаціях.
- Удосконалено звітування про помилки під час вивантаження файлів.
- Отримуйте всі перевірені користувачем електронні листи з GitHub під час автентифікації.
- Додано уникнення зіставлення термінів глосарія з контекстом або ключами.
- Додано сповіщення про вилучення рядків.
- Удосконалено керування неперекладними термінами в глосарії.
- Відтепер можна вказувати кількість учасників команди на сторінці керування командою.
- Додано інтерфейс керування групою.
- Реалізовано показ статистики відгуків, коли ввімкнуто рецензування.
- Додано підтримку пошуку у програмному інтерфейсі модулів.
- Виправлено показ смужки поступу для придатних лише для читання рядків у процесі рецензування.
- Удосконалено перевірки пунктуації бірманською мовою.
- Виправлено збирання мотлоху у даних метрики.

[Докладний опис усіх змін.](#)

4.11 Weblate 4.14

Випущено 22 серпня 2022 року.

- Реалізовано стеження за змінами у додатках у журналі.
- Виправлено обробку перекладів з файлів Windows RC, HTML та текстових файлів.
- Розширено параметри налаштування стилю коду мов.
- Додано підтримку оновлених форм множини з нещодавніх випусків CLDR.
- Зменшено використання пам'яті під час оновлення складників із багатьма перекладами.
- Додано підтримку домену перекладу у SAP Translation Hub.
- Уможливлено абсолютні посилання для місця розташування початкового рядка.

- Удосконалено роботу за деякими оберненими проксі-серверами.
- Програмний інтерфейс розширено для роботи з пам'яттю перекладів.
- Удосконалено процедуру перекладу документів.
- Удосконалено надійність перекладу файлів HTML та текстових файлів.
- Додано підтримку резервного копіювання на рівні проєктів.
- Удосконалено швидкодію та використання пам'яті для пошуку в пам'яті перекладів.

[Докладний опис усіх змін.](#)

4.12 Weblate 4.13.1

Випущено 1 липня 2022 року.

- Виправлено стеження за пропозиціями у журналі.
- Виправлено обробку даних зворотного проксі з Cloudflare.
- Реалізовано блокування перекладу складника при помилці обробки даних.
- Виправлено налаштування проміжного файла у додатку виявлення.
- Виправлено поведінку перекладів DeepL з символами-замінниками.
- Усунено проблему із непридатними до перекладу рядками у програмному інтерфейсі.
- Додано підтримку вилучення користувача з групи за допомогою програмного інтерфейсу.
- Виправлено журнал аудиту щодо повідомлень електронної пошти користувачам.
- Виправлено назви прапорців для рядків форматування Java.

[Докладний опис усіх змін.](#)

4.13 Weblate 4.13

Випущено 15 червня 2022 року.

- Змінено поведінку при оновленні назв мов.
- Додано поділ на сторінки для списку проєктів.
- Програмний інтерфейс для створення модулів у новій версії повертає відомості щодо новоствореного модуля.
- У виявленні складників у новій версії передбачено підтримку налаштування проміжної мови.
- Додано варіанти фіксованого кодування для форматів CSV.
- Змінено обробку контексту та місця для деяких форматів з метою удосконалення сумісності із базовою реалізацією.
- Додано підтримку формату ResourceDictionary.
- Удосконалено кольори смужки поступу для людей із вадами зору.
- Виправлено чищення варіантів при вилученні рядка.
- Забезпечено сумісність із Django 4.1.
- Додано підтримку зберігання екранованих елементів XML у XLIFF.
- Удосконалено форматування повідомлень про помилки при перевірці замінників.
- Переспрямовано /.well-known/change-password до /accounts/password/.

- Служби комп'ютерного перекладу у новій версії можна налаштовувати на рівні проєкту.
- Додано окремі права доступу для обробки коментарів і передано їх до ролі *Рецензування рядків*.
- Додано підтримку зберігання альтернативних перекладів у файлах CSV.
- Реалізовано можливість ігнорування регістру символів під час перевірки символів-замінників.

[Докладний опис усіх змін.](#)

4.14 Weblate 4.12.2

Випущено 11 травня 2022 року.

- виправлено повторну побудову пам'яті перекладів проєкту для деяких складників.
- виправлено упорядкування складників за кількістю неперекладених рядків.
- усунено можливу втрату перекладів під час додавання нової мови.
- забезпечено створення ключа SSH Weblate під час перенесення даних.

[Докладний опис усіх змін.](#)

4.15 Weblate 4.12.1

Випущено 29 квітня 2022 року.

- виправлено заголовок повідомлення щодо запитів щодо злиття.
- удосконалено обробку синтаксичних помилок у форматі Fluent.
- виправлено показ аватару у повідомленнях-сповіщеннях електронної пошти.
- додано підтримку вебмонетизації.
- усунено вилучення застарілих початкових рядків при вилученні перекладів.

[Докладний опис усіх змін.](#)

4.16 Weblate 4.12

Випущено 20 квітня 2022 року.

- додано підтримку амхарської у *Невідповідність кінцевої крапки*.
- додано підтримку бірманської у *Невідповідність знаків питання*.
- розширено параметр додатка *Створення псевдолокалей*.
- додано прапорець `ignore-all-checks` для ігнорування перевірок якості для рядка.
- усунено можливість для додатка *Створення псевдолокалей* ініціювати помилки у перевірках.
- додано підтримку *Запити щодо злиття Gitea*.
- додано код мови у стилі Linux для *Стиль коду мови*.
- додано підтримку для перебудови пам'яті перекладів проєкту.
- удосконалено програмний інтерфейс для створення компонентів з файла.
- додано кнопки копіювання та клонування до інших перекладів.
- уможливлено налаштування повідомлень щодо запиту щодо об'єднання на рівні складників.

- Удосконалено поведінку обмеження щодо максимальної довжини з тегами XML.
- Виправлено завантаження файлів Fluent із додатковими коментарями.

[Докладний опис усіх змін.](#)

4.17 Weblate 4.11.2

Випущено 4 березня 2022 року.

- Виправлено файли МО у двійковому випуску.

[Докладний опис усіх змін.](#)

4.18 Weblate 4.11.1

Випущено 4 березня 2022 року.

- Виправлено пропущене очищення аргументів для Git і Mercurial - CVE-2022-23915, див. [GHSA-3872-f48p-pxqj](#), щоб дізнатися більше.
- Виправлено завантаження неточно перекладених рядків з файлів CSV.
- Додано підтримку створення команд за допомогою програмного інтерфейсу.
- Виправлено показ пропозицій із згадуванням користувача.
- Уможливлено налаштування доступу до жетонів проєкту.

[Докладний опис усіх змін.](#)

4.19 Weblate 4.11

Випущено 25 лютого 2022 року.

- Виправлено XSS - CVE-2022-24710, див. [GHSA-6jp6-9rf9-gc66](#), щоб дізнатися більше.
- Виправлено встановлення додатків за допомогою програмного інтерфейсу.
- *Рядки, що потребують уваги* перейменовано на *Незавершені рядки*.
- Усунено помилкові спрацювання засобу перевірки *Синтаксис ICU MessageFormat*.
- Реалізовано показ блокування і угоду учасника розробки у списку інших випадків.
- Виправлено оновлення файлів PO із застарілими рядками або пропущеними формами множини.
- Удосконалено сумісність додатка squash із Gerrit.
- Реалізовано автоматичну ініціалізацію мов користувача на основі заголовка *Accept-Language*.
- Удосконалено обробку помилок при вилученні рядка.
- Для роботи Weblate з цієї версії потрібен Python 3.7 або новіша версія.
- Виправлено деякі дії із запису при розпізнаванні у проєкті за допомогою жетонів.
- Виправлено стеження за станом рядка, коли до рядка внесено зміни у репозиторії.
- Реалізовано стеження за змінами рядка у репозиторії.
- Реалізовано липкий заголовок у списках перекладів для удосконалення навігації.
- Усунено проблему із непридатними до перекладу рядками у *Властивості Java*.
- Виправлено роботу з Git для гілок, назви яких містять символи по за ASCII.

- Новий додаток *Заповнити переклади початковими рядками*.
- Додано *Стиль злиття Об'єднати без швидкого переспрямування*.
- Виправлено вмикання додатка *Автоматичний переклад* для нових доданих рядків.
- Удосконалено перевірки пунктуації бірманською мовою.
- Додано підтримку визначення нетипових команд на рівні проєкту для надання користувачам доступу, див. *Керування правами доступу на рівні проєкту*.
- До попереджень додано посилання на документацію.
- У новій версії контейнер Docker автоматично вмикає TLS/SSL для вихідної електронної пошти, коли це потрібно.
- Додано підтримку пошуку вирішених коментарів щодо помилок.
- Додано підтримку borgbackup 1.2.
- Виправлено застосування мітки *Перекладено автоматично*.

Докладний опис усіх змін.

4.20 Weblate 4.10.1

Випущено 22 грудня 2021 року.

- Документовані зміни впроваджено як наслідок оновлення до Django 4.0.
- Виправлено показ мітки *Перекладено автоматично*.
- Виправлено показ програмного інтерфейсу гілки у складниках із спільним сховищем коду.
- Удосконалено аналіз при попередженні про невдалий запис до сховища.
- Виправлено редагування сторінки вручну при зміні навігації.
- Удосконалено точність засобу *Використано літеру кашида*.
- У контейнері Weblate для Docker нової версії використано Python 3.10.

Докладний опис усіх змін.

4.21 Weblate 4.10

Випущено 16 грудня 2021 року.

- Додано підтримку формальності і заміників з використанням DeepL.
- У новій версії пакетним редагуванням і пошуком із заміною можна скористатися на рівнях проєкту та мови.
- Додано фільтрування для пошуку із заміною.
- Виправлено: привілеї «Виконувати автоматичний переклад» більше не є частиною групи *Мови*.
- «Виконувати автоматичний переклад» тепер у групі *Адміністрування* і новій групі *Автоматичний переклад*.
- Виправлено створення файлів XLSX зі спеціальними символами.
- До модуля розпізнавання на Github додано можливість перевіряти, чи належить користувач до вказаної організації або команди GitHub.
- Удосконалено повідомлення щодо некоректних параметрів, які було передано програмному інтерфейсу.

- До програмного інтерфейсу додано підтримку жетонів доступу на рівні проєкту.
- Виправлено вилучення рядка у деяких випадках.
- Виправлено переклад нових доданих рядків.
- Реалізовано автоматичне додавання мітки до перекладених автоматично рядків, щоб спростити їхнє фільтрування.

[Докладний опис усіх змін.](#)

4.22 Weblate 4.9.1

Випущено 19 листопада 2021 року.

- Виправлено вивантаження одномовних файлів після зміни шаблону.
- Удосконалено обробку пробілів у прапорцях.
- Додано підтримку фільтрування у програмному інтерфейсі отримання даних.
- Виправлено показ статистичних даних при додаванні нових перекладів.
- Усунено проблеми зі зміною ключів SSH GitHub.

[Докладний опис усіх змін.](#)

4.23 Weblate 4.9

Випущено 10 листопада 2021 року.

- Реалізовано надання ширших відомостей про події у журналі.
- Удосконалено показ журналу.
- Удосконалено швидкодію обробки на сторінках перекладу.
- Додано підтримку обмеження отримання файлів перекладу.
- Нова версія `safe-html` може обробляти Markdown, якщо його використано з `md-text`.
- У новій версії використання `tegy max-length` призводить до ігнорування розмітки XML, якщо його використано з `xml-text`.
- Виправлено обчислення розмірностей обробленого тексту у *Найбільший розмір перекладу*.
- Скорочено довжину заголовка крамниці програм до 30 з метою допомоги у впровадженні майбутніх змін у правилах Google.
- Додано підтримку налаштування виклику SSH за допомогою `SSH_EXTRA_ARGS`.
- Додано перевірки для MessageFormat ICU.
- Удосконалено обробку умов помилки у модулях комп'ютерного перекладу.
- Реалізовано підсвічування нестандартних символів пробілів у рядках.
- Додано параметр для залишення на перекладеному рядку під час редагування.
- Додано підтримку налаштування виклику Borg за допомогою `BORG_EXTRA_ARGS`.
- Виправлено створення файлів МО для одномовних перекладів.
- Додано кінцеву точку програмного інтерфейсу для отримання усіх перекладів складника у форматі файлу ZIP.
- Додано підтримку Python 3.10.
- Додано підтримку повторного надсилання запрошення електронною поштою в інтерфейсі керування.

Докладний опис усіх змін.

4.24 Weblate 4.8.1

Випущено 10 вересня 2021 року.

- Виправлено вилучення запису користувача в адміністративному інтерфейсі Django.
- Докладніше документування параметрів додатків.
- Виправлено помилку JavaScript при роботі з глосарієм.
- Додано обмеження на кількість відповідників у перевірці узгодженості.
- Удосконалено обробку заміників у комп'ютерних перекладах.
- Виправлено створення додатків за допомогою програмного інтерфейсу.
- До нижнього колонтитула сторінки додано параметр `PRIVACY_URL` для додавання посилання на правила конфіденційності.
- Реалізовано приховування адрес електронної пошти учасників від адміністраторів проєкту.
- Удосконалено об'єднання PO gettext, якщо виявлено конфлікти.
- Удосконалено підсвічування у глосарії.
- Удосконалено поведінку прапорця `safe-html` при використанні перевірок XML.
- Виправлено повідомлення про внески для пов'язаних складників.

Докладний опис усіх змін.

4.25 Weblate 4.8

Випущено 21 серпня 2021 року.

- Додано підтримку формату Apple stringsdict.
- Оператор точного пошуку у новій версії відтепер враховує регістр символів для PostgreSQL.
- Виправлено збереження пояснень глосарію в деяких випадках.
- Удосконалення у документації.
- Поліпшення у швидкодії.
- Удосконалено сумісність додатка squash із Gerrit.
- Виправлено додавання рядків до одномовних складників глосарія.
- Удосконалено швидкодію обробки варіантів.
- Усунено помилку, яка призводила до того, що додаток squash іноді пропускав обробку змін у основній гілці.
- Реалізовано збереження суфікса назви файла при отриманні.
- Додано підтримку формату файлів Fluent.
- Додано підтримку використання табуляцій для форматування коду JSON.

Докладний опис усіх змін.

4.26 Weblate 4.7.2

Випущено 15 липня 2021 року.

- Реалізовано підтримку налаштування більшої кількості альтернативних назв мов на рівні проєкту.
- Виправлено перевірку чинності рядка пошуку у програмному інтерфейсі.
- Виправлено адреси засобів експортування Git після зміни домену.
- Виправлено додаток чищення для файлів RC Windows.
- Усунено можливе аварійне завершення роботи під час оновлення XLIFF.

[Докладний опис усіх змін.](#)

4.27 Weblate 4.7.1

Випущено 30 червня 2021 року.

- Удосконалено контекстну панель додавання термінів до глосарія.
- Додано підтримку служби комп'ютерного перекладу LibreTranslate.
- Додано обмеження на частоту при створенні проєктів.
- Поліпшено швидкодію оновлення файлів.

[Докладний опис усіх змін.](#)

4.28 Weblate 4.7

Випущено 17 червня 2021 року.

- Удосконалено перевірку коректності налаштувань.
- Додано підтримку `object-pascal-format`, що використовується у PO gettext, див. [Форматування Object Pascal](#).
- Параметр *Сусідні ключі* перейменовано на *Подібні ключі* з метою краще описати його призначення.
- Додано підтримку *міток mi18n*.
- Удосконалено інтеграцію розпізнавання SAML.
- Виправлення у інтеграції із *Gerrit* для кращої обробки нетипових випадків.
- Тепер Weblate вимагає Django 3.2.
- Виправлено запрошення користувачів, коли автентифікацію електронної пошти вимкнено.
- Удосконалено визначення мов.
- Додано підтримку блокування участі користувачів у проєкті.
- Виправлено автоматичне створення мов глосарія.
- Розширено документацію про додатки.
- Удосконалення швидкодії складників із пов'язаними репозиторіями.
- Додано підтримку безплатного програмного інтерфейсу DeepL.
- Керування користувачами більше не потребує інтерфейсу адміністратора Django.

[Докладний опис усіх змін.](#)

4.29 Weblate 4.6.2

Випущено 8 травня 2021 року.

- Усунено аварійне завершення роботи у відповідь на пересування складника між проєктами.
- Виправлено додавання нових рядків до порожніх файлів властивостей.
- Виправлено вирівнювання піктограми копіювання для мов із писемністю справа ліворуч.
- Розширено перелік статистичних даних на вкладці відомостей.
- Виправлено обробку назв файлів перекладу, які ігнорують у Git.
- Удосконалено швидкодію побудови метрики.
- Усунено можливу ваду у механізмі зберігання глосаріїв.
- Виправлено поведінку перевірки сумісності для мов із різними формами множини.

[Докладний опис усіх змін.](#)

4.30 Weblate 4.6.1

Випущено 2 травня 2021 року.

- Вилучено застарілий код захисту від спаму.
- Удосконалено точність перевірки форм множини.
- Оновлено список мов інтерфейсу у Docker.
- Удосконалено повідомлення щодо помилок при створенні запитів щодо об'єднання.
- Виправлено створення запитів щодо об'єднання у Pagine.
- Виправлено запуск автоматично встановлених додатків.
- Виправлено можливі проблеми із кешем при оновленні.
- Виправлено додавання нових модулів до одномовних перекладів за допомогою вивантаження даних.

[Докладний опис усіх змін.](#)

4.31 Weblate 4.6

Випущено 19 квітня 2021 року.

- У новій версії команда керування `auto_translate` має параметр для визначення режиму перекладу.
- Додано підтримку *Текстові файли*.
- Додано тенденції та метрику для всіх об'єктів.
- Додано підтримку для безпосереднього копіювання тексту із вторинних мов.
- Додано фільтрування за датою при навігації списком змін.
- Удосконалено діаграми діяльності.
- Уможливлено налаштування відправника для повідомлень форми зворотного зв'язку.
- Удосконалено перевірку параметрів у програмному інтерфейсі створення складників.
- Обмеження на частоту більше не стосується надкористувачів.
- Удосконалено швидкодію та надійність автоматизованого перекладу.

