



The Weblate Manual

Реліз 4.4

Michal Čihař

груд. 15, 2020

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	Перевірки і виправлення	23
1.6	Пошук	38
1.7	Процеси перекладу	43
1.8	Поширені питання та відповіді на них	47
1.9	Підтримувані формати файлів	54
1.10	Налаштування інтеграції із керуванням версіями	73
1.11	Програмний інтерфейс REST Weblate	80
1.12	Клієнт Weblate	125
1.13	Програмний інтерфейс Python Weblate	130
2	Документація для адміністраторів	133
2.1	Настанови з налаштування	133
2.2	Розгортання Weblate	191
2.3	Оновлення Weblate	192
2.4	Резервне копіювання і пересування Weblate	197
2.5	Розпізнавання	203
2.6	Керування доступом	213
2.7	Проекти перекладу	220
2.8	Визначення мов	237
2.9	Безперервна локалізація	239
2.10	Ліцензування перекладів	249
2.11	Процес перекладу	250
2.12	Перевірки і виправлення	256
2.13	Машинний переклад	264
2.14	Доповнення	271
2.15	Пам'ять перекладів	281
2.16	Налаштування	283
2.17	Зразок налаштувань	311
2.18	Команди керування	326
2.19	Оголошення	337
2.20	Списки складників	339
2.21	Додаткові модулі Weblate	340
2.22	Налаштовування Weblate	345
2.23	Інтерфейс керування	347
2.24	Отримання підтримки щодо Weblate	354
2.25	Правові документи	355

3	Документація для учасників	357
3.1	Участь у розробці Weblate	357
3.2	Перші кроки у програмуванні коду Weblate	358
3.3	Початковий код Weblate	362
3.4	Діагностика у Weblate	363
3.5	Нутрощі Weblate	364
3.6	Розробка додатків	365
3.7	Оболонка Weblate	367
3.8	Надсилання звітів щодо вад у Weblate	368
3.9	Комплекс тестування Weblate і неперервна інтеграція	369
3.10	Схеми даних	370
3.11	Створення випусків Weblate	374
3.12	Про Weblate	375
3.13	Ліцензія	376
4	Журнал змін	377
4.1	Weblate 4.4	377
4.2	Weblate 4.3.2	378
4.3	Weblate 4.3.1	378
4.4	Weblate 4.3	378
4.5	Weblate 4.2.2	379
4.6	Weblate 4.2.1	380
4.7	Weblate 4.2	380
4.8	Weblate 4.1.1	381
4.9	Weblate 4.1	381
4.10	Weblate 4.0.4	382
4.11	Weblate 4.0.3	383
4.12	Weblate 4.0.2	383
4.13	Weblate 4.0.1	383
4.14	Weblate 4.0	384
4.15	Випуски Weblate 3.x	384
4.16	Випуски Weblate 2.x	396
4.17	Випуски Weblate 1.x	408
4.18	Випуски Weblate 0.x	412
	Python Module Index	416
	HTTP Routing Table	417
	Індекс	420

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 одній програмі).

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*, якщо переклади мають відрізнятись.

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. The main content area has a 'Preferences' tab selected, showing various settings. The settings include: 'Hide completed translations on the dashboard' (checkbox), 'Translation editor mode' (dropdown set to 'Full editor'), 'Zen editor mode' (dropdown set to 'Top to bottom'), 'Number of nearby strings' (input set to '15'), 'Show secondary translations in the Zen mode' (checkbox checked), 'Hide source if a secondary translation exists' (checkbox), 'Editor link' (text input), 'Special characters' (text input), and 'Default dashboard view' (radio buttons with 'Watched translations' selected). At the bottom, there's a 'Save' button.

Powered by Weblate 4.4 [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 усіх 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 проекту, 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 проекту за допомогою змінної `SINGLE_PROJECT` у файлі `settings.py` (див. *Налаштування*), панель приладів не буде показано, оскільки користувача буде переспрямовано до сторінки єдиного проекту 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*.

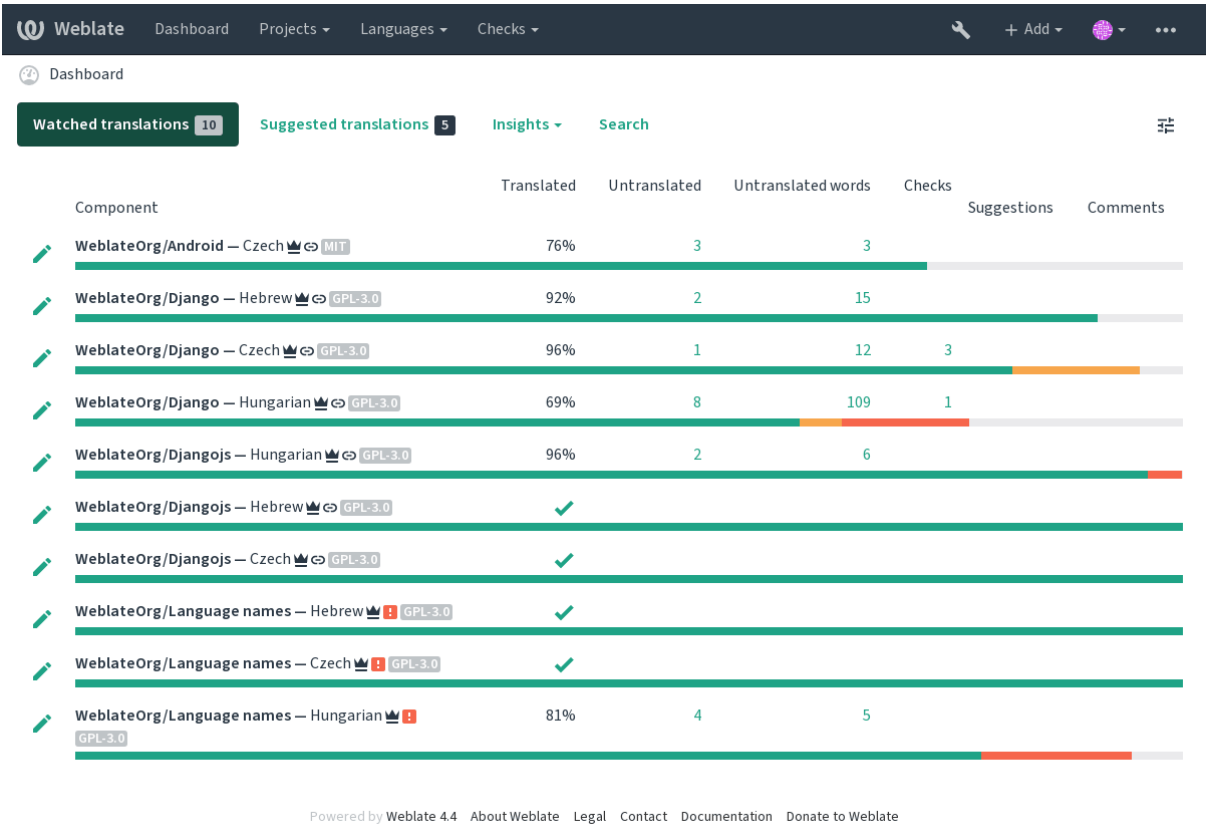
Профіль користувача містить параметри вашого облікового запису. Ім'я і адреса електронної пошти використовуються у внесках до системи керування версіями, тому слід точно вказати ці дані.

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

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

Перекладено мовами

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



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 displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows a translation for the string 'קבצים' (files) in Hebrew. The source language is 'Hebrew' and the target language is 'Czech'. The translation is 'Soubory'. The interface includes buttons for 'Save', 'Suggest', and 'Skip'. A table below shows the status of translations for Hebrew, Hungarian, and English. The right sidebar contains a 'Glossary' section, 'Source information', and 'Screenshot context'.

Language	Status	Translation	Edit
Hebrew	✓	קבצים	Edit
Hungarian	✓	Fájlok	Edit
English	🔍	Files	Edit

Усталений 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

Загальнодоступний профіль

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

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

Посилання на редактор

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

Підказка: Встановленням *Посилання на редактор* ви можете використати ваш локальний редактор для відкриття файла 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*.

1.2.4 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*.



Примітка: Ви не отримуватиме сповіщень щодо ваших власних дій.


Powered by [Weblate 4.4](#) [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.2.5 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.

 Weblate [Dashboard](#) [Projects](#) [Languages](#) [Checks](#) [+ Add](#)  [...](#)

 Your profile

[Languages](#) [Preferences](#) [Notifications](#) **Account** [Profile](#) [Licenses](#) [Audit log](#) [API access](#)

Account

Username

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

Full name






E-mail

You can add another e-mail address below.


Your name and e-mail will appear as commit authorship.

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

1.2.6 API access

You can get or reset your API access token here.

1.2.7 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 усталені процедури перекладу, будь ласка, зверніться до розділу *Процеси перекладу*.