- Уможливлено налаштування обмеження за частотою у контейнері Docker.
- Реалізовано автоматичне використання *Внутрішні адреси Weblate* у програмному інтерфейсі для створення складників.
- Індикація спрощеного стану під час показу списку рядків.
- У новій версії для створення хешів паролів типово використовується Argon2.
- Спрощено смужки поступу, які показують стан перекладу.
- перейменовано *Додавання нових мов* — нова назва точніше вказує призначення.
- виправлено зберігання стану рядка до XLIFF.
- Додано пошук у межах мови.
- Початкова підтримка розгортання *горизонтального масштабування* для Docker.

Докладний опис усіх змін.

4.32 Weblate 4.5.3

Випущено 1 квітня 2021 року.

- виправлено збірку метрики.
- усунено можливе аварійне завершення роботи при додаванні рядків.
- удосконалено приклади запитів щодо пошуку.
- усунено можливу втрату щойно доданих рядків при вивантаженні із заміною.

4.33 Weblate 4.5.2

Випущено 26 березня 2021 року.

- Реалізовано придатний до налаштування розклад автоматичного перекладу.
- Додано перевірку форматування Lua.
- Реалізовано ігнорування рядків форматування у перевірці *Послідовне повторення слів*.
- Уможливлено вивантаження знімка екрана зі сторінки перекладу.
- Додано примусову синхронізацію файлів до механізмів супроводу репозиторію коду.
- виправлено автоматичні пропозиції для мов із довгим кодом.
- удосконалено швидкодію при додаванні нових рядків.
- виправлено декілька вад у перевірках якості.
- Декілька удосконалень у швидкодії.
- Додано інтеграцію із *Виявлення Weblate*.
- виправлено поведінку перевірок для придатних лише до читання рядків.

Докладний опис усіх змін.

4.34 Weblate 4.5.1

Випущено 5 березня 2021 року.

- Виправлено редагування прапорців глосарія в деяких рідкісних випадках.
- Розширене використання показників, щоб підвищити продуктивність кількох сторінок.
- Зберігайте правильну початкову мову в файлах TMX.
- Краща обробка завантаження одномовного РО за допомогою API.
- Удосконалено роботу застережень глосаріїв.
- Удосконалено перевірки посилань Markdown.
- Укажіть глосарій та мову джерела в навігаційних стежках.
- Перелік сторінок складників величезних проєктів.
- Удосконалено швидкодію обробки пам'яті перекладу, вилучень складників чи проєктів.
- Удосконалено швидкодію пакетного редагування.
- Виправлено збереження потреби редагування та затверджених станів файлів ODF.
- Удосконалено інтерфейс налаштувань завантаження файлів перекладу

[Докладний опис усіх змін.](#)

4.35 Weblate 4.5

Випущено 19 лютого 2021 року.

- Додано підтримку `lua-format`, що використовується у РО `gettext`.
- Додано підтримку спільного використання складників між проєктами.
- Виправлено перевірку на декілька змінних без назв при використанні декількох прапорців форматування.
- Поле списку листування виключено з налаштувань проєкту на користь загальних настанов для перекладачів.
- Додано додаток створення псевдолокалей.
- Додано підтримку файлів TermBase eXchange.
- Додано підтримку власноручного визначення варіантів рядків за допомогою позначки.
- Удосконалено швидкодію перевірки узгодженості.
- Удосконалено швидкодію пам'яті перекладу для довгих рядків.
- Додано підтримку пошуку у поясненнях.
- У новій версії уможливлено додавання і вилучення рядків у двомовних форматах.
- Розширено список підтримуваних мов комп'ютерного перекладу в Amazon Translate.
- Реалізовано автоматичне вмикання перевірок Java MessageFormat для властивостей Java.
- Додано новий метод вивантаження для додавання до перекладу нових рядків.
- Додано простий інтерфейс для навігації перекладом.
- Глосарії у новій версії зберігаються як звичайні складники.
- Вилучено спеціалізовані програмні інтерфейси для глосаріїв, оскільки для них тепер використовується програмний інтерфейс складників.

- Додано спрощений інтерфейс для вмикання і вимикання деяких з прапорців.
- Додано підтримку непридатних до перекладу або заборонених термінів у глосарії.
- Додано підтримку визначення термінології у глосарії.
- Перемикач напрямку запису мови пересунуто з метою отримати більше місце для віртуальної клавіатури.
- Додано пункт для автоматичного спостереження за проєктами, учасником яких є користувач.
- Додано перевірку того, чи переклад відповідає глосарію.
- Додана підтримка налаштування кольору тексту навігації.

[Докладний опис усіх змін.](#)

4.36 Weblate 4.4.2

Випущено 14 січня 2021 року.

- Виправлено помилки у одному із поширюваних файлів МО.

4.37 Weblate 4.4.1

Випущено 13 січня 2021 року.

- Виправлено скасування змін у формах множини.
- Виправлено вигляд довідки параметрів проєкту.
- Удосконалено керування користувачами.
- Удосконалено обробку контексту в одномовних файлах PO.
- Виправлено поведінку додатка очищення під час роботи з форматами HTML, ODF, IDML і Windows RC.
- Виправлено обробку місця у файлах CSV.
- Реалізовано використання стиснення даних для отримання файлів.
- Удосконалено інтерфейс імпортування даних з файлів ZIP.
- Удосконалено виявлення формату файлів для вивантажених файлів.
- Усунено дублювання запитів щодо злиття на Pagine.
- Пришвидшено показ перекладів-привидів.
- Переписано редактор перекладів з метою використання вбудованої текстової області браузера.
- Виправлено помилку, пов'язану із тим, що додаток чищення заважав додаванню нових рядків.
- Додано програмний інтерфейс для роботи з додатками.

[Докладний опис усіх змін.](#)

4.38 Weblate 4.4

Випущено 15 грудня 2020 року.

- Удосконалено перевірку під час створення складника.
- Тепер Weblate вимагає Django 3.1.
- Додано підтримку налаштування вигляду в інтерфейсі управління.
- Виправлено обробку стану лише для читання під час масового редагування.
- Поліпшена інтеграція з CodeMirror.
- Додано додаток для вилучення порожніх рядків із файлів перекладу.
- Для перекладів у новій версії використано редактор CodeMirror.
- Підсвічування синтаксичних конструкцій у редакторі перекладів для XML, HTML, Markdown і reStructuredText.
- Реалізовано підсвічування замінників у редакторі перекладів.
- Удосконалено підтримку нестандартних кодів мов.
- Додано попередження щодо використання неоднозначних кодів мов.
- Реалізовано показу користувачеві фільтрованого списку мов при додаванні нових перекладів.
- Розширено можливості пошуку для змін у журналі.
- Удосконалено сторінки подробиць щодо рахунків та процедуру розміщення вільних проєктів.
- API розширеної статистики перекладів.
- Поліпшено вкладку інших перекладів під час перекладу.
- Додано API завдань.
- Поліпшено швидкодію вивантаження файлів.
- Удосконалено вигляд визначених користувачем спеціальних символів.
- Удосконалено швидкодію автоматичного перекладу.
- Кілька вдосконалень інтерфейсу користувача.
- Удосконалено схему іменування отримуваних файлів ZIP.
- Додано пункт отримання сповіщень проєктів, за якими не ведеться спостереження.

[Докладний опис усіх змін.](#)

4.39 Weblate 4.3.2

Випущено 4 листопада 2020 року.

- Виправлено збої роботи під час використання певних масок файлів складників.
- Удосконалено точність перевірки на послідовне дублювання слів.
- Додано підтримку запитів щодо об'єднання Page.
- Удосконалено повідомлення про помилку при помилках реєстрації.
- Скасовано показ коментарів розробника у Markdown.
- Спрощено налаштування репозиторіїв Git з типовою гілкою, іншою ніж «master».
- Гілка main тепер є типовою для новостворених внутрішніх репозиторіїв.
- Знижено частоту помилкових спрацьовувань незмінених перекладів під час перекладу reStructuredText.

- Виправлено проблеми показу CodeMirror у деяких ситуаціях.
- Групу «Шаблони» перейменовано на «Джерела» для прояснення її призначення.
- Виправлено обробку запитів щодо об'єднання у GitLab для репозиторіїв із довгими шляхами.

[Докладний опис усіх змін.](#)

4.40 Weblate 4.3.1

Випущено 21 жовтня 2020 року.

- Удосконалено швидкодію автоматизованого перекладу.
- Виправлено застарівання сеансів для користувачів, які пройшли розпізнавання.
- Додано підтримку приховування даних щодо версії.
- Удосконалено сумісність сценаріїв спостереження із сервером Bitbucket.
- Удосконалено швидкодію обробки пам'яті перекладу.
- Зменшено споживання пам'яті.
- Поліпшено швидкодію матричних подань.
- Додано підтвердження перед вилученням користувача з проєкту.

[Докладний опис усіх змін.](#)

4.41 Weblate 4.3

Випущено 15 жовтня 2020 року.

- До програмного інтерфейсу включено статистичні дані користувача.
- Виправлено упорядкування складників на поділених на частини сторінках.
- Реалізовано визначення початкової мови для глосарія.
- Переписано підтримку запитів щодо об'єднання для GitHub і GitLab.
- Виправлено обчислення статистичних даних після вилучення пропозиції.
- Розширено дані загальнодоступного профілю користувача.
- Виправлено налаштування для примусових перевірок.
- Удосконалено документацію щодо вбудованих засобів резервного копіювання.
- Атрибут початкової мови пересунуто з проєкту до складника.
- Додано перевірку форматування I18n Vue.
- У перевірці типових заміників у новій версії реалізовано використання формальних виразів.
- Удосконалено вигляд режиму матриці.
- Пункт «Автоматизація» перейменовано на «Автоматичні пропозиції».
- Додано підтримку взаємодії із декількома екземплярами GitLab або GitHub.
- Розширено API — враховано оновлення і вилучення проєктів та модулів, а також глосарії.
- У API модулів належним чином реалізовано обробку рядків із формами множини.
- У новій версії засіб створення складників може обробляти файл ZIP або вивантажений документ.
- Консолідовано коди стану відповідей API .

- Реалізовано підтримку Markdown в угоді з перекладачами.
- Удосконалено відстеження початкових рядків.
- Удосконалено сумісність з форматами JSON, YAML і CSV.
- Додано підтримку вилучення рядків.
- Поліпшено швидкодію під час отримання файлів.
- Удосконалено панель керування репозиторіями.
- Реалізовано автоматичне вмикання формату java для Android.
- Додано підтримку локалізованих знімків вікон.
- Додано підтримку Python 3.9.
- Виправлено переклад файлів HTML за певних умов.

[Докладний опис усіх змін.](#)

4.42 Weblate 4.2.2

Випущено 2 вересня 2020 року.

- Виправлено встановлення відповідності початкових рядків для форматів JSON.
- Виправлено переспрямування для входу до системи для деяких варіантів розпізнавання.
- Виправлено розпізнавання за LDAP при синхронізації груп.
- Виправлено аварійне завершення роботи при звітуванні щодо поступу автоматичного перекладу.
- Виправлено об'єднання внесків Git, якщо увімкнено трейлери.
- Виправлено створення локальних складників системи керування версіями за допомогою програмного інтерфейсу.

4.43 Weblate 4.2.1

Випущено 21 серпня 2020 року.

- Виправлено збереження форм множини для деяких локалей у ресурсах Android.
- Виправлено аварійне завершення роботи у додатку очищення для деяких файлів XLIFF.
- Уможливлено налаштування CDN локалізації в образі Docker.

4.44 Weblate 4.2

Випущено 18 серпня 2020 року.

- Удосконалено сторінки користувачів і додано список користувачів.
- Відкинуто підтримку перенесення даних з випусків 3.x — перенесення слід виконувати із проміжними версіями 4.1 або 4.0.
- Додано експортування до декількох одномовних форматів.
- Удосконалено діаграми діяльності.
- Уможливлено налаштування кількості показаних сусідніх рядків.
- Додано підтримку блокування складників із помилками, пов'язаними з репозиторієм.

- Спрощено основну навігацію (кнопки замінено піктограмами).
- Удосконалено обробку кодів мов у інтеграції із Google Перекладачем.
- Реалізовано можливість створення трейлерів `Co-authored-by`: у додатку об'єднання внесків Git.
- Удосконалено обробник запитів щодо пошуку.
- Удосконалено повідомлення перевірок щодо форматування рядків.
- Підвищено швидкодію обробки пакетних змін у стані.
- Додано переспрямовування для сумісності після перейменування проєкту або складника.
- Додано сповіщення щодо підтвердження рядків, блокування складників та зміни умов ліцензування.
- Додано підтримку ModernMT.
- Уможливлено вимикання перезаписування затверджених перекладів при вивантаженні файла перекладу.
- Відкинуто підтримку деяких переспрямувань адрес для підтримки сумісності.
- Додано перевірку для літералів шаблонів ECMAScript.
- Додано параметр для спостереження за складником.
- Вилучено початкову крапку з ключів модулів JSON.
- Вилучено окрему чергу обробки Celery для пам'яті перекладу.
- Дозволити переклад усіх складників одразу.
- Дозволити налаштовувати заголовки HTTP Content-Security-Policy.
- Додано підтримку альтернативних назв мови на рівні проєкту.
- Новий додаток, який полегшує локалізацію коду HTML або JavaScript, див. [CDN локалізації JavaScript](#).
- Домен Weblate тепер налаштовується у параметрах, див. [SITE_DOMAIN](#).
- Додано підтримку пошуку за складником і проєктом.

4.45 Weblate 4.1.1

Випущено 19 червня 2020 року.

- виправлено зміну налаштувань автовиправлення або додатків у Docker.
- виправлено можливе аварійне завершення роботи на сторінці відомостей про Weblate.
- удосконалено встановлення скомпільованих файлів локалей.
- виправлено додавання слів до глосарія.
- виправлено клавіатурні скорочення для комп'ютерного перекладу.
- вилучено діагностичне виведення, яке спричиняло у певних випадках відкидання записів подій журналу.
- виправлено індикатор блокування у списку проєктів.
- виправлено побудову списку ключів GPG у деяких конфігураціях.
- додано параметр для визначення версії програмного інтерфейсу DeepL, яким слід скористатися.
- додано підтримку роботи у режимі надавача служб SAML, див. [Розпізнавання за SAML](#).

4.46 Weblate 4.1

Випущено 15 червня 2020 року.

- Додано підтримку створення нових перекладів, назви локалей яких містять код країни.
- Додано підтримку пошуку початкових рядків на основі знімка вікна.
- Розширено дані, які доступні для статистичних представлень.
- Удосконалено редагування критеріїв пошуку на сторінках «Переклад».
- Удосконалено обробку конкурентних оновлень вмісту репозиторію.
- До форми створення проєкту включено початкову мову.
- До подяк включено кількість внесених змін.
- Виправлено засіб вибору мови інтерфейсу у деяких випадках.
- Уможливлено додавання до «білого» списку методів реєстрації із закритими реєстраціями.
- Удосконалено пошук відповідних термінів у глосарії.
- Удосконалено встановлення відповідності у пам'яті перекладів.
- Реалізовано групування однакових результатів із комп'ютерного перекладу.
- На сторінку перекладу додано безпосереднє посилання на редагування знімка екрана.
- Удосконалено діалогове вікно підтвердження вилучення.
- До архіву ZIP з отриманими перекладами включено шаблони.
- Додано підтримку коду Markdown та налаштування сповіщень до оголошень проєкту.
- Розширено подробиці у списках із позначками.
- Додано підтримку нових форматів файлів: *Рядки PHP Laravel*, *Файли HTML*, *Формат OpenDocument*, *Формат IDML*, *Файли RC Windows*, *Переклади INI*, *Переклади INI Inno Setup*, *Властивості GWT*, *Файли JSON go-i18n*, *Файл ARB*.
- Реалізовано послідовне використання стану «пропущено» для пропущених перевірок.
- Додано підтримку налаштування типових додатків для вмикання.
- Виправлено клавіатурне скорочення редактора для пропускання перевірок.
- Удосконалено комп'ютерний переклад рядків із заміниками.
- Реалізовано показ проєкту перекладу для мов користувача з метою спрощення початкового етапу перекладу.
- Удосконалено обробку кодів мов.
- Реалізовано показ перекладів мовою користувача першими у списку.
- «Форми» перейменовано на загальніший термін — «варіанти».
- Додано нові перевірки якості: *Декілька змінних без назв*, *Тривалий час не перекладено*, *Послідовне повторення слів*.
- Повторно впроваджено підтримку витирання пам'яті перекладів.
- Виправлено параметр для ігнорування перевірок початкових рядків.
- Додано підтримку налаштування іншої гілки для запису змін.
- У новій версії програмний інтерфейс повідомляє про стан щодо обмеження на частоту виклику у заголовках HTTP.
- Додано підтримку програмного інтерфейсу Google Translate V3 (розширеного).
- Додано можливість обмеження доступу на рівні складників.

- Додано підтримку пробілів та інших спеціальних символів у прапорцях перекладу, див. [Налаштування поведінки за допомогою прапорців](#).
- Реалізовано безумовний показ перевірки на обробку тексту, якщо її увімкнено.
- У програмному інтерфейсі передбачено фільтрування змін.
- Додано підтримку спільного використання глосаріїв між проектами.

4.47 Weblate 4.0.4

Випущено 7 травня 2020 року.

- Виправлено виконання комплекту текстів у деяких середовищах Python 3.8.
- Виправлено друкарські помилки у документації.
- Виправлено створення складників за допомогою програмного інтерфейсу у деяких випадках.
- Виправлено помилки JavaScript, які заважали навігації на мобільних пристроях.
- Виправлено аварійне завершення роботи при показі результатів деяких перевірок.
- Виправлено побудову списку знімків вікон.
- Виправлено щомісячні сповіщення-резюме.
- Виправлено обробку проміжних перекладів із модулями, яких не існує у перекладі.

4.48 Weblate 4.0.3

Випущено 2 травня 2020 року.