Варіанти видимості проєкту перекладу:

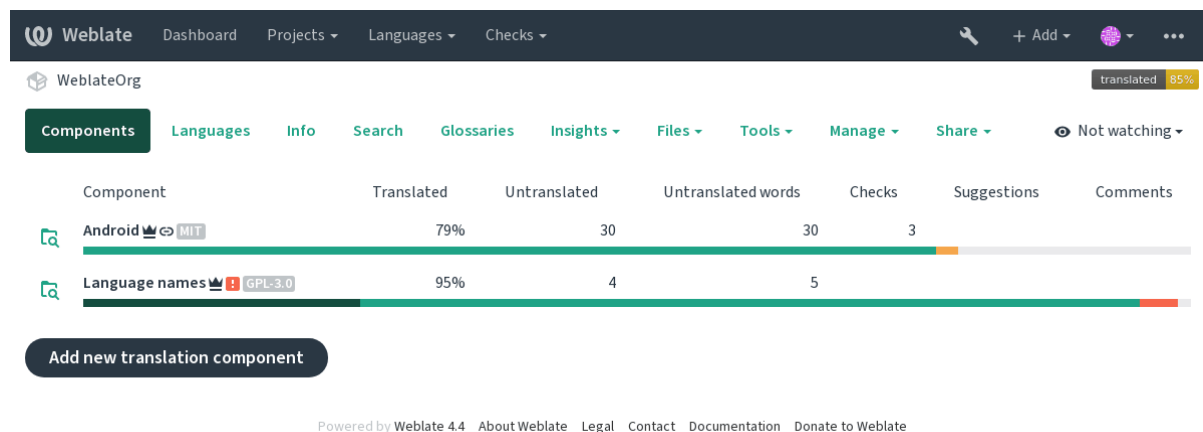
- Видимий для усіх, участь може взяти будь-хто
- Видимий лише певній групі перекладачів

Дивись також:

Керування доступом, Процеси перекладу

1.3.1 Проєкти перекладу

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



1.3.2 Посилання на переклади

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

The screenshot shows the Weblate interface for a project named 'Django' in the 'Czech' language. The top navigation bar includes 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main content area is divided into several sections:

- Translation status:** Shows progress for 'Strings' (26, 96% translated) and 'Words' (183, 93% translated). A 'Translate' button is visible.
- Strings status:** A detailed list of string categories and counts:
 - 26 All strings — 183 words
 - 25 Translated strings — 171 words
 - 1 Strings needing action — 12 words
 - 1 Not translated strings — 12 words
 - 1 Strings needing action without suggestions — 12 words
 - 3 Strings with any failing checks — 11 words
 - 3 Translated strings with any failing checks — 11 words
 - 1 Failed check: Unchanged translation — 4 words
 - 1 Failed check: Mismatched full stop — 4 words
 - 1 Failed check: Python format — 3 words
- Other components:** A table showing the translation status for different components:

Component	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Android	76%	3	3			
Language names	✓					
Djangojs	✓					

A 'Browse all components' button is located at the bottom of the 'Other components' section.

Powered by Weblate 4.4 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.3.3 Пропозиції

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

Анонімні користувачі можуть лише (якщо дозволено) 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

Коментарі можна створювати для двох областей перекладу — початкового рядка і перекладу. 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 не можете його зрозуміти.

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

Дивись також:

report-source

1.3.5 Варіанти

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

Дивись також:

variants

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.

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 декілька початкових рядків і поля для редагування перекладу із описом та мітками форм множини.

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). Якщо ви вибрали другорядні мови у налаштуваннях, над початковим рядком буде показано переклади іншими мовами (див. *Допоміжні мови*).

Під перекладом буде показано усі пропозиції інших перекладачів. Ви можете прийняти їх, прийняти зі змінами або вилучити пропозицію.

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 перекладеній програмі залежать від налаштованої формули форм множини. Weblate показує базову інформацію, але ви можете знайти докладніший опис 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 shows the translation of a string with plural forms. The English source string is '%(count)s word'. The Czech translation for the singular form is '%(count)s slovo' and for the plural form is '%(count)s slova'. The interface includes buttons for 'Save', 'Suggest', and 'Skip'. The right sidebar contains a 'Glossary' section, 'Source information' (including screenshot context, explanation, labels, and flags), and 'String age' (10 seconds ago). The bottom section shows a 'New comment' form for translators and developers to discuss the translation.

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 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.
Alt Enter або Ctrl Enter або Cmd Enter	Зберегти поточний переклад.
Ctrl Shift Enter або Cmd Shift Enter	Скасовує позначення перекладу як неточного і надсилає його до сховища.
: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. Автоматичні пропозиції .
Від Ctrl 1 до Ctrl 9 або Від Cmd 1 до Cmd 9	Копіює заміник із вказаним номером з початкового рядка.
Ctrl M від 1 до 9 або Cmd M від 1 до 9	Копіює комп'ютерний переклад із вказаним номером до поточного перекладу.
Ctrl I від 1 до 9 або Cmd I від 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	Передає фокусування до поля пошуку.
Ctrl O або Cmd O	Копіює початковий рядок.
:kbd: Ctrl Y або Cmd Y	Перемикає прапорець <i>Потребує редагування</i> .