- Виправлено можливе аварійне завершення при побудові звітів.
- Усунено врахування реєстру при обробці згадок користувачів у коментарях.
- Виправлено перенесення даних PostgreSQL для користувачів, які не є надкористувачами.
- Виправлено зміну адреси репозиторію при створенні складника.
- Виправлено аварійне завершення роботи, якщо зникає основний репозиторій коду.

4.49 Weblate 4.0.2

Випущено 27 квітня 2020 року.

- Удосконалено швидкодію обробки статистики перекладу.
- Удосконалено швидкодію при зміні міток.
- Удосконалено швидкодію пакетного редагування.
- Удосконалено швидкодію пам'яті перекладів.
- Виправлено ймовірне аварійне завершення роботи при вилучення складників.
- Виправлено показ змін у перекладі у деяких рідкісних випадках.
- Удосконалено попередження щодо надто довгої черги celery.
- Виправлено можливі помилкові підсумки під час перевірки узгодженості.
- Виправлено зависання під час зміни пов'язаного зі складником репозиторію.
- Для списків CSV і звітів включено дані щодо відстані редагування у змінах.

- Усунено помилкове сповіщення для перевірок пробілів навколо символів пунктуації для канадської французької.
- Виправлено експортування даних XLIFF із заміниками.
- Виправлено помилкове спрацьовування у перевірці пробілів нульової ширини.
- Удосконалено звітування щодо помилок у налаштуваннях.
- Виправлено вивантаження двомовних початкових файлів.
- Реалізовано автоматичне виявлення підтримуваних мов для комп'ютерного перекладу DeepL.
- Виправлено показ смужки поступу у деяких рідкісних випадках.
- Виправлено спрацьовування деяких перевірок на неперекладених рядках.

4.50 Weblate 4.0.1

Випущено 16 квітня 2020 року.

- Виправлено встановлення пакунка з PyPI.

4.51 Weblate 4.0

Випущено 16 квітня 2020 року.

- Для роботи Weblate з цієї версії потрібен Python 3.6 або новіша версія.
- Додано огляд керування для сповіщень щодо складників.
- Додано сповіщення щодо складників для помилкових адрес навігатора репозиторієм.
- Удосконалено сторінки входу та реєстрації.
- Керування доступом до проєкту та налаштування робочого процесу інтегровано до параметрів проєкту.
- Додано перевірку та засіб підсвічування для інтерполяції та вкладення і18next.
- Додано перевірку та засіб підсвічування для заміників із символом відсотків.
- Реалізовано показ пропозицій, які не проходять перевірки.
- Реалізовано запис змін у початковому рядку до журналу.
- Оновлено програмний інтерфейс Microsoft Translator до версії 3.
- Повторно реалізовано модуль пам'яті перекладів.
- Додано підтримку для декількох фільтрів `is`: у [Пошук](#).
- Уможливлено запобігання [перевірці на повторення](#) за вбудованим «чорним» списком.
- Удосконалено видобування коментарів з одномовних файлів `po`.
- Повідомлення дошки перейменовано на «оголошення».
- Виправлено певні проблеми із поштовими повідомленнями щодо реєстрації.
- Удосконалено додаток оновлення LINGUAS з метою розширення спектра обробки синтаксичних варіантів.
- Виправлено редагування одномовних початкових файлів XLIFF.
- Додано підтримку точної відповідності у [Пошук](#).
- Розширено програмний інтерфейс для роботи із знімками вікон, записами користувачів, груп, списками складників та розширено засоби створення проєктів.

- Додано підтримку вивантаження початкових даних для двомовних перекладів.
- Додано підтримку проміжної мови для розробників.
- Додано підтримку рецензування початкових рядків.
- Розширено параметри отримання для платформи із широкою пам'яттю перекладів.

4.52 Випуски Weblate 3.x

4.52.1 Weblate 3.11.3

Випущено 11 березня 2020 року.

- Виправлено пошук за полями у певній пріоритетності.
- Виправлено попередньо визначений запит щодо нещодавно доданих рядків.
- Виправлено повернення під час пошуку дублікатів.
- Виправлено показ сповіщень у Gmail.
- Виправлення показ змін із скасуванням у журналі.
- Додано посилання на записи подій у сповіщеннях-резюме.
- Виправлено адресу електронної пошти для підтвердження вилучення облікового запису.
- Додано підтримку розпізнавання Slack у контейнері Docker.
- Усунено надсилання сповіщень для мов без підписки.
- У огляд швидкодії включено черги Celery.
- Виправлено посилання на додатки у документації.
- Зменшено частину помилкових результатів у перевірках незмінених перекладів.
- Підвищено залежність bleach з метою усунення наслідків CVE-2020-6802.
- Виправлено побудову списку змін на рівні проєкту у журналі.
- Виправлено скасування результатів обчислення статистики у деяких рідкісних випадках.
- Виправлено пошук для певних станів рядків.
- Удосконалено поведінку перевірок форматування рядків на пропущений символ відсотків.
- Виправлено розпізнавання на основі даних сторонніх надавачів послуг.

4.52.2 Weblate 3.11.2

Випущено 22 лютого 2020 року.

- Виправлено показ пропозицій.
- Виправлено помилкове сповіщення щодо відсутності слів у деяких рядках.

4.52.3 Weblate 3.11.1

Випущено 20 лютого 2020 року.

- Документовано зміни у конфігурації Celery.
- Удосконалено перевірку назв файлів при створенні складників.
- Виправлено мінімальні версії деяких залежностей.
- Виправлено додавання груп і певними версіями Django.
- Виправлено запис вручну до основного репозиторію коду.
- Удосконалено встановлення відповідності записам глосарія.

4.52.4 Weblate 3.11

Випущено 17 лютого 2012 року.

- Уможливлено використання адреси запису системи керування версіями під час створення складників за допомогою програмного інтерфейсу.
- Реалізовано показ зображення-результату у перевірці ширини результату перекладу.
- Виправлено посилання у сповіщеннях електронною поштою.
- Удосконалено вигляд простих текстових повідомлень електронної пошти.
- Реалізовано показ проігнорованих перевірок і уможливлено їхню повторну активацію.
- Реалізовано показ сусідніх ключів для одномовних перекладів.
- Додано підтримку групування форм рядків.
- Рекомендовано оновлення до новіших версій Weblate у перевірках системи.
- Реалізовано докладніший аналіз для попереджень щодо дублювання мов.
- На сторінках проєкту реалізовано включення докладних відомостей щодо ліцензування.
- Реалізовано автоматичне поглиблення локальних копій, якщо це потрібно.
- Виправлено отримання рядків, які потребують дій.
- Нове попередження щодо використання двічі тієї самої маски файлів.
- Удосконалено видобування позиційних аргументів XML.
- Реалізовано примусове переспрямовування на вибраний проєкт за допомогою `SINGLE_PROJECT`.
- Додано параметр для розв'язання проблем із коментарями.
- Додано пакетне редагування прапорців.
- Додано підтримку міток.
- Додано додаток пакетного редагування.
- Додано параметр для *Примусові перевірки*.
- Підвищено типовий рівень чинності для посилань щодо підтвердження.
- Удосконалено інтеграцію із Matomo.
- Виправлено *перевірку перекладеності* так, щоб у ній належним чином оброблялися зміни у початковому рядку.
- Розширено налаштування автоматичного оновлення за допомогою змінної `AUTO_UPDATE`.
- Реалізовано повну синхронізацію перекладів у Weblate для додатків LINGUAS.

4.52.5 Weblate 3.10.3

Випущено 18 січня 2020 року.

- Реалізовано підтримку translate-toolkit 2.5.0.

4.52.6 Weblate 3.10.2

Випущено 18 січня 2020 року.

- Додано позначку блокування для проєктів.
- Виправлено ваду CSS, яка спричиняла блимання у деяких браузерах.
- Виправлено пошук у системах із локалями, відмінними від англійської.
- Удосконалено встановлення відповідності у репозиторіях для спостережних скриптів GitHub і Bitbucket.
- Виправлено перенесення даних для деяких систем зі встановленим Python 2.7.
- Уможливлено налаштування поверхневого клонування Git.
- Удосконалено обробку фонових сповіщень.
- Виправлено помилкове подання форми при навігації назад у браузері.
- Новий додаток для налаштування форматування YAML.
- Виправлено перевірку однаковості форм множини: у новій версії вона не ініціюється для мов із однією формою множини.
- Виправлено пошук за формальними виразами у деяких полях.

4.52.7 Weblate 3.10.1

Випущено 9 січня 2020 року.

- Програмний інтерфейс розширено для створення перекладів.
- Виправлено обробку деяких рідкісних випадків при перенесенні даних.
- Забезпечено сумісність із Django 3.0.
- Поліпшено швидкість очищення даних.
- Додано підтримку нетипового security.txt.
- Удосконалено структуру журналу змін.
- Удосконалено список перекладів на панелі приладів.
- Удосконалено відповіді HTTP для спостережних скриптів.
- Додано підтримку запитів щодо об'єднання GitLab у контейнері Docker.

4.52.8 Weblate 3.10

Випущено 20 грудня 2019 року.

- Удосконалено інтерфейс користувача застосунку.
- Додано перевірку на подвійні пробіли.
- Виправлено створення нових мов.
- Усунено можливість надсилання сповіщень щодо журналу рецензування на вилучені адреси електронної пошти.
- Додано підтримку придатних лише для читання рядків.
- Додано підтримку коду Markdown у коментарях.
- Уможливлено розташування тексту настанови щодо перекладу у відомостях щодо проєкту.
- Додано копіювання до буфера обміну для вторинних мов.
- Удосконалено підтримку для Mercurial.
- Удосконалено швидкодію отримання даних зі репозиторіїв Git.
- Додано підтримку пошуку за віком рядка.
- Реалізовано показ початкової мови для всіх перекладів.
- Реалізовано показ контексту для сусідніх рядків.
- Додано підтримку сповіщень щодо дій у репозиторії.
- Удосконалено списки перекладів.
- Розширено можливості з пошуку.
- Додано підтримку автоматично перекладних рядків, які позначено для редагування.
- Усунено надсилання дублікатів сповіщень для пов'язаних зі складниками попереджень.
- Удосконалено типове повідомлення запиту щодо об'єднання.
- Удосконалено показ стану рядка у дзен-режимі.
- Додано підтримку ширшого спектра мов у перекладачі Yandex.
- Удосконалено вигляд повідомлень електронної пошти зі сповіщеннями.
- Надано можливість вибору умов ліцензування перекладу.

4.52.9 Weblate 3.9.1

Випущено 28 жовтня 2019 року.

- Зі списків резервного копіювання вилучено декілька непотрібних файлів.
- Виправлено можливе аварійне завершення при побудові звітів.
- Виправлено помилку при перенесенні даних між базами даних.
- Додано підтримку примусового запису до репозиторіїв Git.
- Зменшено ризик втрати чинності ключем реєстрації.
- Виправлено помилку, пов'язану із тим, що засіб вилучення облікових записів перевищував обмеження на частоту звернення до програмного інтерфейсу.
- Додано пошук на основі пріоритетності.
- Усунено можливе аварійне завершення роботи при додаванні рядків до файла JSON.

- У новій версії враховано розміру початкового рядка при перевірці HTML на безпечність і виправленні рядків.
- Усунено надсилання сповіщень запрошеним або вилученим користувачам.
- Виправлено з'єднання SSL із redis у Celery у контейнері Docker.

4.52.10 Weblate 3.9

Випущено 15 жовтня 2019 року.

- До отримуваних файлів включено метадані Weblate.
- Удосконалено інтерфейс перегляду непройдених перевірок.
- У перевірках форматування позначено пропущені рядки.
- Реалізовано окрему перевірку для пробілів навколо пунктуації у перекладах французькою.
- Додано підтримку для виправлення деяких помилок, пов'язаних із якістю перекладів.
- Додано окремі права доступу для створення проєктів.
- Розширено статистичні дані для кількості символів.
- Удосконалено підтримку для кодів мов у стилі Java.
- Додано нову загальну перевірку символів-замінників.
- Додано підтримку замінників WebExtension JSON.
- Додано підтримку простого формату XML.
- Програмний інтерфейс розширено можливостями вилучення і створення проєктів, складників та перекладів.
- Додано підтримку скриптів стеження для Gitea і Gitee.
- Додано нову перевірку на основі нетипових формальних виразів.
- Уможливлено налаштування прав доступу до внесення записів до спільної пам'яті перекладів.
- Додано можливість отримання архівів ZIP для ширшого спектра файлів перекладів.
- XLIFF узгоджено зі стандартом шляхом обробки записів maxwidth і font.
- Додано нову перевірку і засіб виправлення для безпечної розмітки HTML для перекладу вебзастосунків.
- Додано попередження щодо складників при невідтримуваних налаштуваннях.
- Додано додаток автоматичного перекладу для побудови перекладів.
- Засоби автоматичного перекладу розширено на випадок додавання пропозицій.
- Реалізовано показ параметрів додатка в огляді.
- Підтримку Sentry реалізовано за допомогою сучасного Sentry SDK замість Raven.
- Змінено параметри прикладу так, щоб вони краще відповідали промисловому середовищу.
- Додано автоматичне створення резервних копій за допомогою BorgBackup.
- Відділено додаток очищення для RESX для запобігання небажаним оновленням файлів.
- Додано розширені можливості з пошуку.
- Уможливлено отримання власних звітів користувачами.
- Додано підручник з локалізації, який має допомогти налаштувати складники.
- Додано підтримку запитів щодо об'єднання GitLab.
- Удосконалено показ стану репозиторію.

- Реалізовано виконання автоматичного перекладу у фоновому режимі.

4.52.11 Weblate 3.8

Випущено 15 серпня 2019 року.

- Додано підтримку спрощеного створення подібних складників.
- Додано підтримку обробки прапорців перекладу із заснованих на XML форматах файлів.
- Реалізовано запис виключено до журналу Celery.
- Удосконалено швидкодію додатків, які діють на рівні репозиторію.
- Удосконалено вигляд повідомлень електронної пошти зі сповіщеннями.
- Виправлено поведінку при скиданні пароля.
- Удосконалено швидкодію обробки на більшості сторінок перекладу.
- Виправлено побудову списку для мов, які невідомі Weblate.
- Додано підтримку клонування додатків до виявлених складників.
- Додано підтримку заміни вмісту файлів при вивантаженні.
- Додано підтримку перекладу даних, які не зберігаються у системі керування версіями.
- Додано зображення віджета OpenGraph для використання у соціальних мережах.
- Додано підтримку анімованих знімків вікон.
- Удосконалено обробку одномовних файлів XLIFF.
- Усунено можливість надсилання декількох сповіщень для однієї події.
- Додано підтримку змін у фільтруванні.
- Розширено спектр попередньо визначених періодів для звітування.
- Додано підтримку скриптів стеження для репозиторіїв Azure.
- Нові включні сповіщення щодо пропозицій у черзі або неперекладених рядків.
- До повідомлень зі сповіщеннями додано посилання для скасування підписки у одне клацання.
- Виправлено помилки у перевірці «Було перекладено».
- Новий інтерфейс керування для адміністраторів.
- У новій версії можна встановити пріоритетність рядка за допомогою прапорців.
- Додано сторінки керування мовами.
- Додано перевірки для форматуваних рядків для бібліотеки Qt і Ruby.
- Додано налаштування для кращої відповідності встановленим екземплярам із одним проєктом.
- Реалізовано сповіщення щодо нового рядка при зміні рядків у одномовних перекладах.
- Додано окрему панель пам'яті перекладів із можливостями пошуку.

4.52.12 Weblate 3.7.1

Випущено 28 червня 2019 року.

- Оновлення документації.
- Виправлено декілька обмежень-вимог.
- Оновлено базу даних мов.
- Оновлення локалізації.
- Різноманітні удосконалення інтерфейсу користувача.
- Удосконалено обробку непідтримуваних, але придатних до виявлення файлів перекладу.
- Реалізовано докладніше звіт щодо пропущених вимог для форматів файлів.

4.52.13 Weblate 3.7

Випущено 21 червня 2019 року.

- Додано окрему чергу Celery для сповіщень.
- Використано однорідний вигляд із застосунком для сторінок навігації програмним інтерфейсом.
- Реалізовано включення статистики щодо затверджених перекладів до звітів.
- Реалізовано звітування щодо поступу при оновленні складника перекладу.
- Уможливлено переривання запущеного фонового оновлення складника.
- Мову шаблонів розширено для обробки назв файлів.
- Реалізовано використання шаблонів для посилань редактора та адреси навігатора репозиторієм коду.
- Реалізовано показ максимальної довжини і поточної кількості символів при редагуванні перекладу.
- Удосконалено обробку аббревіатур у перевірці незмінених перекладів.
- Оновлено початкову сторінку для нових учасників команди.
- Додано підтримку налаштування додатка msgmerge.
- Реалізовано затримку відкриття SMTP-з'єднання при надсиланні сповіщень.
- Удосконалено журналювання помилок.
- Уможливлено визначення нетипового місця для додатка створення МО.
- Додано додатки для очищення застарілих пропозицій або коментарів.
- Додано параметр для вмикання горизонтального режиму у джен-редактора.
- Удосконалено швидкодію імпортування даних із багатьма пов'язаними складниками.
- Виправлено приклади встановлення у деяких випадках.
- Удосконалено показ попереджень у змінах.
- Додано новий горизонтальний віджет статистики.
- Удосконалено перевірку форматування рядків у формах множини.
- Додано засіб керування шрифтами.
- Реалізовано нову перевірку розмірів обробленого тексту.
- Додано підтримку форматів субтитрів.
- Включено загальну статистику завершеності для мов.
- Додано звітування на рівні проєктів та загальному рівні.

- Удосконалено інтерфейс користувача при показі стану перекладу.
- Новий логотип Weblate та схема кольорів.
- Новий вигляд растрових значків.

4.52.14 Weblate 3.6.1

Випущено 26 квітня 2019 року.

- Удосконалено обробку одномовних файлів XLIFF.
- Виправлено сповіщення-резюме у деяких рідкісних випадках.
- Виправлено попередження щодо помилки скрипту додатка.
- Виправлено створення файла МО для одномовних файлів РО.
- Виправлено показ невстановлених перевірок.
- Реалізовано індикацію для адміністрованих проєктів у списку проєктів.
- Уможливлено оновлення для відновлення, якщо не виявлено репозиторію системи керування версіями.

4.52.15 Weblate 3.6

Випущено 20 квітня 2019 року.

- Додано підтримку отримання даних користувача.
- У новій версії додатки автоматично вмикаються після встановлення.
- Удосконалено настанови щодо вирішення конфліктів об'єднання.
- У новій версії додаток очищення сумісний із перекладами метаданих крамниці програм.
- Реалізовано придатний до налаштування синтаксис коду мови при додаванні нових перекладів.
- Реалізовано попередження щодо використання Python 2, планову підтримку якого буде завершено у квітні 2020 року.
- Реалізовано відобування спеціальних символів із початкового рядка для екранної клавіатури.
- Розширено статистичні дані щодо учасників з метою показу одразу даних щодо джерела та перекладу.
- У новій версії адміністратори та додатки забезпечення однорідності можуть додавати переклади, навіть якщо таку можливість вимкнено для користувачів.
- Виправлено опис перемикача, який вимикає обробку заголовка Language-Team.
- Реалізовано сповіщення користувачів, яких згадано у коментарях.
- Із налаштувань складників вилучено автоматичне визначення формату файлів.
- Виправлено створення файла МО для одномовних файлів РО.
- Додано сповіщення-резюме.
- Додано підтримку вимикання сповіщень щодо складників.
- Додано сповіщення щодо нових попереджень, повідомлень на дошці та складників.
- У новій версії можна налаштувати сповіщення для адміністративних проєктів.
- Удосконалено обробку трілітерних кодів мов.

4.52.16 Weblate 3.5.1

Випущено 10 березня 2019 року.

- Виправлено приклад модуля Celery для systemd.
- Виправлено сповіщення від репозиторіїв HTTP із реєстрацією.
- Виправлено конкуренцію при редагуванні початкового рядка для одномовних перекладів.
- До журналу включено дані, які виведено під час невдалих спроб виконання додатків.
- Удосконалено перевірку варіантів для додавання нової мови.
- Уможливлено редагування формату файлів у параметрах складників.
- Оновлено настанови щодо встановлення — зазначено пріоритетність Python 3.
- Удосконалено швидкодію та однорідність при завантаженні перекладів.
- Реалізовано сумісність служби Microsoft Terminology із поточними випусками Zeep.
- Оновлення локалізації.

4.52.17 Weblate 3.5

Випущено 3 березня 2019 року.

- Удосконалено швидкодію обробки вбудованої пам'яті перекладу.
- Додано інтерфейс для керування загальною пам'яттю перекладів.
- Удосконалено попередження щодо помилкового стану складників.
- Додано інтерфейс користувача для керування повідомленнями дошки.
- Уможливлено налаштування повідомлення про внесок від додатка.
- Зменшено кількість внесків при оновленні основного репозиторію коду.
- Виправлено можливу втрату метаданих при пересуванні складника між проектами.
- Удосконалено навігацію у дзен-режимі.
- Додано декілька нових перевірок якості (пов'язаних із Markdown та адресами).
- Додано підтримку файлів метаданих крамниць програм.
- Додано підтримку вмикання і вимикання інтеграції із GitHub або Gerrit.
- Додано перевірку для літер-кашид.
- Додано пункт для об'єднання внесків на основі імені автора.
- Удосконалено підтримку формату файлів XLSX.
- Реалізовано сумісність із Tesseract 4.0.
- Додаток рахунків у новій версії вилучає проекти із несплачених рахунків після 45 днів.

4.52.18 Weblate 3.4

Випущено 22 січня 2019 року.

- Додано підтримку для символів-замінників XLIFF.
- У новій версії Celery може використовувати декілька черг завдань.
- Додано підтримку перейменування та пересування проєктів та складників.
- До звітів включено дані щодо кількості символів.
- Додано кероване додавання складників перекладу із автоматичним виявленням файлів перекладу.
- Реалізовано придатні до налаштування повідомлення для внесків із об'єднанням у Git.
- Додано візуальну індикацію попереджень щодо складників при навігації.
- Удосконалено швидкодію завантаження файлів перекладу.
- Реалізовано новий додаток для об'єднання внесків перед запису до репозиторію.
- Удосконалено показ змін у перекладі.
- Змінено типовий стиль об'єднання для перенесення і уможливлено його налаштування.
- Удосконалено обробку підлеглих міток приватного використання у кодї мов.
- Удосконалено швидкодію повнотекстових оновлень покажчика.
- Розширено перелік параметрів програмного інтерфейсу вивантаження даних.

4.52.19 Weblate 3.3

Випущено 30 листопада 2018 року.

- Додано підтримку вилучення складників і проєктів.
- Удосконалено швидкодію обробки деяких одномовних перекладів.
- Додано попередження щодо складників перекладу для акцентування проблем перекладу.
- Реалізовано показ назви ресурсу рядка XLIFF як контексту, якщо такий доступний.
- Додано підтримку станів XLIFF.
- Додано перевірку щодо непридатних до запису файлів у DATA_DIR.
- Удосконалено експортування даних щодо змін до CSV.

4.52.20 Weblate 3.2.2

Випущено 20 жовтня 2018 року.

- Вилучено непотрібну вже залежність від Babel.
- Оновлено визначення мов.
- Удосконалено документацію щодо додатків, LDAP і Celery.
- Виправлено вмикання нових прапорців dos-eol і auto-java-messageformat.
- Виправлено запуск перевірки setup.py з пакунка PyPI.
- Удосконалено обробку форм множини.
- Виправлено помилку у програмному інтерфейсі вивантаження перекладів у рідкісних випадках.
- Виправлено оновлення налаштувань Git, якщо ці налаштування було змінено вручну.

4.52.21 Weblate 3.2.1

Випущено 10 жовтня 2018 року.

- Документовано залежність від backports.csv у Python 2.7.
- виправлено роботу комплексу тестування при запуску від імені root.
- Удосконалено обробку помилок у модулі gitexport.
- виправлено звітування щодо поступу для нових доданих мов.
- Реалізовано правильне звітування про помилки обробника Celery до Sentry.
- виправлено створення перекладів за допомогою Qt Linguist.
- Усунено можливі помилки оновлення повнотекстового покажчика.
- Удосконалено перевірки при створенні складників.
- Додано підтримку вилучення застарілих пропозицій.

4.52.22 Weblate 3.2

Випущено 6 жовтня 2018 року.

- Додано команду керування install_addon для автоматичного встановлення додатків.
- Уможливлено точніше визначення параметрів обмеження за частотою звернень у програмному інтерфейсі.
- Додано підтримку експортування та імпортування даних файлів Excel.
- Удосконалено очищення складників, якщо використано декілька додатків виявлення складників.
- Переписано модуль комп'ютерного перекладу за допомогою Microsoft Terminology.
- Нова версія Weblate використовує Celery для розвантаження засобів обробки.
- Удосконалено можливості з пошуку і додано пошук за формальними виразами.
- Додано підтримку програмного інтерфейсу комп'ютерного перекладу для Youdao Zhiyun.
- Додано підтримку програмного інтерфейсу комп'ютерного перекладу для Baidu.
- Завдання з супроводу та чищення об'єднано за допомогою Celery.
- Швидкодія завантаження перекладів збільшено на майже 25%.
- Вилучено підтримку об'єднання заголовків під час вивантаження.
- Вилучено підтримку нетипових повідомлень про внески.
- Реалізовано можливість налаштування режиму редагування (дзен-режим або повний режим).
- Додано підтримку звітування про помилки до Sentry.
- Додано підтримку автоматично щоденного оновлення репозиторіїв.
- Додано підтримку створення проєктів і складників користувачами.
- У новій версії пам'ять перекладів автоматично зберігає виконані переклади.
- Реалізовано можливість імпортування наявних записів пам'яті перекладів користувачами та керівниками проєктів.
- Удосконалено керування пов'язаними зі знімками вікон рядками.
- Додано підтримку перевірки MessageFormat Java.

Див. [етап 3.2 на GitHub](#), щоб дізнатися більше про усунені вади.

4.52.23 Weblate 3.1.1

Випущено 27 липня 2018 року.

- Виправлено помилки комплексу тестування у деяких конфігураціях.

4.52.24 Weblate 3.1

Випущено 27 липня 2018 року.

- Підтримку з версій, старіших за 3.0.1, вилучено.
- Уможливлено перевизначення типових повідомлень про внесок з параметрів проекту.
- Удосконалено сумісність скриптів спостереження із власними встановленими середовищами.
- Додано підтримку Amazon Translate.
- Забезпечено сумісність із Django 2.1.
- Для діагностування проблем зі встановленням у новій версії використовуються перевірки системи Django.
- Вилучено підтримку служби libavatar, яку невдовзі буде закрито.
- Реалізовано новий додаток для позначення незмінених перекладів як таких, що потребують редагування.
- Додано підтримку переходу до певного місця у перекладі під час перекладу.
- У новій версії передбачено налаштування параметрів отриманих файлів перекладів.
- Удосконалено обчислення подібності рядків у системі відповідників з пам'яті перекладів.
- Додано підтримку підписування внесків Git за допомогою GnuPG.

4.52.25 Weblate 3.0.1

Випущено 10 червня 2018 року.

- Виправлено можливі проблеми із перенесенням даних у 2.20.
- Оновлення локалізації.
- Вилучено застарілі приклади скриптів спостереження.
- Удосконалено документацію щодо кешування.
- Виправлено показ документації з адміністрування.
- Удосконалено обробку довгих назв мов.

4.52.26 Weblate 3.0

Випущено 1 червня 2018 року.

- Переписано систему керування доступом.
- Декілька чищень коду, які призвели до пересування і перейменування модулів.
- Новий додаток для автоматичного виявлення складників.
- Команда керування `import_project` у новій версії дещо змінила параметри.
- Додано базову підтримку файлів RC Windows.
- Новий додаток для зберігання імен учасників перекладу у заголовках файлів PO.

- Вилучено скрипти стеження для окремих складників — користуйтеся замість них додатками.
- Додано підтримку для збирання угод із учасниками перекладу.
- Записи щодо змін керування доступом у новій версії записуються до журналу.
- Новий додаток, який забезпечує однакові переклади в усіх складниках проєкту.
- Реалізовано підтримку більшої кількості змінних у шаблонах повідомлень щодо внесків.
- Додано підтримку забезпечення додаткового текстового контексту.

4.53 Випуски Weblate 2.x

4.53.1 Weblate 2.20

Випущено 4 квітня 2018 року.

- Підвищення швидкості клонування репозиторіїв subversion.
- Змінено блокування репозиторію — використано сторонню бібліотеку.
- Додано підтримку для отримання лише рядків, які потребують дій.
- Додано підтримку пошуку одразу у декількох перекладах різними мовами.
- Новий додаток для налаштування перенесення рядків у gettext.
- Новий додаток для налаштування форматування JSON.
- Додано підтримку розпізнавання у програмному інтерфейсі за допомогою сумісного із RFC 6750 розпізнавання Bearer.
- Додано підтримку автоматичного перекладу з використанням служб комп'ютерного перекладу.
- Додано підтримку розмітки HTML у повідомленнях дошки.
- Додано підтримку пакетної зміни стану рядків.
- Для роботи нової версії потрібен Translate-toolkit версії принаймні 2.3.0, підтримку застарілих версій вилучено.
- Додано вбудовану пам'ять перекладів.
- На панель приладів та сторінки огляду списків складників додано огляд списку складників.
- Додано підтримку служби комп'ютерного перекладу DeepL.
- У новій версії результати комп'ютерного перекладу кешуються у Weblate.
- Додано підтримку переупорядкування надісланих змін.

4.53.2 Weblate 2.19.1

Випущено 20 лютого 2018 року.

- Виправлено помилку перенесення даних при оновленні з версії 2.18.
- Удосконалено перевірку програмного інтерфейсу вивантаження файлів.

4.53.3 Weblate 2.19

Випущено 15 лютого 2018 року.

- Виправлено імпортування із перетворенням форматів файлів.
- Реалізовано показу зручних для читання даних щодо браузера у журналі аудиту.
- Додано засіб експортування TMX для файлів.
- Різноманітні удосконалення швидкодії для завантаження файлів перекладу.
- Додано пункт для вимикання керування доступом у Weblate і використання керування доступом Django.
- Удосконалено швидкість пошуку у глосарії для великих рядків.
- Реалізовано сумісність із django_auth_ldap 1.3.0.
- У новій версії повідомлення про помилки зберігається і записуються до звіту послідовно.
- Реалізовано врахування прапорців ігнорування у засобі автоматичного виправлення пробілів.
- Удосконалено сумісність із деякими конфігураціями Subversion.
- Удосконалено службу вбудованого комп'ютерного перекладу.
- Додано підтримку служб SAP Translation Hub.
- Додано підтримку служби Microsoft Terminology.
- Вилучено підтримку реклами у листах сповіщень.
- Удосконалено звітування щодо поступу перекладу на рівні мови.
- Удосконалено підтримку різних формул форм множини.
- Додано підтримку репозиторіїв Subversion, у яких не використовується stdlayout.
- Додано додатки для налаштування процедур перекладу.

4.53.4 Weblate 2.18

Випущено 15 грудня 2017 року.

- Розширено статистичні дані щодо учасників перекладу.
- Удосконалено налаштування віртуальної клавіатури для спеціальних символів.
- Додано підтримку формату файлів DTD.
- Змінено клавіатурні скорочення так, щоб зменшити ймовірність конфлікту із клавіатурними скороченнями браузера або системи.
- Удосконалено підтримку прапорця approved у файлах XLIFF.
- Додано підтримку вимикання перенесення довгих рядків у файлах PO gettext.
- Додано кнопку для копіювання посилання для поточного перекладу.
- Відкинута підтримку Django 1.10 і додано підтримку Django 2.0.
- Вилучено блокування перекладів на час перекладу.
- Додано підтримку додавання нових рядків до одномовних перекладів.
- Додано підтримку процесів перекладу із спеціалізованими рецензентами.

4.53.5 Weblate 2.17.1

Випущено 13 жовтня 2017 року.

- Виправлено роботу комплексу тестування у певних ситуаціях.
- Оновлення локалей.

4.53.6 Weblate 2.17

Випущено 13 жовтня 2017 року.

- Реалізовано типове поверхнєве клонування Weblate репозиторіїв Git.
- Удосконалено швидкодію при оновленні великих файлів перекладу.
- Додано підтримку блокування певних адрес електронної пошти при реєстрації.
- У новій версії передбачено можливість вилучення власних коментарів користувачами.
- Додано крок попереднього перегляду для можливості пошуку із заміною.
- Реалізовано збереження параметрів пошуку і форм вивантаження на боці клієнта.
- Розширено можливості з пошуку.
- Реалізовано точніші налаштування ACL на рівні проєкту.
- Змінено типове значення BASE_DIR.
- Додано двокрокову процедуру вилучення облікового запису, щоб запобігти випадковому вилученню.
- Передбачено можливість редагування параметрів керування доступом до проєкту.
- Додано необов'язковий захист за допомогою Akismet від спаму пропозиціями.

4.53.7 Weblate 2.16

Випущено 11 серпня 2017 року.

- Різноманітні удосконалення швидкодії.
- Додано підтримку для вкладеного формату JSON.
- Додано підтримку формату WebExtension JSON.
- Виправлено розпізнавання у засобі експортування git.
- Удосконалено імпортування CSV у деяких випадках.
- Удосконалено вигляд віджета «Інші переклади».
- У новій версії перевірки на максимальну довжину примусово встановлюють довжину тексту у формі.
- Уможливлено налаштування віку commit_pending на рівні складника.
- Різноманітні чищення інтерфейсу користувача.
- Виправлено пошук перекладів у складниках, проєктах і місцях.

4.53.8 Weblate 2.15

Випущено 30 червня 2017 року.

- Реалізовано показ більшої кількості пов'язаних перекладів у списку інших перекладів.
- Додано пункт для перегляду перекладів поточного рядка іншими мовами.
- Типово використано 4 форми множини для литовської мови.
- Виправлено вивантаження одномовних файлів у іншому форматі.
- Удосконалено повідомлення про помилку при помилках розпізнавання.
- Реалізовано збереження стану сторінки при вилученні слова з глосарія.
- Додано безпосереднє посилання для редагування перекладу вторинної мови.
- Додано перевірку якості форматування рядків Perl.
- Додано підтримку для відкидання повторно використаних паролів.
- Розширено панель інструментів для редагування мов із записом справа ліворуч.

4.53.9 Weblate 2.14.1

Випущено 24 травня 2017 року.

- Виправлено можливу помилку при поділі результатів пошуку на сторінки.
- Виправлено перенесення даних з застарілих версій у деяких рідкісних випадках.
- Виправлено можливе CSRF при вмиканні і вимиканні спостереження за проєктом.
- Усунено потребу у розпізнаванні користувача при скиданні пароля.
- Виправлено можливість обходу CAPTCHA при забутому паролі.

4.53.10 Weblate 2.14

Випущено 17 травня 2017 року.

- Додано записи глосарія із використанням AJAX.
- У новій версії для виходу із системи використовується POST для уникнення CSRF.
- У новій версії для скидання ключа до програмного інтерфейсу використовується POST для уникнення CSRF.
- У новій версії Weblate типово встановлює Content-Security-Policy.
- Реалізовано перевірку чинності адреси локального редактора для уникнення циклічного XSS.
- Реалізовано типову перевірку якості пароля.
- Реалізовано сповіщення користувачів щодо важливих дій із їхнім обліковим записом, зокрема зміни пароля.
- У експортованих даних CSV реалізовано екранування потенційних формул.
- Різноманітні незначні удосконалення захисту.
- Реалізовано обмеження за частотою спроб пройти розпізнавання.
- Реалізовано зберігання вмісту пропозицій у журналі.
- Реалізовано зберігання важливих дій із обліковим записом у журналі аудиту.
- Реалізовано запит щодо пароля при вилученні облікового запису або додаванні нових прив'язок.
- Реалізовано показ часу, коли було внесено пропозицію.

- Реалізовано нову перевірку якості для кінцевих крапок із комою.
- Забезпечено можливість поширення пошукових посилань.
- До програмного інтерфейсу включено дані щодо початкового рядка і знімків вікон.
- Виправлено перезапис перекладів під час вивантаження даних за допомогою програмного інтерфейсу.