Візуальна клавіатура

Оразу над полем перекладу буде показано невеличку візуальну клавіатуру. Ця клавіатура може бути корисною для введення символів, яких немає на клавіатурі або які важко ввести.

Показані символи розподілено на три категорії:

- Налаштовані користувачем 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. The top navigation bar includes 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main content area is titled 'WeblateOrg / Django / Hebrew / Translate' and shows a translation progress of 92%. The central 'Translation' panel displays the source string 'Files' and the target string 'קבצים' (Kvitsim) in Hebrew. Below the target string, there are buttons for 'Save', 'Suggest', and 'Skip'. To the right, a sidebar contains sections for 'Glossary', 'Source information', and 'String age'. At the bottom, a table titled 'Nearby strings' shows translations for Czech, Hungarian, and English.

Language	Status	Translation	Edit
Czech	✓	Soubory	Edit
Hungarian	✓	Fájlok	Edit
English	🔍	Files	Edit

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 додаткову інформацію 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 перекладу Погляд на подібні рядки, які вже було перекладено, див. *Memory Management*.

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 відомості щодо сховища системи керування версіями проекту.

Якщо у форматі перекладу передбачено підтримку такої можливості, ви також можете переходити за вказаними посиланнями до відповідної ділянки коду, який містить початковий рядок.

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 удесятеро 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 Глосарій

З метою зберігання термінології із кожним проектом можна пов'язати глосарій для будь-якої мови. За допомогою глосаріїв можна сприяти однорідності перекладів. Терміни з поточного перекладеного рядка може бути показано під вкладками.

Керування глосаріями

За допомогою вкладки *Глосарії* кожної сторінки проекту ви можете редагувати наявні глосарії.

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Glossaries

Components Languages Info Search **Glossaries** Insights Files Tools Manage Share

WeblateOrg

Catalan 0 Czech 1 Dutch 0 English 0 French 0 Galician 0 German 0 Hebrew 0 Hungarian 0
 Chinese (Simplified) 0 Polish 0 Russian 0 Spanish 0

Glossary name
 WeblateOrg

Color
 Navy Blue Aqua Teal Olive Green Lime Yellow Orange Red Maroon Fuchsia Purple Black Gray Silver

Source language
 English

Additional projects
 Search...
 Available: Chosen:

Choose additional projects where this glossary can be used.

Save Delete

Create new glossary

Glossary name

Color
 Navy Blue Aqua Teal Olive Green Lime Yellow Orange Red Maroon Fuchsia Purple Black Gray Silver

Source language
 English

Additional projects
 Search...
 Available: Chosen:

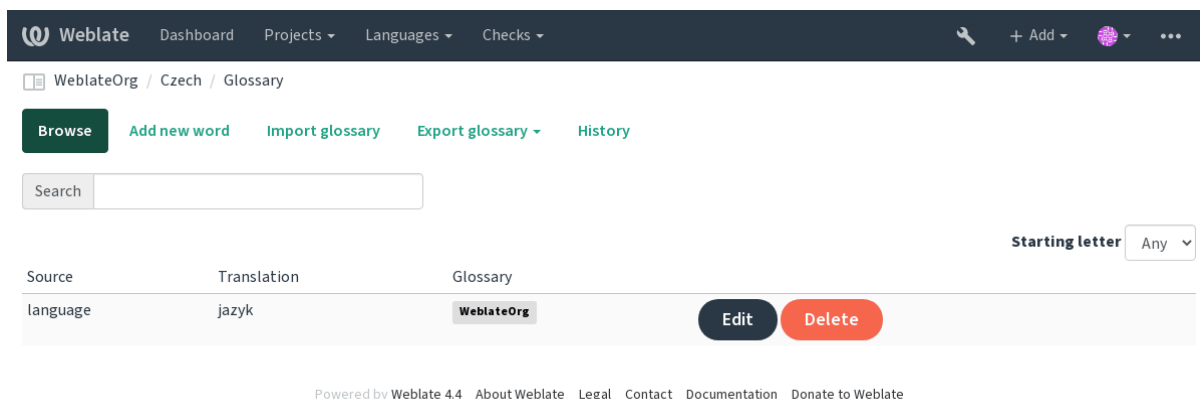
Choose additional projects where this glossary can be used.