4.53.11 Weblate 2.13.1

Випущено 12 квітня 2017 року.

- Виправлено побудову списку керованих проєктів у профілі.
- Виправлено помилку перенесення у випадках, коли не вистачало деяких прав доступу.
- Виправлено додавання до списку поточного формату файла при отриманні перекладу.
- Реалізовано повернення помилки HTTP 404 при спробі доступу до проєкту, якщо у користувача немає відповідних прав доступу.

4.53.12 Weblate 2.13

Випущено 12 квітня 2017 року.

- Виправлено перевірки якості шаблонів перекладу.
- Додано вмикання перевірки якості при втраті перекладу.
- Додано пункт для перегляду пропозицій від користувача з черги.
- Додано пункт для автоматичної побудови списків складників.
- Уможливлено налаштування типової панелі приладів для незареєстрованих користувачів.
- Додано пункт для навігації 25 випадковими рядками для рецензування.
- У новій версії зміни рядків додаються до журналу.
- Реалізовано краще звітування щодо помилок при додаванні нового перекладу.
- Додано пошук за мовами у межах проєкту.
- Уможливлено обмеження за певними правами доступу групових ACL.
- ACL на рівні проєкту у новій версії реалізовано за допомогою групових ACL.
- Додано засоби точнішого керування правами доступу.
- Різноманітні незначні удосконалення інтерфейсу користувача.

4.53.13 Weblate 2.12

Випущено 3 березня 2017 року.

- Удосконалено адміністративний інтерфейс для груп.
- Додано підтримку програмного інтерфейсу перекладача Yandex.
- Збільшено швидкість загального пошуку за сайтом.
- Додано режими пошуку за проєктом і за складником.
- Додано режими пошуку із заміною за проєктом і за складником.
- Удосконалено обробку суперечливих перекладів.
- Додано підтримку відкриття початкових файлів у локальному редакторі.

- Додано підтримку налаштування віртуальної клавіатури зі спеціальними символами.
- Удосконалено керування знімками вікон — реалізовано підтримку оптичного розпізнавання тексту для встановлення відповідності із рядками початкового коду.
- До типового повідомлення щодо внеску включено дані щодо перекладу та адресу.
- Додано підтримку формату перекладів Joomla.
- Удосконалено надійність імпортування даних між форматами файлів.

4.53.14 Weblate 2.11

Випущено 31 січня 2017 року.

- До сторінки мови включено докладні дані щодо мови.
- Удосконалення модуля роботи з Mercurial.
- Додано пункт для визначення пріоритетності складників перекладу.
- Реалізовано однорідніше використання групового ACL, навіть із рідкісними правами доступу.
- До спостережних скриптів додано змінну WL_BRANCH.
- Удосконалено документацію для розробників.
- Удосконалено сумісність із різноманітними версіями Git у додатку експортування даних до Git.
- Включено статистичні дані за проєктами та складниками.
- Додано прив'язку до кодів мов для удосконалення підтримки програмного інтерфейсу Microsoft Translate.
- Очищення повнотекстової бази даних пересунуто до фонового завдання, щоб пришвидшити вилучення перекладів.
- Виправлено показ початкових форм множини для мов із однією формою множини.
- Удосконалено обробку помилок у import_project.
- Різноманітні удосконалення швидкодії.

4.53.15 Weblate 2.10.1

Випущено 20 січня 2017 року.

- Усунено витік даних щодо існування облікового запису у формі скидання пароля (CVE-2017-5537).

4.53.16 Weblate 2.10

Випущено 15 грудня 2016 року.

- Додано перевірку якості для перевірки того, чи по-різному перекладено форми множини.
- Виправлено спостережні скрипти GitHub для репозиторіїв із розпізнаванням користувачів.
- Додано додатковий модуль експортування Git.
- Реалізовано підтримку програмного інтерфейсу когнітивних служб перекладу Microsoft.
- Спрощено інтерфейс користувача для проєктів і складників.
- Додано засіб автоматичного виправлення для вилучення керівних символів.
- Додано огляд окремих мов у проєкті.
- Додано підтримку експортування даних у форматі CSV.

- Додано можливість отримання статистики у форматі файла CSV.
- Додано матричний перегляд для швидкого огляду всіх перекладів.
- Додано базовий програмний інтерфейс для змін і рядків.
- Додано підтримку сервера комп'ютерного перекладу Apertium APu.

4.53.17 Weblate 2.9

Випущено 4 листопада 2016 року.

- Розширено перелік параметрів команди керування createadmin.
- Розширено можливості import_json для обробки наявних складників.
- Додано підтримку файлів YAML.
- У новій версії власники проекту можуть налаштовувати складник перекладу і подробиці щодо проекту.
- Використано термін «Watched» замість «Subscribed» для проектів.
- Реалізовано спостереження за проектом безпосередньо зі сторінки проекту.
- Додано віджет стану для декількох мов.
- Реалізовано підсвічування вторинної мови, якщо не показано джерело.
- Реалізовано запис вилучення пропозицій до журналу.
- Удосконалено інтерфейс вибору мов у профілі.
- Виправлено показ повідомлень дошки для складника.
- Усунено втрату позначення вкладки налаштувань після збереження.
- Удосконалено показ коментаря до початкового рядка.
- Реалізовано автоматичне встановлення драйвера об'єднання PO Gettext для репозиторіїв Git.
- Додано можливість пошуку із заміною.
- Додано підтримку вивантаження візуального контексту (знімків вікон) для перекладів.

4.53.18 Weblate 2.8

Випущено 31 серпня 2016 року.

- Удосконалення у документації.
- Переклади.
- Оновлено пов'язані бібліотеки JavaScript.
- Додано команду керування list_translators.
- Вилучено підтримку Django 1.8.
- Виправлено сумісність із Django 1.10.
- Додано підтримку Subversion.
- Перевірку коректності XML відокремлено від перевірки відповідності тегів XML.
- Виправлено програмний інтерфейс — враховано параметри HIDE_REPO_CREDENTIALS.
- Реалізовано показ зміни джерела у дзен-режимі.
- Реалізовано працездатність Alt+PageUp/PageDown/Home/End у дзен-режимі.
- Додано панелі підказки із показом точного часу внесення змін.

- Додано пункт для вибору фільтрів і пошуку зі сторінки перекладу.
- Додано інтерфейс для вилучення перекладів.
- Удосконалено роботу засобу вставлення заміників.
- Виправлено проблеми із автоматичним блокуванням у дзен-режимі.

4.53.19 Weblate 2.7

Випущено 10 липня 2016 року.

- Вилучено вебінтерфейс Google для комп'ютерних перекладів.
- Удосконалено повідомлення про внесок при додаванні перекладу.
- Виправлено програмний інтерфейс перекладача Google для івриту.
- Реалізовано сумісність із Mercurial 3.8.
- Додано команду керування `import_json`.
- Виправлено упорядкування перекладів у списку.
- Реалізовано показ повного тексту пропозиції, а не лише різниці.
- Розширено програмний інтерфейс (докладний стан репозиторію, статистика, ...).
- Для комплексу тестування більше не потрібен доступ до мережі для тестування репозиторіїв.

4.53.20 Weblate 2.6

Випущено 28 квітня 2016 року.

- Виправлено перевірку чинності складників за допомогою фільтра мов.
- Удосконалено підтримку файлів XLIFF.
- Виправлено комп'ютерний переклад для джерел, відмінних від джерел англійською мовою.
- Додано програмний інтерфейс REST.
- Реалізовано сумісність із Django 1.10.
- Додано категорії повідомлень дошки.

4.53.21 Weblate 2.5

Випущено 10 березня 2016 року.

- Виправлено автоматичний переклад для власників проєктів.
- Удосконалено швидкодію створення внесків і запису внесків до репозиторію.
- Реалізовано нову команду для додавання пропозицій з командного рядка.
- Додано підтримку об'єднання коментарів при вивантаженні файлів.
- Додано підтримку деяких розширень GNU до формату `printf C`.
- Удосконалення у документації.
- Додано підтримку створення подяк перекладачам.
- Додано підтримку показу статистичних даних щодо учасників перекладу.
- Загальний пошук лише за одною мовою.
- Удосконалено перевірки якості перекладів вірменською мовою.

- Реалізовано підтримку започаткування складників перекладу без наявних перекладів.
- Реалізовано підтримку додавання нових перекладів у Qt TS.
- Удосконалено підтримку перекладу файлів PHP.
- Поліпшення у швидкодії перевірок якості.
- виправлено загальний пошук за сайтом для рядків, які не пройшли перевірок.
- Додано пункт для визначення початкової мови.
- Удосконалено підтримку файлів XLIFF.
- Розширено список параметрів `import_project`.
- Удосконалено визначення аудиторії призначення для повідомлень дошки.
- Реалізовано підтримку автоматичного перекладу однакових повідомлень у різних проєктах.
- Оптимізовано повнотекстовий показник для пошуку.
- Додано команду керування для автоматичного перекладу.
- Додано підтримку підсвічування символів-замінників.
- Додано клавіатурні скорочення для вставлення символів-замінників, перевірок та комп'ютерного перекладу.
- Удосконалено блокування перекладів.
- Додано перевірку якості для інтерполяції AngularJS.
- Додано розширені ACL на основі груп.
- Прояснено термінологію для рядків, які потребують редагування (раніше «неточні»).
- Прояснено термінологію для рядків, які потребують дій і неперекладених рядків.
- Реалізовано підтримку Python 3.
- Відкинуто підтримку Django 1.7.
- Відкинуто залежність від msginit для створення файлів PO gettext.
- Додано придатні до налаштування панелі приладів.
- Удосконалено сповіщення щодо помилок обробки.
- Додано параметр для імпортування складників із дублюванням назв для `import_project`.
- Удосконалено підтримку перекладу файлів PHP.
- Додано експортування XLIFF для словника.
- Додано експортування XLIFF і PO gettext для всіх перекладів.
- Удосконалення у документації.
- Додано підтримку для придатних до налаштування автоматичних призначень груп.
- Удосконалено додавання нових перекладів.

4.53.22 Weblate 2.4

Випущено 20 вересня 2015 року.

- Удосконалено підтримку файлів PHP.
- Реалізовано можливість додавання ACL для анонімного користувача.
- Удосконалено придатність до налаштування команди `import_project`.
- До журналу додано дамп CSV.
- Усунено помилки копіювання зі вставленням із пробільними символами.
- Додано підтримку скриптів спостереження Bitbucket.
- Ужорсточено контроль за неточними перекладами при вивантаженні перекладу.
- Було змінено декілька адрес. Вам варто оновити ваші закладки.
- Реалізовано запуск скриптів стеження з кореневого каталогу системи керування версіями.
- Реалізовано запуск скриптів стеження із змінними середовища, які описують поточний складник.
- Додано команду керування для оптимізації повнотекстового показчика.
- Додано підтримку звітування про помилки до Rollbar.
- Уможливлено визначення декількох власників проекту.
- Уможливлено керування власниками проекту з боку власників проекту.
- Додано підтримку `javascript-format`, що використовується у PO gettext.
- Реалізовано підтримку додавання нових перекладів у XLIFF.
- Удосконалено автоматичне виявлення формату файлів.
- Розширено спектр клавіатурних скорочень.
- Удосконалено засоби встановлення відповідності за словником для декількох мов.
- Удосконалено компонування більшості сторінок.
- Реалізовано підтримку додавання слів до словника під час перекладу.
- Додано підтримку керування фільтруванням мов з боку Weblate.
- Додано підтримку перекладу та імпортування файлів CSV.
- Переписано засоби обробки статичних файлів.
- Реалізовано безпосередні посилання на вхід або реєстрацію за допомогою сторонньої служби, якщо ця служба є єдиною.
- Реалізовано внесення змін з черги при вилученні облікового запису.
- Додано команду керування для зміни назви сайту.
- Додано параметр для налаштування типового автора внеску.
- Додано скрипт стеження після додавання нового перекладу.
- Додано параметр для визначення декількох файлів, які слід додати у внеску.

4.53.23 Weblate 2.3

Випущено 22 травня 2015 року.

- Відкинуто підтримку Django 1.6 та перенесення South.
- Реалізовано підтримку додавання нових перекладів при використанні файлів властивостей Java.
- Уможливлено прийняття пропозиції без редагування.
- Удосконалено підтримку Google OAuth 2.0.
- Додано підтримку файлів .resx Microsoft.
- Скориговано типовий файл robots.txt з метою запобігання великим витокам перекладів.
- Спрощено процедуру прийняття пропозицій.
- Додано власників проєкту, які завжди отримують важливі сповіщення.
- Дозволено вимикати редагування одномовного шаблону.
- Реалізовано докладнішу панель перегляду стану репозиторію.
- Реалізовано безпосередні посилання на редагування шаблону при зміні перекладу.
- Уможливлено додавання прав доступу для власників проєктів.
- Уможливлено показ вторинної мови у дзен-режимі.
- Реалізовано приховування початкового рядка і показ замість нього рядка вторинною мовою.

4.53.24 Weblate 2.2

Випущено 19 лютого 2015 року.

- Поліпшення у швидкодії.
- Реалізовано повнотекстовий пошук у полях розташування та коментарів.
- Реалізовано нові діаграми активності на основі SVG/JavaScript.
- Реалізовано підтримку Django 1.8.
- Реалізовано підтримку вилучення коментарів.
- Додано власну емблему SVG.
- Додано підтримку Google Analytics.
- Удосконалено обробку назв файлів перекладу.
- Додано підтримку одномовних перекладів JSON.
- Реалізовано запис блокування складників до журналу.
- Реалізовано підтримку редагування початкової мови (мови шаблону) для одномовних перекладів.
- Додано базову підтримку Gerrit.

4.53.25 Weblate 2.1

Випущено 5 грудня 2014 року.

- Додано підтримку репозиторіїв Mercurial.
- Замінено шрифт Glyphicon, автором якого є Awesome.
- Додано піктограми для служб розпізнавання за соціальними мережами.
- Удосконалено однорідність кольорів кнопок і піктограм.
- Удосконалення у документації.
- Різні виправлення помилок.
- Реалізовано автоматичне приховування стовпчиків у списку перекладів на малих екранах.
- Змінено налаштування шляхів у файловій системі.
- Удосконалено обробку із збереження ключів SSH.
- Удосконалено блокування репозиторіїв.
- Реалізовано придатні до налаштування перевірки якості для окремих початкових рядків.
- Уможливлено приховування виконаних перекладів з панелі приладів.

4.53.26 Weblate 2.0

Випущено 6 листопада 2014 року.

- Новий швидкий інтерфейс із використанням Bootstrap.
- Переписано модуль систем керування версіями.
- Удосконалення у документації.
- Додано дошку для загальних повідомлень.
- Уможливлено визначення пріоритетності рядків.
- Додано підтримку формату файлів JSON.
- Виправлено створення файлів то у деяких випадках.
- Додано підтримку сповіщень GitLab.
- Додано підтримку вимикання пропозицій до перекладу.
- Реалізовано підтримку Django 1.7.
- У проєктах ACL передбачено можливості керування записами користувачів.
- Розширено можливості пошуку.
- Реалізовано додаткові підказки щодо форм множини для перекладачів.
- Виправлено блокування репозиторіїв Git.
- Реалізовано сумісність із застарілими версіями Git.
- Удосконалено підтримку ACL.
- Додано кнопки для лапок та інших спеціальних символів в окремих мовах.
- Реалізовано підтримку експортування статистичних даних у форматі JSONP.

4.54 Випуски Weblate 1.x

4.54.1 Weblate 1.9

Випущено 6 травня 2014 року.

- Реалізовано сумісність із Django 1.6.
- Вилучено підтримку сумісності із Django 1.4.
- Реалізовано команди керування для блокування та розблокування перекладів.
- Удосконалено підтримку файлів TS Qt.
- Уможливлено вилучення облікового запису для самих користувачів.
- Уможливлено вимикання аватарів.
- Об'єднано атрибути імені і прізвища.
- Реалізовано отримання і кешування аватарів на боці сервера.
- Додано підтримку емблеми shields.io.

4.54.2 Weblate 1.8

Випущено 7 листопада 2013 року.

- Настанови щодо оновлення наведено у підручнику.
- Удосконалено список даних у резюме проекту.
- Удосконалено показ пунктів поширення даних.
- Удосконалено контроль за правами доступу анонімних користувачів.
- Реалізовано підтримку використання сторонніх служб для входу до системи. Ознайомтеся із підручником, щоб дізнатися більше.
- Уможливлено вхід користувачів за адресою електронної пошти замість імені користувача.
- Удосконалення у документації.
- Удосконалено рецензування початкових рядків.
- Реалізовано пошук в усіх рядках.
- Удосконалено прив'язку початкових рядків.
- Реалізовано захист CAPTCHA реєстрації.

4.54.3 Weblate 1.7

Випущено 7 жовтня 2013 року.

- Настанови щодо оновлення наведено у підручнику.
- Реалізовано підтримку перевірки рядків форматування із фігурними дужками у Python.
- Реалізовано налаштування перевірок якості для окремих складників.
- Реалізовано показ докладної статистики для окремих перекладів.
- Змінено спосіб прив'язки пропозицій, перевірок і коментарів до рядків.
- Уможливлено додавання користувачами тексту до повідомлення про внесок.
- Реалізовано підтримку підписки на запити щодо нових мов.

- Реалізовано підтримку додавання нових перекладів.
- Віджети і діаграми у новій версії обробляються з використанням Pillow замість Pango + Cairo.
- Додано віджет емблеми стану.
- Викинуто некоректну перевірку напрямку запису тексту.
- Реалізовано запис змін у словнику до журналу.
- Удосконалено швидкодію для панелі перекладу.

4.54.4 Weblate 1.6

Випущено 25 липня 2013 року.