Save

Під час створення проєкту автоматично створюється порожній глосарій. Глосарії є спільними для усіх складників одного проєкту. Ви можете налаштувати систему так, щоб глосарій був спільним для декількох проє-

ктів. Налаштовувати глосарії можна лише для тих проєктів, які ви адмініструєте.

У цьому списку ви можете вибрати, яким глосарієм слід керувати (буде показано усі мови у поточному проєкті). Перехід за одним з посиланням мов відкриє сторінку, за допомогою якої ви можете редагувати, імпортувати або експортувати вибраний глосарій або переглядати журнал редагувань:



1.3.9 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 *Машинний переклад*.

1.3.10 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:

Automatic translation

Automatic translation takes existing translations in this project and applies them to the selected strings. It can be used to push translations to a different branch, to fix inconsistent translations or to translate a new component using existing translations.

Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.

Automatic translation mode

Add as suggestion

Search filter

Strings needing action

Automatic translation source

☐ Other translation components ☒ Machine translation

Machine translation engines

Search...

Available:	Chosen:
Weblate Weblate Translation Memory	Weblate

Score threshold

80

Apply

Powered by Weblate 4.4 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Можна користуватися двома способами дій:

- Skorистатися іншими складниками Weblate як джерелом перекладів.
- Skorистатися 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, веб-сайтом і програмою) або при започаткуванні перекладу для нового складника на основі наявних перекладів (пам'яті перекладів).

Дивись також:

Підтримання синхронізації перекладів у різних складниках

1.3.11 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 описано у підручнику з адміністрування у розділі *Обмеження частоти*, але їх можна скоригувати у налаштуваннях.

1.3.12 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 — ризик невиправної шкоди мінімальний. Процедура поділена на два кроки: програма покаже вам попередній перегляд редагувань, перш ніж справжні зміни буде внесено.

1.3.13 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. *Налаштовування поведінки*)
- Коригування міток рядків (див. labels)

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

Дивись також:

Додаток пакетного редагування

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 пункти може бути обмежено *Керування доступом*.

1.4.1 Отримання перекладів

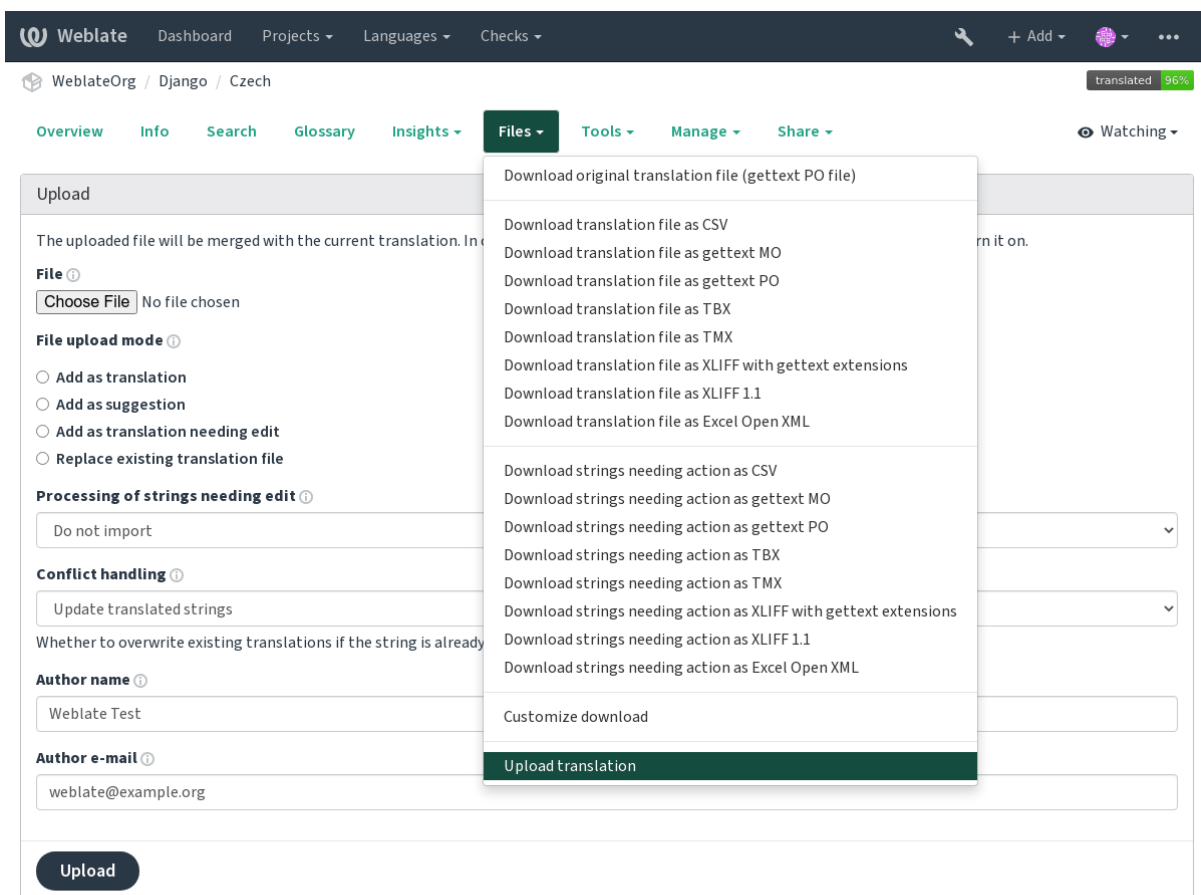
З панелі прикладів проекту або складника, придатні до перекладу файли можна отримати за допомогою пункту *Звантажити початковий файл перекладу* у меню *Файли*. У результаті буде створено копію початкового файла із тим самим вмістом, що і в основній системі керування версіями.

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.

Доступними є декілька форматів файлів, зокрема зібраний файл, яким можна скористатися у вибраній вами програмі (наприклад, файл ``.mo`` для GNU Gettext). Відповідний варіант можна вибрати за допомогою меню *Файли*.

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

Вивантажений файл об'єднується із наявними даними для оновлення перекладу. Типово, наявні записи буде оновлено (цю можливість можна вимкнути на сторінці вивантаження даних).

Способи імпортування

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

Додати як переклад (*translate*) Імпортовані переклади буде додано як переклади. Це найпоширеніший варіант, який є типовим.

Додати як пропозицію (*suggest*) Імпортовані переклади буде додано як пропозиції. Виберіть цей варіант, якщо ви хочете вивантажити переклади для рецензування.

Додати як переклад, що потребує редагування (*fuzzy*) Імпортовані переклади буде додано як переклади, які потребують редагування. Цей варіант може бути корисним, якщо ви хочете, щоб переклади було використано, але також і рецензовано.

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)*.

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 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:

W Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

🔧

+ Add ▾

...

WeblateOrg / Django / Czech

translated 96%

The translation has come to an end.

Overview

Info

Search

Glossary

Insights ▾

Files ▾

Tools ▾

Manage ▾

Share ▾

Watching ▾

Translation status

26 Strings

96%

183 Words

93%

Translate

Strings status

26

All strings — 183 words

25

Translated strings — 171 words

1

Strings needing action — 12 words

1

Not translated strings — 12 words

1

Strings needing action without suggestions — 12 words

3

Strings with any failing checks — 11 words

3

Translated strings with any failing checks — 11 words

1

Failed check: Unchanged translation — 4 words

1

Failed check: Mismatched full stop — 4 words

1

Failed check: Python format — 3 words

Other components

Component	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Android MIT	76%	3	3			
Language names GPL-3.0	✓					
Djangojs GPL-3.0	✓					

Browse all components

Powered by Weblate 4.4

About Weblate

Legal

Contact

Documentation

Donate to Weblate

1.5.1 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

1.5.2 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*

1.5.3 Перевірки перекладів

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

Розмітка BBcode

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

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

Текст містить те саме слово двічі поспіль:

Нове в версії 4.1.

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 *Надсилення звітів щодо вад у Weblate*.

Подвійний пробіл

Переклад містить подвійну прогалину

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 означати-
ме, що подвійний пробіл було використано навмисно.