- Удосконалено обробку помилок при реєстрації.
- Навігація змінами.
- виправлено упорядкування пропозицій комп'ютерного перекладу.
- Удосконалено підтримку комп'ютерного перекладу MyMemory.
- Додано підтримку комп'ютерного перекладу Amagama.
- Різноманітні оптимізації на часто використовуваних сторінках.
- Реалізовано підсвічування пошукового рядка у результатах пошуку.
- Реалізовано підтримку автоматичного виправлення під час збереження повідомлення.
- Реалізовано стеження за журналом перекладів і параметр для скасування змін.
- Додано підтримку програмного інтерфейсу Google Translate.
- Додано підтримку керування ключами SSH вузла.
- Різноманітні удосконалення перевірки форми.
- Різноманітні удосконалення перевірок якості.
- Удосконалення швидкодії для імпортування.
- Додано підтримку голосування за пропозиції.
- Очищення адміністративного інтерфейсу.

4.54.5 Weblate 1.5

Випущено 16 квітня 2013 року.

- Настанови щодо оновлення наведено у підручнику.
- Додано загальнодоступні сторінки користувачів.
- Удосконалено іменування форм множини.
- Додано підтримку експортування TBX до глосарія.
- Додано підтримку сповіщень Bitbucket.
- Реалізовано діаграми активності для кожного перекладу, мови і користувача.
- Розширено перелік параметрів адміністративної команди `import_project`.
- Реалізовано сумісність із Django 1.5.
- У новій версії аватари показуються із використанням libavatar.
- Додано можливість форматowanego виведення при експортуванні JSON.

- Різноманітні удосконалення швидкодії.
- Реалізовано індикацію помилок перевірок або неточно перекладених рядків на смужках поступу для проєктів або мов.
- Додано підтримку нетиповий спостережних скриптів, які спрацьовують до створення внеску, і внесення додаткових файлів.
- Переписано засоби пошук для удосконалення швидкодії і зручності.
- Новий інтерфейс для комп'ютерного перекладу.
- Додано підтримку одномовних файлів ро.
- Розширено обсяг кешованих метаданих для пришвидшення різноманітних режимів пошуку.
- Реалізовано показ кількості слів.

4.54.6 Weblate 1.4

Випущено 23 січня 2013 року.

- Виправлено вилучення перевірок і коментарів при вилученні рядків.
- Додано пункт для вимикання автоматичного поширення перекладів.
- Додано пункт для оформлення підписки на помилки об'єднання.
- Реалізовано належне імпортування для проєктів, які потребують нетипового завантажувача tkit.
- Додано карти місць для спрощення доступу для «повзунів».
- Реалізовано безпосередні посилання на рядок у повідомленнях-сповіщеннях та подачах.
- Різноманітні удосконалення адміністративного інтерфейсу.
- Реалізовано підказки щодо промислових конфігурацій у адміністративному інтерфейсі.
- Додано віджети для окремих мов та сторінку залучення.
- Удосконалено обробку блокування перекладів.
- Реалізовано показ фрагментів для віджетів у ширшому спектрі варіантів.
- Реалізовано індикацію помилок перевірок або неточно перекладених рядків на смужках поступу.
- Розширено параметри форматування повідомлення щодо внеску.
- Виправлено помилку при роботі зі службами комп'ютерного перекладу.
- Удосконалено обробку блокування при автоматичному перекладі.
- Реалізовано підтримку показу зміни для попереднього початкового рядка.
- Додано підтримку пошуку за частиною рядків.
- Різноманітні удосконалення перевірок якості.
- Реалізовано підтримку окремих ACL на рівні проєкту.
- Базове покриття коду тестами модулів.

4.54.7 Weblate 1.3

Випущено 16 листопада 2012 року.

- Реалізовано сумісність із модулем бази даних PostgreSQL.
- Реалізовано вилучення мов, які вилучено у репозиторії git основної гілки розробки.
- Удосконалено обробку перевірок якості.
- Додано нові перевірки (ВВ-код, розмітка і розриви рядків XML).
- Реалізовано необов'язкове перенесення коду замість об'єднання.
- Реалізовано можливість перенесення Weblate (наприклад, запуску його з каталогу /weblate).
- Реалізовано підтримку вибору типу файлів вручну, якщо не вдається визначити тип файлів автоматично.
- Удосконалено підтримку ресурсів Android.
- Реалізовано підтримку створення ключа SSH з вебінтерфейсу.
- Більше видимих засобів експортування даних.
- Нові кнопки для введення деяких спеціальних символів.
- Реалізовано підтримку експортування словника.
- Реалізовано підтримку блокування усього встановленого екземпляра Weblate.
- Реалізовано перевірки початкових рядків і підтримку рецензування початкових рядків.
- Реалізовано підтримку коментарів користувача для перекладів і початкових рядків.
- Удосконалено стеження за змінами у журналі.
- У новій версії за змінами можна стежити за допомогою RSS.
- Удосконалено підтримку мов із записом справа ліворуч.

4.54.8 Weblate 1.2

Випущено 14 серпня 2012 року.

- Використано South для перенесення бази даних. Будь ласка, ознайомтеся із настановами щодо оновлення, якщо ви оновлюєте систему.
- Виправлено незначні проблеми із пов'язаними репозиторіями git.
- Нова вступна сторінка для залучення користувачів до перекладу за допомогою Weblate.
- Додано віджети, якими можна скористатися для просування проєктів з перекладу.
- Додано пункт для скидання стану репозиторію до початкового (для привілейованих користувачів).
- Передбачено можливість блокування проєкту або складника для перекладів.
- Реалізовано можливість вимикання деяких перекладів.
- Реалізовано придатні до налаштування варіанти додавання нових перекладів.
- Уможливлено налаштування параметрів внесків git для окремих проєктів.
- Реалізовано простий захист від спаму.
- Удосконалено компонування головної сторінки.
- Реалізовано підтримку автоматично запису змін для кожного внеску.
- Реалізовано підтримку сповіщень перекладачів електронною поштою.
- Реалізовано показ у налаштуваннях лише використаних мов.

- Удосконалено обробку невідомих мов при імпортуванні проєкту.
- Реалізовано підтримку блокування перекладу перекладачем.
- Реалізовано необов'язкову підтримку заголовка `Language-Team` у файлі PO.
- Додано певні статистичні дані на сторінку відомостей щодо проєкту.
- Реалізовано підтримку (і обов'язкове використання) `django-registration 0.8`.
- Реалізовано кешування кількостей рядків, які не пройшли перевірки.
- Реалізовано перевірку вимог під час налаштування.
- Удосконалення у документації.

4.54.9 Weblate 1.1

Випущено 4 липня 2012 року.

- Удосконалено декілька перекладів.
- Удосконалено перевірку при створенні складника.
- Додано підтримку спільних репозиторіїв git для складників.
- Усунено обов'язкові внески при кожній спробі отримання даних з віддаленого репозиторію.
- Додано підтримку розвантажувального індексування.

4.54.10 Weblate 1.0

Випущено 10 травня 2012 року.

- Удосконалено перевірку при додаванні і збереженні складника.
- Реалізовано експериментальну підтримку файлів складників Android (потребує латок до `ttkit`).
- Реалізовано запуск оновлень від скриптів стеження у фоновому режимі.
- Удосконалено настанови щодо встановлення.
- Удосконалено навігацію у словнику.

4.55 Випуски Weblate 0.x

4.55.1 Weblate 0.9

Випущено 18 квітня 2012 року.

- Виправлено імпортування невідомих мов.
- Удосконалено побудову списку сусідніх повідомлень.
- Удосконалено декілька перевірок.
- Оновлення документації.
- Додано визначення для декількох інших мов.
- Різноманітні чищення коду.
- Удосконалення у документації.
- Змінено компонування файлів.
- Оновлено допоміжні скрипти для Django 1.4.

- Удосконалено навігацію під час перекладу.
- Удосконалено обробку перейменовування файлів PO.
- Удосконалено перевірку при створенні складника.
- Усі налаштування інтегровано до syncdb.
- Додано список нещодавніх змін на усі сторінки перекладу.
- Під час перевірки на неперекладені рядки реалізовано нехтування повідомлень, які складаються лише з рядків форматування.

4.55.2 Weblate 0.8

Випущено 3 квітня 2012 року.

- Власний повнотекстовий пошук замінено на Whoosh.
- Різноманітні виправлення і удосконалення у перевірках.
- Нова команда updatechecks.
- Багато оновлень перекладів.
- Додано словник для зберігання більшості часто використовуваних термінів.
- Додано /admin/report/ для огляду стану репозиторіїв.
- Усунено блокування завантаження сторінок службами комп'ютерного перекладу.
- Нова версія інтерфейсу керування містить корисні пункти для оновлення даних.
- Реалізовано запис до журналу змін, які внесено користувачами.
- Реалізовано можливість відкладення внеску до Git для зменшення кількості внесків від одного користувача.
- Реалізовано можливість навігації непройденими перевірками.
- Реалізовано автоматичний переклад на основі вже перекладених рядків.
- Нова сторінка відомостей щодо проєкту із даними щодо використаних версій.
- Реалізовано сумісність із Django 1.4.
- Реалізовано можливість записування змін до віддаленого репозиторію за допомогою вебінтерфейсу.
- Додано огляд перекладів, які виконано іншими користувачами.

4.55.3 Weblate 0.7

Випущено 16 лютого 2012 року.

- Реалізовано безпосередню підтримку сповіщень GitHub.
- Додано підтримку очищення від застарілих перевірок та перекладів.
- Реалізовано підтримку показу сусідніх рядків під час перекладу.
- Реалізовано підтримку показу подібних рядків під час перекладу.
- Удосконалено пошук рядка.

4.55.4 Weblate 0.6

Випущено 14 лютого 2012 року.

- Додано різноманітні перевірки перекладених повідомлень.
- Реалізовано придатне налаштування керування доступом.
- Удосконалено обробку перекладів із символами розриву рядка.
- Додано сортування таблиць на боці клієнта.
- Будь ласка, ознайомтеся із настановами щодо оновлення, якщо ви оновлюєте систему.

4.55.5 Weblate 0.5

Випущено 12 лютого 2012 року.

- **Реалізовано підтримку комп'ютерного перекладу за допомогою таких служб:**
 - Apertium
 - Microsoft Translator
 - MyMemory
- Декілька нових перекладів.
- Удосконалено об'єднання із змінами, які внесено у основному репозиторії.
- Удосконалено обробку конкурентного отримання даних з git і перекладу.
- Реалізовано поширення між складниками змін у неточних перекладах.
- Реалізовано поширення перекладів між складниками при вивантаженні файлів.
- Виправлено отримання файлів при використанні FastCGI (а також, ймовірно, інших варіантів).

4.55.6 Weblate 0.4

Випущено 8 лютого 2012 року.

- До документації додано підручник з користування.
- Виправлено скрипти стеження програмного інтерфейсу так, щоб використовувався захист CSRF.

4.55.7 Weblate 0.3

Випущено 8 лютого 2012 року.

- Удосконалено показ початкового рядка для перекладу форм множини.
- Нова документація у форматі Sphinx.
- Реалізовано показ перекладів вторинними мовами під час перекладу.
- Удосконалено сторінку помилок — надано список наявних проєктів.
- Нова статистика для окремих мов.

4.55.8 Weblate 0.2

Випущено 7 лютого 2012 року.

- Удосконалено перевірку коректності декількох форм.
- Реалізовано попередження користувачів щодо оновлення профілю.
- Реалізовано запам'ятовування адреси для входу.
- Іменування текстових областей при введенні форм множини.
- Автоматичне розширення області перекладу.

4.55.9 Weblate 0.1

Випущено 6 лютого 2012 року.

- Початковий випуск.

W

wlc, [163](#)
wlc.config, [164](#)
wlc.main, [164](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component)/(string:group)/, 138
ANY /, 109	GET /api/components/(string:project)/(string:component)/(string:group)/, 133
/api	GET /api/components/(string:project)/(string:component)/(string:group)/, 135
GET /api/, 112	GET /api/components/(string:project)/(string:component)/(string:group)/, 136
/api/addons	GET /api/components/(string:project)/(string:component)/(string:group)/, 134
GET /api/addons/, 150	GET /api/components/(string:project)/(string:component)/(string:group)/, 133
GET /api/addons/(int:id)/, 150	GET /api/components/(string:project)/(string:component)/(string:group)/, 137
PUT /api/addons/(int:id)/, 151	GET /api/components/(string:project)/(string:component)/(string:group)/, 136
DELETE /api/addons/(int:id)/, 151	POST /api/components/(string:project)/(string:component)/(string:group)/, 150
PATCH /api/addons/(int:id)/, 150	POST /api/components/(string:project)/(string:component)/(string:group)/, 138
/api/changes	POST /api/components/(string:project)/(string:component)/(string:group)/, 134
GET /api/changes/, 147	POST /api/components/(string:project)/(string:component)/(string:group)/, 135
GET /api/changes/(int:id)/, 147	POST /api/components/(string:project)/(string:component)/(string:group)/, 136
/api/component-lists	PUT /api/components/(string:project)/(string:component)/(string:group)/, 132
GET /api/component-lists/, 151	DELETE /api/components/(string:project)/(string:component)/(string:group)/, 132
GET /api/component-lists/(str:slug)/, 151	DELETE /api/components/(string:project)/(string:component)/(string:group)/, 138
POST /api/component-lists/(str:slug)/components/, 152	PATCH /api/components/(string:project)/(string:component)/(string:group)/, 131
PUT /api/component-lists/(str:slug)/, 151	/api/groups
DELETE /api/component-lists/(str:slug)/, 152	GET /api/groups/, 116
DELETE /api/component-lists/(str:slug)/components/(string:component)/(string:group)/, 152	GET /api/groups/(int:id)/, 116
PATCH /api/component-lists/(str:slug)/, 151	POST /api/groups/, 116
/api/components	POST /api/groups/(int:id)/changes/, 118
GET /api/components/, 128	POST /api/groups/(int:id)/components/, 118
GET /api/components/(string:project)/(string:component)/(string:group)/, 128	
GET /api/components/(string:project)/(string:component)/(string:group)/, 133	
GET /api/components/(string:project)/(string:component)/(string:group)/, 133	

POST /api/groups/(int:id)/languages/, 118
 POST /api/groups/(int:id)/projects/, 118
 POST /api/groups/(int:id)/roles/, 117
 PUT /api/groups/(int:id)/, 117
 DELETE /api/groups/(int:id)/, 117
 DELETE /api/groups/(int:id)/componentlist/, 119
 DELETE /api/groups/(int:id)/components/(int:component_id)/, 118
 DELETE /api/groups/(int:id)/languages/(string:language_code_id)/, 118
 DELETE /api/groups/(int:id)/projects/(int:project_id)/, 118
 PATCH /api/groups/(int:id)/, 117

/api/languages

GET /api/languages/, 120
 GET /api/languages/(string:language)/, 120
 GET /api/languages/(string:language)/statistics/, 121
 POST /api/languages/, 120
 PUT /api/languages/(string:language)/, 121
 DELETE /api/languages/(string:language)/, 121
 PATCH /api/languages/(string:language)/, 121

/api/memory

GET /api/memory/, 144
 DELETE /api/memory/(int:memory_object_id)/, 144

/api/metrics

GET /api/metrics/, 153

/api/projects

GET /api/projects/, 122
 GET /api/projects/(string:project)/, 122
 GET /api/projects/(string:project)/changes/, 123
 GET /api/projects/(string:project)/components/, 125
 GET /api/projects/(string:project)/languages/, 127
 GET /api/projects/(string:project)/repositories/, 124
 GET /api/projects/(string:project)/statistics/, 128
 POST /api/projects/, 122
 POST /api/projects/(string:project)/components/, 125
 POST /api/projects/(string:project)/repositories/, 124

/api/roles

GET /api/roles/, 119
 POST /api/roles/, 119
 DELETE /api/roles/(int:id)/, 120
 PATCH /api/roles/(int:id)/, 120

/api/screenshots

GET /api/screenshots/, 148
 GET /api/screenshots/(int:id)/, 148
 GET /api/screenshots/(int:id)/file/, 148
 POST /api/screenshots/, 149
 POST /api/screenshots/(int:id)/file/, 148
 POST /api/screenshots/(int:id)/units/, 148
 PUT /api/screenshots/(int:id)/, 149
 DELETE /api/screenshots/(int:id)/, 150
 DELETE /api/screenshots/(int:id)/units/(int:unit_id)/, 149
 PATCH /api/screenshots/(int:id)/, 149

/api/tasks

GET /api/tasks/, 152
 GET /api/tasks/(str:uuid)/, 152

/api/translations

GET /api/translations/, 138
 GET /api/translations/(string:project)/(string:component)/, 138
 GET /api/translations/(string:project)/(string:component)/changes/, 141
 GET /api/translations/(string:project)/(string:component)/components/, 142
 GET /api/translations/(string:project)/(string:component)/languages/, 143
 GET /api/translations/(string:project)/(string:component)/repositories/, 144
 GET /api/translations/(string:project)/(string:component)/statistics/, 144
 POST /api/translations/(string:project)/(string:component)/, 141
 POST /api/translations/(string:project)/(string:component)/components/, 142
 POST /api/translations/(string:project)/(string:component)/repositories/, 141

DELETE /api/translations/(string:project)/(string:component)/(string:language)/, 141
 POST /hooks/pagure/, 154

/api/units

GET /api/units/, 145
 GET /api/units/(int:id)/, 145
 PUT /api/units/(int:id)/, 146
 DELETE /api/units/(int:id)/, 146
 PATCH /api/units/(int:id)/, 146

/api/users

GET /api/users/, 112
 GET /api/users/(str:username)/, 113
 GET /api/users/(str:username)/notifications/, 115
 GET /api/users/(str:username)/notifications/(int:subscription_id)/, 115
 GET /api/users/(str:username)/statistics/, 114
 POST /api/users/, 112
 POST /api/users/(str:username)/groups/, 114
 POST /api/users/(str:username)/notifications/, 115
 PUT /api/users/(str:username)/, 113
 PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 115
 DELETE /api/users/(str:username)/, 114
 DELETE /api/users/(str:username)/groups/, 114
 DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 115
 PATCH /api/users/(str:username)/, 114
 PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 115

/exports

GET /exports/rss/, 157
 GET /exports/rss/(string:project)/, 157
 GET /exports/rss/(string:project)/(string:component)/, 157
 GET /exports/rss/(string:project)/(string:component)/(string:language)/, 157
 GET /exports/rss/language/(string:language)/, 157
 GET /exports/stats/(string:project)/(string:component)/, 156