Форматовані рядки

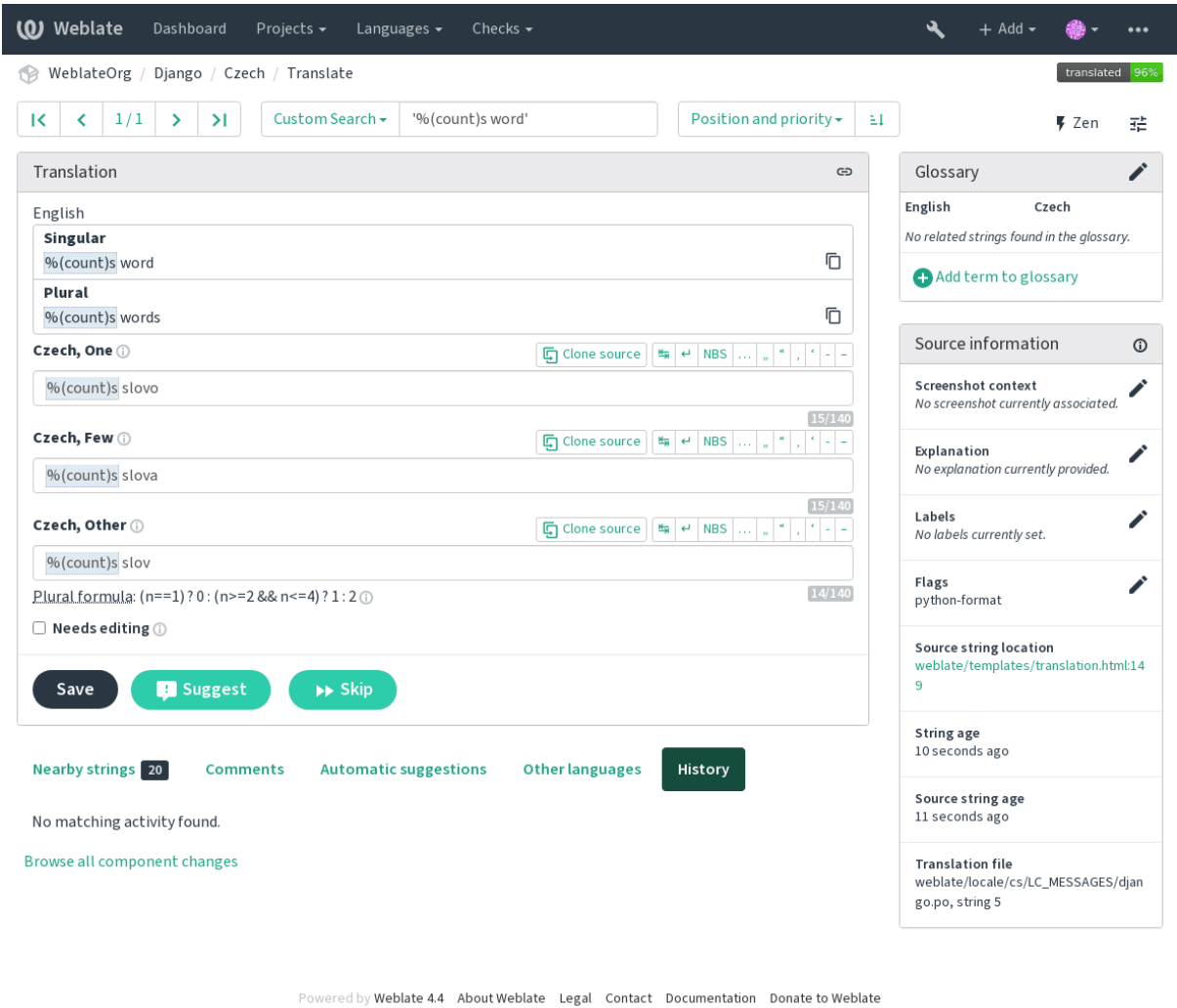
Перевіряє, чи є однаковим форматування у початковому рядку і перекладі. Пропускання рядків форматування у перекладі, зазвичай, спричиняє серйозні проблеми, тому форматування у рядках перекладів має, зазвичай, збігатися із форматуванням у початковому рядку.

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:



Рядок інтерполяції AngularJS

Рядки інтерполяції AngularJS не відповідають джерелу

Іменований рядок форматування	Your balance is {{amount}} {{ currency }}
Прапорець для вмикання	angularjs-format

Дивись також:

AngularJS: API: \$interpolate

Формат C

Рядок формату C# не відповідає джерелу

Простий рядок форматування	There are %d apples
Позиційний рядок форматування	Your balance is %1\$d %2\$s
Прапорець для вмикання	c-format

Дивись також:

Рядки форматування C, форматування printf C

Формат C#

Рядок формату C# не відповідає джерелу

Позиційний рядок форматування	There are {0} apples
Прапорець для вмикання	<i>c-sharp-format</i>

Дивись також:

Форматування рядків C#

Літерали шаблонів ECMAScript

Літерали шаблонів ECMAScript не відповідають джерелу

Інтерполяція	There are \${number} apples
Прапорець для вмикання	<i>es-format</i>

Дивись також:

Літерали шаблонів

i18next інтерполяція

Інтерполяція i18next не відповідає джерелу

Нове в версії 4.0.