/hooks

GET /hooks/update/(string:project)/, 153
 GET /hooks/update/(string:project)/(string:component)/, 153
 POST /hooks/azure/, 155
 POST /hooks/bitbucket/, 154
 POST /hooks/gitea/, 155
 POST /hooks/gitee/, 155
 POST /hooks/github/, 153

Symbols

- .XML resource file
 - file format, [92](#)
- add
 - auto_translate command line option, [398](#)
- addon
 - install_addon command line option, [405](#)
- age
 - commit_pending command line option, [399](#)
- author
 - add_suggestions command line option, [397](#)
- author-email
 - wlc command line option, [161](#)
- author-name
 - wlc command line option, [161](#)
- base-file-template
 - import_project command line option, [402](#)
- check
 - importusers command line option, [404](#)
- config
 - wlc command line option, [159](#)
- config-section
 - wlc command line option, [159](#)
- configuration
 - install_addon command line option, [405](#)
- convert
 - wlc command line option, [160](#)
- email
 - createadmin command line option, [400](#)
- file-format
 - import_project command line option, [403](#)
- force
 - loadpo command line option, [406](#)
- force-commit
 - pushgit command line option, [407](#)
- format
 - wlc command line option, [159](#)
- fuzzy
 - wlc command line option, [161](#)
- ignore
 - import_json command line option, [401](#)
- inconsistent
 - auto_translate command line option, [398](#)
- input
 - wlc command line option, [161](#)
- key
 - wlc command line option, [159](#)
- lang
 - loadpo command line option, [406](#)
- language-code
 - list_translators command line option, [405](#)
- language-map
 - import_memory command line option, [402](#)
- language-regex
 - import_project command line option, [403](#)
- license
 - import_project command line option, [403](#)
- license-url
 - import_project command line option, [403](#)
- main-component
 - import_json command line option, [401](#)
 - import_project command line option, [403](#)
- method
 - wlc command line option, [161](#)
- mode
 - auto_translate command line option, [398](#)
- mt
 - auto_translate command line option, [398](#)
- name
 - createadmin command line option, [400](#)
- name-template

- import_project command line option, 402
 - new-base-template
 - import_project command line option, 403
 - no-password
 - createadmin command line option, 400
 - no-privs-update
 - setupgroups command line option, 407
 - no-projects-update
 - setupgroups command line option, 407
 - no-update
 - setuplang command line option, 408
 - output
 - wlc command line option, 160
 - overwrite
 - auto_translate command line option, 398
 - wlc command line option, 161
 - password
 - createadmin command line option, 400
 - project
 - import_json command line option, 401
 - source
 - auto_translate command line option, 398
 - threshold
 - auto_translate command line option, 398
 - update
 - createadmin command line option, 400
 - import_json command line option, 401
 - install_addon command line option, 405
 - url
 - wlc command line option, 159
 - user
 - auto_translate command line option, 398
 - username
 - createadmin command line option, 400
 - vcs
 - import_project command line option, 403
- ## A
- add_suggestions
 - weblate admin command, 397
 - add_suggestions command line option
 - author, 397
 - ADMINS
 - setting, 216
 - AKISMET_API_KEY
 - setting, 353
 - ALLOWED_HOSTS
 - setting, 216
 - Android
 - file format, 86
 - ANONYMOUS_USER_NAME
 - setting, 354
 - API, 109, 157, 163
 - Apple strings
 - file format, 87
 - ARB
 - file format, 91
 - AUDITLOG_EXPIRY
 - setting, 354
 - AUTH_LOCK_ATTEMPTS
 - setting, 354
 - AUTH_TOKEN_VALID
 - setting, 355
 - auto_translate
 - weblate admin command, 398
 - auto_translate command line option
 - add, 398
 - inconsistent, 398
 - mode, 398
 - mt, 398
 - overwrite, 398
 - source, 398
 - threshold, 398
 - user, 398
 - AUTO_UPDATE
 - setting, 354
 - AUTOFIX_LIST
 - setting, 355
 - AVATAR_URL_PREFIX
 - setting, 355
- ## B
- BACKGROUND_TASKS
 - setting, 356
 - BaseAddon (*клас в weblate.addons.base*), 444
 - BASIC_LANGUAGES
 - setting, 356
 - bilingual
 - translation, 77
 - BITBUCKETSERVER_CREDENTIALS
 - setting, 364
 - BORG_EXTRA_ARGS
 - setting, 357
- ## C
- CACHE_DIR
 - setting, 357
 - can_install() (*weblate.addons.base.BaseAddon class method*), 444
 - CELERY_BACKUP_OPTIONS, 170, 188
 - CELERY_BEAT_OPTIONS, 170, 188
 - CELERY_MAIN_OPTIONS, 170, 188
 - CELERY_MEMORY_OPTIONS, 170, 188
 - CELERY_NOTIFY_OPTIONS, 170, 188
 - celery_queues
 - weblate admin command, 398
 - CELERY_TRANSLATE_OPTIONS, 170, 188
 - changes

- wlc command line option, 160
- CHECK_LIST
 - setting, 358
- checkgit
 - weblate admin command, 399
- cleanup
 - wlc command line option, 160
- cleanup_ssh_keys
 - weblate admin command, 400
- cleanuptrans
 - weblate admin command, 399
- Comma separated values
 - file format, 93
- Command (*клас в wlc.main*), 164
- COMMENT_CLEANUP_DAYS
 - setting, 358
- commit
 - wlc command line option, 160
- commit_pending
 - weblate admin command, 399
- commit_pending command line option
 - age, 399
- COMMIT_PENDING_HOURS
 - setting, 358
- commitgit
 - weblate admin command, 399
- configure() (*weblate.addons.base.BaseAddon ме-
мод*), 444
- CONTACT_FORM
 - setting, 358
- createadmin
 - weblate admin command, 400
- createadmin command line option
 - email, 400
 - name, 400
 - no-password, 400
 - password, 400
 - update, 400
 - username, 400
- CSP_CONNECT_SRC
 - setting, 357
- CSP_FONT_SRC
 - setting, 357
- CSP_IMG_SRC
 - setting, 357
- CSP_SCRIPT_SRC
 - setting, 357
- CSP_STYLE_SRC
 - setting, 357
- CSV
 - file format, 93

D

- daily() (*weblate.addons.base.BaseAddon мемод*), 444
- DATA_DIR
 - setting, 359
- DATABASE_BACKUP

- setting, 359
- DATABASES
 - setting, 217
- DEBUG
 - setting, 217
- DEFAULT_ACCESS_CONTROL
 - setting, 360
- DEFAULT_ADD_MESSAGE
 - setting, 360
- DEFAULT_ADDON_MESSAGE
 - setting, 360
- DEFAULT_ADDONS
 - setting, 361
- DEFAULT_AUTO_WATCH
 - setting, 360
- DEFAULT_COMMIT_MESSAGE
 - setting, 360
- DEFAULT_COMMITER_EMAIL
 - setting, 361
- DEFAULT_COMMITER_NAME
 - setting, 361
- DEFAULT_DELETE_MESSAGE
 - setting, 360
- DEFAULT_FROM_EMAIL
 - setting, 217
- DEFAULT_LANGUAGE
 - setting, 361
- DEFAULT_MERGE_MESSAGE
 - setting, 360
- DEFAULT_MERGE_STYLE
 - setting, 362
- DEFAULT_PAGE_LIMIT
 - setting, 370
- DEFAULT_PULL_MESSAGE
 - setting, 362
- DEFAULT_RESTRICTED_COMPONENT
 - setting, 360
- DEFAULT_SHARED_TM
 - setting, 362
- DEFAULT_TRANSLATION_PROPAGATION
 - setting, 362
- download
 - wlc command line option, 160
- DTD
 - file format, 95
- dump_memory
 - weblate admin command, 400
- dumpuserdata
 - weblate admin command, 401

E

- ENABLE_AVATARS
 - setting, 362
- ENABLE_HOOKS
 - setting, 362
- ENABLE_HTTPS
 - setting, 363
- ENABLE_SHARING

setting, 363
 EXTRA_HTML_HEAD
 setting, 363

F

file format
 .XML resource file, 92
 Android, 86
 Apple strings, 87
 ARB, 91
 Comma separated values, 93
 CSV, 93
 DTD, 95
 gettext, 80
 go-i18n, 90
 gotext, 91
 GWT properties, 84
 i18next, 89
 INI translations, 85
 Java properties, 84
 Joomla translations, 86
 JSON, 88
 mi18n lang, 84
 PHP strings, 88
 PO, 80
 Qt, 86
 RC, 96
 ResourceDictionary, 93
 RESX, 92
 Ruby YAML, 95
 Ruby YAML Ain't Markup Language, 95
 string resources, 86
 TS, 86
 WPF, 93
 XLIFF, 82
 XML, 96
 YAML, 94
 YAML Ain't Markup Language, 94

G

get() (*wlc.Weblate memod*), 163
 get_add_form() (*weblate.addons.base.BaseAddon*
class method), 444
 GET_HELP_URL
 setting, 363
 get_settings_form()
 (*weblate.addons.base.BaseAddon memod*),
 444
 gettext
 file format, 80
 GITEA_CREDENTIALS
 setting, 363
 GITHUB_CREDENTIALS
 setting, 364
 GITLAB_CREDENTIALS
 setting, 364
 go-i18n
 file format, 90

GOOGLE_ANALYTICS_ID
 setting, 365
 gotext
 file format, 91
 GWT properties
 file format, 84

H

HIDE_REPO_CREDENTIALS
 setting, 365
 HIDE_VERSION
 setting, 365

I

i18next
 file format, 89
 import_demo
 weblate admin command, 401
 import_json
 weblate admin command, 401
 import_json command line option
 --ignore, 401
 --main-component, 401
 --project, 401
 --update, 401
 import_memory
 weblate admin command, 402
 import_memory command line option
 --language-map, 402
 import_project
 weblate admin command, 402
 import_project command line option
 --base-file-template, 402
 --file-format, 403
 --language-regex, 403
 --license, 403
 --license-url, 403
 --main-component, 403
 --name-template, 402
 --new-base-template, 403
 --vcs, 403
 importuserdata
 weblate admin command, 404
 importusers
 weblate admin command, 404
 importusers command line option
 --check, 404
 INI translations
 file format, 85
 install_addon
 weblate admin command, 405
 install_addon command line option
 --addon, 405
 --configuration, 405
 --update, 405
 INTERLEDGER_PAYMENT_POINTERS
 setting, 365
 iOS

- translation, 87
- IP_BEHIND_REVERSE_PROXY
 - setting, 366
- IP_PROXY_HEADER
 - setting, 366
- IP_PROXY_OFFSET
 - setting, 366
- J**
- Java properties
 - file format, 84
- Joomla translations
 - file format, 86
- JSON
 - file format, 88
- L**
- LEGAL_TOS_DATE
 - setting, 367
- LEGAL_URL
 - setting, 367
- LICENSE_EXTRA
 - setting, 367
- LICENSE_FILTER
 - setting, 367
- LICENSE_REQUIRED
 - setting, 368
- LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
 - setting, 368
- list_languages
 - weblate admin command, 405
- list_translators
 - weblate admin command, 405
- list_translators command line option
 - language-code, 405
- list_versions
 - weblate admin command, 406
- list-components
 - wlc command line option, 159
- list-languages
 - wlc command line option, 159
- list-projects
 - wlc command line option, 159
- list-translations
 - wlc command line option, 159
- load() (*wlc.config.WeblateConfig memod*), 164
- loadpo
 - weblate admin command, 406
- loadpo command line option
 - force, 406
 - lang, 406
- LOCALIZE_CDN_PATH
 - setting, 368
- LOCALIZE_CDN_URL
 - setting, 368
- lock
 - wlc command line option, 160
- lock_translation

- weblate admin command, 406
- lock-status
 - wlc command line option, 160
- LOGIN_REQUIRED_URLS
 - setting, 368
- LOGIN_REQUIRED_URLS_EXCEPTIONS
 - setting, 369
- ls
 - wlc command line option, 159

M

- main() (*в модулі wlc.main*), 164
- MATOMO_SITE_ID
 - setting, 369
- MATOMO_URL
 - setting, 369
- mi18n lang
 - file format, 84
- monolingual
 - translation, 77
- move_language
 - weblate admin command, 406

N

- NEARBY_MESSAGES
 - setting, 370

P

- PAGURE_CREDENTIALS
 - setting, 370
- PHP strings
 - file format, 88
- PIWIK_SITE_ID
 - setting, 369
- PIWIK_URL
 - setting, 369
- PO
 - file format, 80
- post() (*wlc.Weblate memod*), 163
- post_add() (*weblate.addons.base.BaseAddon memod*), 444
- post_commit() (*weblate.addons.base.BaseAddon memod*), 444
- post_push() (*weblate.addons.base.BaseAddon memod*), 444
- post_update() (*weblate.addons.base.BaseAddon memod*), 444
- pre_commit() (*weblate.addons.base.BaseAddon memod*), 445
- pre_push() (*weblate.addons.base.BaseAddon memod*), 445
- pre_update() (*weblate.addons.base.BaseAddon memod*), 445
- PRIVACY_URL
 - setting, 370
- PRIVATE_COMMIT_EMAIL_OPT_IN
 - setting, 371
- PRIVATE_COMMIT_EMAIL_TEMPLATE

setting, 371
 PROJECT_BACKUP_KEEP_COUNT
 setting, 371
 PROJECT_BACKUP_KEEP_DAYS
 setting, 371
 PROJECT_NAME_RESTRICT_RE
 setting, 372
 PROJECT_WEB_RESTRICT_HOST
 setting, 372
 PROJECT_WEB_RESTRICT_NUMERIC
 setting, 372
 PROJECT_WEB_RESTRICT_RE
 setting, 372
 pull
 wlc command line option, 160
 push
 wlc command line option, 160
 pushgit
 weblate admin command, 407
 pushgit command line option
 --force-commit, 407
 Python, 163
 Python Enhancement Proposals
 PEP 484, 440, 455

Q

Qt
 file format, 86

R

RATELIMIT_ATTEMPTS
 setting, 372
 RATELIMIT_LOCKOUT
 setting, 373
 RATELIMIT_WINDOW
 setting, 373
 RC
 file format, 96
 REDIS_PASSWORD, 185
 register_command() (*в модулі wlc.main*), 164
 REGISTRATION_ALLOW_BACKENDS
 setting, 373
 REGISTRATION_CAPTCHA
 setting, 373
 REGISTRATION_EMAIL_MATCH
 setting, 374
 REGISTRATION_OPEN
 setting, 374
 REGISTRATION_REBIND
 setting, 374
 repo
 wlc command line option, 160
 REPOSITORY_ALERT_THRESHOLD
 setting, 374
 REQUIRE_LOGIN
 setting, 375
 reset
 wlc command line option, 160

ResourceDictionary
 file format, 93
 REST, 109
 RESX
 file format, 92
 RFC
 RFC 5646, 77
 Ruby YAML
 file format, 95
 Ruby YAML Ain't Markup Language
 file format, 95

S

save_state() (*weblate.addons.base.BaseAddon ме-
 mod*), 445
 SECRET_KEY
 setting, 217
 SENTRY_DSN
 setting, 375
 SERVER_EMAIL
 setting, 217
 SESSION_COOKIE_AGE_AUTHENTICATED
 setting, 375
 SESSION_ENGINE
 setting, 217
 setting
 ADMINS, 216
 AKISMET_API_KEY, 353
 ALLOWED_HOSTS, 216
 ANONYMOUS_USER_NAME, 354
 AUDITLOG_EXPIRY, 354
 AUTH_LOCK_ATTEMPTS, 354
 AUTH_TOKEN_VALID, 355
 AUTO_UPDATE, 354
 AUTOFIX_LIST, 355
 AVATAR_URL_PREFIX, 355
 BACKGROUND_TASKS, 356
 BASIC_LANGUAGES, 356
 BITBUCKETSERVER_CREDENTIALS, 364
 BORG_EXTRA_ARGS, 357
 CACHE_DIR, 357
 CHECK_LIST, 358
 COMMENT_CLEANUP_DAYS, 358
 COMMIT_PENDING_HOURS, 358
 CONTACT_FORM, 358
 CSP_CONNECT_SRC, 357
 CSP_FONT_SRC, 357
 CSP_IMG_SRC, 357
 CSP_SCRIPT_SRC, 357
 CSP_STYLE_SRC, 357
 DATA_DIR, 359
 DATABASE_BACKUP, 359
 DATABASES, 217
 DEBUG, 217
 DEFAULT_ACCESS_CONTROL, 360
 DEFAULT_ADD_MESSAGE, 360
 DEFAULT_ADDON_MESSAGE, 360
 DEFAULT_ADDONS, 361

- DEFAULT_AUTO_WATCH, 360
- DEFAULT_COMMIT_MESSAGE, 360
- DEFAULT_COMMITTER_EMAIL, 361
- DEFAULT_COMMITTER_NAME, 361
- DEFAULT_DELETE_MESSAGE, 360
- DEFAULT_FROM_EMAIL, 217
- DEFAULT_LANGUAGE, 361
- DEFAULT_MERGE_MESSAGE, 360
- DEFAULT_MERGE_STYLE, 362
- DEFAULT_PAGE_LIMIT, 370
- DEFAULT_PULL_MESSAGE, 362
- DEFAULT_RESTRICTED_COMPONENT, 360
- DEFAULT_SHARED_TM, 362
- DEFAULT_TRANSLATION_PROPAGATION, 362
- ENABLE_AVATARS, 362
- ENABLE_HOOKS, 362
- ENABLE_HTTPS, 363
- ENABLE_SHARING, 363
- EXTRA_HTML_HEAD, 363
- GET_HELP_URL, 363
- GITEA_CREDENTIALS, 363
- GITHUB_CREDENTIALS, 364
- GITLAB_CREDENTIALS, 364
- GOOGLE_ANALYTICS_ID, 365
- HIDE_REPO_CREDENTIALS, 365
- HIDE_VERSION, 365
- INTERLEDGER_PAYMENT_POINTERS, 365
- IP_BEHIND_REVERSE_PROXY, 366
- IP_PROXY_HEADER, 366
- IP_PROXY_OFFSET, 366
- LEGAL_TOS_DATE, 367
- LEGAL_URL, 367
- LICENSE_EXTRA, 367
- LICENSE_FILTER, 367
- LICENSE_REQUIRED, 368
- LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LANGUAGE, 368
- LOCALIZE_CDN_PATH, 368
- LOCALIZE_CDN_URL, 368
- LOGIN_REQUIRED_URLS, 368
- LOGIN_REQUIRED_URLS_EXCEPTIONS, 369
- MATOMO_SITE_ID, 369
- MATOMO_URL, 369
- NEARBY_MESSAGES, 370
- PAGURE_CREDENTIALS, 370
- PIWIK_SITE_ID, 369
- PIWIK_URL, 369
- PRIVACY_URL, 370
- PRIVATE_COMMIT_EMAIL_OPT_IN, 371
- PRIVATE_COMMIT_EMAIL_TEMPLATE, 371
- PROJECT_BACKUP_KEEP_COUNT, 371
- PROJECT_BACKUP_KEEP_DAYS, 371
- PROJECT_NAME_RESTRICT_RE, 372
- PROJECT_WEB_RESTRICT_HOST, 372
- PROJECT_WEB_RESTRICT_NUMERIC, 372
- PROJECT_WEB_RESTRICT_RE, 372
- RATELIMIT_ATTEMPTS, 372
- RATELIMIT_LOCKOUT, 373
- RATELIMIT_WINDOW, 373
- REGISTRATION_ALLOW_BACKENDS, 373
- REGISTRATION_CAPTCHA, 373
- REGISTRATION_EMAIL_MATCH, 374
- REGISTRATION_OPEN, 374
- REGISTRATION_REBIND, 374
- REPOSITORY_ALERT_THRESHOLD, 374
- REQUIRE_LOGIN, 375
- SECRET_KEY, 217
- SENTRY_DSN, 375
- SERVER_EMAIL, 217
- SESSION_COOKIE_AGE_AUTHENTICATED, 375
- SESSION_ENGINE, 217
- SIMPLIFY_LANGUAGES, 375
- SINGLE_PROJECT, 376
- SITE_DOMAIN, 375
- SITE_TITLE, 376
- SPECIAL_CHARS, 376
- SSH_EXTRA_ARGS, 376
- STATUS_URL, 377
- SUGGESTION_CLEANUP_DAYS, 377
- UPDATE_LANGUAGES, 377
- URL_PREFIX, 377
- VCS_API_DELAY, 378
- VCS_BACKENDS, 378
- VCS_CLONE_DEPTH, 378
- WEBLATE_ADDONS, 379
- WEBLATE_EXPORTERS, 380
- WEBLATE_FORMATS, 380
- WEBLATE_GPG_IDENTITY, 380
- WEBLATE_MACHINERY, 380
- WEBSITE_REQUIRED, 380
- setupgroups
 - weblate admin command, 407
 - setupgroups command line option
 - no-privs-update, 407
 - no-projects-update, 407
- setuplang
 - weblate admin command, 408
- setuplang command line option
 - no-update, 408
- show
 - wlc command line option, 159
- SIMPLIFY_LANGUAGES
 - setting, 375
- SINGLE_PROJECT
 - setting, 376
- SITE_DOMAIN
 - setting, 375
- SITE_TITLE
 - setting, 376
- SPECIAL_CHARS
 - setting, 376
- SSH_EXTRA_ARGS
 - setting, 376
- stats

wlc command line option, 160
 STATUS_URL
 setting, 377
 store_post_load()
 (*weblate.addons.base.BaseAddon* *memod*),
 445
 string resources
 file format, 86
 SUGGESTION_CLEANUP_DAYS
 setting, 377

T

translation
 bilingual, 77
 iOS, 87
 monolingual, 77
 TS
 file format, 86

U

unit_pre_create()
 (*weblate.addons.base.BaseAddon* *memod*),
 445
 unlock
 wlc command line option, 160
 unlock_translation
 weblate admin command, 407
 UPDATE_LANGUAGES
 setting, 377
 updatechecks
 weblate admin command, 408
 updategit
 weblate admin command, 408
 upload
 wlc command line option, 160
 URL_PREFIX
 setting, 377

V

VCS_API_DELAY
 setting, 378
 VCS_BACKENDS
 setting, 378
 VCS_CLONE_DEPTH
 setting, 378
 version
 wlc command line option, 159

W

WEB_WORKERS, 170, 188
 Weblate (*клас в wlc*), 163
 weblate admin command
 add_suggestions, 397
 auto_translate, 398
 celery_queues, 398
 checkgit, 399
 cleanup_ssh_keys, 400
 cleanuptrans, 399

commit_pending, 399
 commitgit, 399
 createadmin, 400
 dump_memory, 400
 dumpuserdata, 401
 import_demo, 401
 import_json, 401
 import_memory, 402
 import_project, 402
 importuserdata, 404
 importusers, 404
 install_addon, 405
 list_languages, 405
 list_translators, 405
 list_versions, 406
 loadpo, 406
 lock_translation, 406
 move_language, 406
 pushgit, 407
 setupgroups, 407
 setuplang, 408
 unlock_translation, 407
 updatechecks, 408
 updategit, 408
 WEBLATE_ADDONS
 setting, 379
 WEBLATE_ADMIN_EMAIL, 170, 172
 WEBLATE_ADMIN_NAME, 170, 172
 WEBLATE_ADMIN_PASSWORD, 166, 170--172
 WEBLATE_ADMIN_PASSWORD_FILE, 172
 WEBLATE_AKISMET_API_KEY, 415
 WEBLATE_ALLOWED_HOSTS, 217, 221, 376
 WEBLATE_API_RATELIMIT_ANON, 111, 112
 WEBLATE_API_RATELIMIT_USER, 111, 112
 WEBLATE_AUTH_LDAP_BIND_PASSWORD, 179
 WEBLATE_DEBUG, 242
 WEBLATE_EMAIL_HOST_PASSWORD, 186
 WEBLATE_EMAIL_PORT, 185, 186
 WEBLATE_EMAIL_USE_SSL, 185, 186
 WEBLATE_EMAIL_USE_TLS, 185, 186
 WEBLATE_ENABLE_HTTPS, 253
 WEBLATE_EXPORTERS
 setting, 380
 WEBLATE_FORMATS
 setting, 380
 WEBLATE_GITHUB_HOST, 242
 WEBLATE_GPG_IDENTITY
 setting, 380
 WEBLATE_LOCALIZE_CDN_PATH, 187
 WEBLATE_MACHINERY
 setting, 380
 WEBLATE_RATELIMIT_ATTEMPTS, 417
 WEBLATE_REQUIRE_LOGIN, 375
 WEBLATE_SECURE_PROXY_SSL_HEADER, 173
 WEBLATE_SILENCED_SYSTEM_CHECKS, 249
 WEBLATE_SITE_DOMAIN, 219, 238
 WEBLATE_WORKERS, 170, 188, 189
 WeblateConfig (*клас в wlc.config*), 164

WeblateException, 163

WEBSITE_REQUIRED

setting, 380

wlc, 157

модуль, 163

wlc command line option

--author-email, 161

--author-name, 161

--config, 159

--config-section, 159

--convert, 160

--format, 159

--fuzzy, 161

--input, 161

--key, 159

--method, 161

--output, 160

--overwrite, 161

--url, 159

changes, 160

cleanup, 160

commit, 160

download, 160

list-components, 159

list-languages, 159

list-projects, 159

list-translations, 159

lock, 160

lock-status, 160

ls, 159

pull, 160

push, 160

repo, 160

reset, 160

show, 159

stats, 160

unlock, 160

upload, 160

version, 159

wlc.config

модуль, 164

wlc.main

модуль, 164

WPF

file format, 93

X

XLIFF

file format, 82

XML

file format, 96

Y

YAML

file format, 94

YAML Ain't Markup Language

file format, 94



змінна оточення

CELERY_BACKUP_OPTIONS, 170, 188

CELERY_BEAT_OPTIONS, 170, 188

CELERY_MAIN_OPTIONS, 170, 188

CELERY_MEMORY_OPTIONS, 170, 188

CELERY_NOTIFY_OPTIONS, 170, 188

CELERY_TRANSLATE_OPTIONS, 170, 188

CLIENT_MAX_BODY_SIZE, 178

POSTGRES_ALTER_ROLE, 183

POSTGRES_CONN_MAX_AGE, 183

POSTGRES_DATABASE, 183

POSTGRES_DISABLE_SERVER_SIDE_CURSORS,
184

POSTGRES_HOST, 183

POSTGRES_PASSWORD, 183

POSTGRES_PASSWORD_FILE, 183

POSTGRES_PORT, 183

POSTGRES_SSL_MODE, 183

POSTGRES_USER, 183

REDIS_DB, 184

REDIS_HOST, 184

REDIS_PASSWORD, 184, 185

REDIS_PASSWORD_FILE, 184

REDIS_PORT, 184

REDIS_TLS, 185

REDIS_VERIFY_SSL, 185

ROLLBAR_ENVIRONMENT, 187

ROLLBAR_KEY, 187

SENTRY_DSN, 187

SENTRY_ENVIRONMENT, 187

SOCIAL_AUTH_SLACK_SECRET, 182

WEB_WORKERS, 170, 188

WEBLATE_ADD_ADDONS, 188

WEBLATE_ADD_APPS, 187

WEBLATE_ADD_AUTOFIX, 188

WEBLATE_ADD_CHECK, 187

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS,
174

WEBLATE_ADMIN_EMAIL, 170--172

WEBLATE_ADMIN_NAME, 170--172

WEBLATE_ADMIN_PASSWORD, 166, 170--172

WEBLATE_ADMIN_PASSWORD_FILE, 172

WEBLATE_AKISMET_API_KEY, 176, 415

WEBLATE_ALLOWED_HOSTS, 172, 217, 221,
376

WEBLATE_API_RATELIMIT_ANON, 111, 112,
177

WEBLATE_API_RATELIMIT_USER, 111, 112,
177

WEBLATE_AUTH_LDAP_BIND_DN, 179

WEBLATE_AUTH_LDAP_BIND_PASSWORD,
179

WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE,
179

WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS,
179

WEBLATE_AUTH_LDAP_SERVER_URI, 179

WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 179
 WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 179
 WEBLATE_AUTH_LDAP_USER_SEARCH, 179
 WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 179
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 179
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 179
 WEBLATE_AUTO_UPDATE, 186
 WEBLATE_AVATAR_URL_PREFIX, 177
 WEBLATE_BASIC_LANGUAGES, 177
 WEBLATE_BITBUCKETSERVER_HOST, 175
 WEBLATE_BITBUCKETSERVER_TOKEN, 175
 WEBLATE_BITBUCKETSERVER_USERNAME, 175
 WEBLATE_BORG_EXTRA_ARGS, 177
 WEBLATE_CONTACT_FORM, 172
 WEBLATE_CORS_ALLOWED_ORIGINS, 178
 WEBLATE_CSP_CONNECT_SRC, 176
 WEBLATE_CSP_FONT_SRC, 176
 WEBLATE_CSP_IMG_SRC, 176
 WEBLATE_CSP_SCRIPT_SRC, 176
 WEBLATE_CSP_STYLE_SRC, 176
 WEBLATE_DATABASE_BACKUP, 184
 WEBLATE_DEBUG, 171, 242
 WEBLATE_DEFAULT_ACCESS_CONTROL, 176
 WEBLATE_DEFAULT_AUTO_WATCH, 177
 WEBLATE_DEFAULT_COMMITER_EMAIL, 176
 WEBLATE_DEFAULT_COMMITER_NAME, 176
 WEBLATE_DEFAULT_FROM_EMAIL, 172
 WEBLATE_DEFAULT_PULL_MESSAGE, 175
 WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 176
 WEBLATE_DEFAULT_SHARED_TM, 176
 WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 176
 WEBLATE_EMAIL_BACKEND, 186
 WEBLATE_EMAIL_HOST, 185
 WEBLATE_EMAIL_HOST_PASSWORD, 185, 186
 WEBLATE_EMAIL_HOST_PASSWORD_FILE, 186
 WEBLATE_EMAIL_HOST_USER, 185
 WEBLATE_EMAIL_PORT, 185, 186
 WEBLATE_EMAIL_USE_SSL, 185, 186
 WEBLATE_EMAIL_USE_TLS, 185, 186
 WEBLATE_ENABLE_AVATARS, 177
 WEBLATE_ENABLE_HOOKS, 177
 WEBLATE_ENABLE_HTTPS, 173, 253
 WEBLATE_ENABLE_SHARING, 178
 WEBLATE_EXTRA_HTML_HEAD, 178
 WEBLATE_GET_HELP_URL, 186
 WEBLATE_GITEA_HOST, 175
 WEBLATE_GITEA_TOKEN, 175
 WEBLATE_GITEA_USERNAME, 175
 WEBLATE_GITHUB_HOST, 175, 242
 WEBLATE_GITHUB_TOKEN, 175
 WEBLATE_GITHUB_USERNAME, 175
 WEBLATE_GITLAB_HOST, 175
 WEBLATE_GITLAB_TOKEN, 175
 WEBLATE_GITLAB_USERNAME, 175
 WEBLATE_GOOGLE_ANALYTICS_ID, 174
 WEBLATE_GPG_IDENTITY, 176
 WEBLATE_HIDE_VERSION, 176
 WEBLATE_INTERLEDGER_PAYMENT_POINTERS, 174
 WEBLATE_IP_PROXY_HEADER, 174
 WEBLATE_LEGAL_URL, 186
 WEBLATE_LICENSE_FILTER, 176
 WEBLATE_LICENSE_REQUIRED, 176
 WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 177
 WEBLATE_LOCALIZE_CDN_PATH, 187
 WEBLATE_LOCALIZE_CDN_URL, 187
 WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 174
 WEBLATE_LOGLEVEL, 171
 WEBLATE_LOGLEVEL_DATABASE, 171
 WEBLATE_NO_EMAIL_AUTH, 183
 WEBLATE_PAGURE_HOST, 175
 WEBLATE_PAGURE_TOKEN, 175
 WEBLATE_PAGURE_USERNAME, 175
 WEBLATE_PRIVACY_URL, 186
 WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN, 178
 WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE, 178
 WEBLATE_RATELIMIT_ATTEMPTS, 177, 417
 WEBLATE_RATELIMIT_LOCKOUT, 177
 WEBLATE_RATELIMIT_WINDOW, 177
 WEBLATE_REGISTRATION_ALLOW_BACKENDS, 173
 WEBLATE_REGISTRATION_OPEN, 173
 WEBLATE_REGISTRATION_REBIND, 173
 WEBLATE_REMOVE_ADDONS, 188
 WEBLATE_REMOVE_APPS, 187
 WEBLATE_REMOVE_AUTOFIX, 188
 WEBLATE_REMOVE_CHECK, 187
 WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 174
 WEBLATE_REQUIRE_LOGIN, 174, 375
 WEBLATE_SAML_IDP_ENTITY_ID, 183
 WEBLATE_SAML_IDP_IMAGE, 183
 WEBLATE_SAML_IDP_TITLE, 183
 WEBLATE_SAML_IDP_URL, 183
 WEBLATE_SAML_IDP_X509CERT, 183
 WEBLATE_SECURE_PROXY_SSL_HEADER, 173, 174
 WEBLATE_SERVER_EMAIL, 172
 WEBLATE_SERVICE, 189
 WEBLATE_SILENCED_SYSTEM_CHECKS, 176, 249
 WEBLATE_SIMPLIFY_LANGUAGES, 175
 WEBLATE_SITE_DOMAIN, 171, 219, 238

WEBLATE_SITE_TITLE, 171
 WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 181
 WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 181
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 181
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET, 181
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID, 181
 WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 180
 WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY, 180
 WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET, 180
 WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, 180
 WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 180
 WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 180
 WEBLATE_SOCIAL_AUTH_FEDORA, 182
 WEBLATE_SOCIAL_AUTH_GITEA_API_URL, 181
 WEBLATE_SOCIAL_AUTH_GITEA_KEY, 181
 WEBLATE_SOCIAL_AUTH_GITEA_SECRET, 181
 WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 180
 WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY, 180
 WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME, 180
 WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET, 180
 WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 180
 WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID, 180
 WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY, 180
 WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET, 180
 WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, 181
 WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 181
 WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, 181
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, 181
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, 181
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS, 181
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, 181
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, 182
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM, 182
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL, 182
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE, 182
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY, 182
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY, 182
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET, 182
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE, 182
 WEBLATE_SOCIAL_AUTH_OIDC_KEY, 182
 WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT, 182
 WEBLATE_SOCIAL_AUTH_OIDC_SECRET, 182
 WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY, 182
 WEBLATE_SOCIAL_AUTH_OPENINFRA, 182
 WEBLATE_SOCIAL_AUTH_OPENSUSE, 182
 WEBLATE_SOCIAL_AUTH_SLACK_KEY, 182
 WEBLATE_SOCIAL_AUTH_UBUNTU, 182
 WEBLATE_SSH_EXTRA_ARGS, 177
 WEBLATE_STATUS_URL, 186
 WEBLATE_TIME_ZONE, 173
 WEBLATE_URL_PREFIX, 176
 WEBLATE_WEBSITE_REQUIRED, 176
 WEBLATE_WORKERS, 170, 188, 189
 WL_BRANCH, 350
 WL_COMPONENT_NAME, 350
 WL_COMPONENT_SLUG, 350
 WL_COMPONENT_URL, 351
 WL_ENGAGE_URL, 351
 WL_FILE_FORMAT, 350
 WL_FILEMASK, 350
 WL_LANGUAGE, 350
 WL_NEW_BASE, 350
 WL_PATH, 350
 WL_PREVIOUS_HEAD, 350
 WL_PROJECT_NAME, 351
 WL_PROJECT_SLUG, 350
 WL_REPO, 350
 WL_TEMPLATE, 350
 WL_VCS, 350

модуль

wlc, 163

wlc.config, 164

wlc.main, 164