Інтерполяція	There are {{number}} apples
Вкладення	There are \$t(number) apples
Прапорець для вмикання	<i>i18next-interpolation</i>

Дивись також:

Інтерполяція i18next

Формат Java

Рядок формату Java не відповідає джерелу

Простий рядок форматування	There are %d apples
Позиційний рядок форматування	Your balance is %1\$d %2\$s
Прапорець для вмикання	<i>java-format</i>

Дивись також:

Рядки форматування Java

Формат Java Message

Рядок формату Java Message не відповідає джерелу

Позиційний рядок форматування	There are {0} apples
Прапорець для вмикання	<i>java-messageformat</i> вмикає перевірку безумовно
	<i>auto-java-messageformat</i> вмикає перевірку, лише якщо у початковому рядку є рядок форматування

Дивись також:

[MessageFormat Java](#)

Формат JavaScript

Рядок, формату JavaScript, не відповідає джерелу

Простий рядок форматування	There are %d apples
Прапорець для вмикання	<i>javascript-format</i>

Дивись також:

[Рядки форматування JavaScript](#)

Відсоток заповнювачів

Відсоток заповнювачів не відповідає джерелам

Нове в версії 4.0.

Простий рядок форматування	There are %number% apples
Прапорець для вмикання	<i>percent-placeholders</i>

Формат Perl

Рядок формату Perl не відповідає джерелу

Простий рядок форматування	There are %d apples
Позиційний рядок форматування	Your balance is %1\$d %2\$s
Прапорець для вмикання	<i>perl-format</i>

Дивись також:

[Perl sprintf, Рядки форматування Perl](#)

Формат PHP

Рядок, формату PHP, не відповідає джерелу

Простий рядок форматування	There are %d apples
Позиційний рядок форматування	Your balance is %1\$d %2\$s
Прапорець для вмикання	<i>php-format</i>

Дивись також:

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

Формат дужок Python

Рядок, формату Python brace, не відповідає джерелу

Простий рядок форматування	There are {} apples
Іменованний рядок форматування	Your balance is {amount} {currency}
Прапорець для вмикання	<i>python-brace-format</i>

Дивись також:

[Форматування із фігурними дужками у Python](#), [Рядки форматування у Python](#)

Формат Python

Рядок форматування Python не відповідає джерелу

Простий рядок форматування	There are %d apples
Іменованний рядок форматування	Your balance is %(amount) %(currency)
Прапорець для вмикання	<i>python-format</i>

Дивись також:

[Форматування рядків у Python](#), [Рядки форматування у Python](#)

Формат Qt

Рядок формату Qt не відповідає джерелу

Позиційний рядок форматування	There are %1 apples
Прапорець для вмикання	<i>qt-format</i>

Дивись також:

[Qt QString::arg\(\)](#)

Формат Qt у множині

Рядок формату Qt у множині не відповідає джерелу

Рядок форматування форм множини	There are %Ln apple(s)
Прапорець для вмикання	<i>qt-plural-format</i>

Дивись також:

[Підручник з i18n Qt](#)

Формат Ruby

Рядок формату Ruby не збігається з джерелом

Простий рядок форматування	There are %d apples
Позиційний рядок форматування	Your balance is %1\$f %2\$s
Іменованний рядок форматування	Your balance is %+.2<amount>f %<currency>s
Рядок із іменованим шаблоном	Your balance is %{amount} %{currency}
Прапорець для вмикання	<i>ruby-format</i>

Дивись також:

[Ruby Kernel#sprintf](#)

Форматування i18n Vue

Форматування i18n Vue не збігається із початковим рядком

Іменоване форматування	There are {count} apples
Форматування i18n Rails	There are %{count} apples
Пов'язані повідомлення локалі	@:message.dio @:message.the_world!
Прапорець для вмикання	<i>vue-format</i>

Дивись також:

[Форматування I18n Vue, пов'язані повідомлення локалі i18n Vue](#)

Вже перекладено

Цей рядок перекладено раніше

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

Неузгодженість

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

Weblate шукає різні переклади однакових рядків в 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*.

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

Декоративні букви кашиди не повинні використовуватися

Нове в версії 3.5.

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

Дивись також:

Кашида у Вікіпедії

Посилання Markdown

Посилання Markdown не відповідають джерелу

Нове в версії 3.5.

Посилання Markdown не відповідають джерелу.

Дивись також:

Посилання Markdown

Markdown посилання

Посилання Markdown не збігаються з джерелом

Нове в версії 3.5.

Посилання Markdown не збігаються з джерелом.

Дивись також:

Посилання Markdown

Синтаксис Markdown

Синтаксис Markdown не відповідає джерелу

Нове в версії 3.5.

Синтаксис Markdown не відповідає джерелу

Дивись також:

[Елементи розмітки span](#)

Максимальна довжина перекладу

Переклад не перевищує заданої довжини

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.

Найбільший розмір перекладу

Перекладений текст не повинен перевищувати заданий розмір

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. [Налаштовування поведінки](#)). 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 усіх рядків у складнику. Ви можете перевизначити значення на рівні рядка, якщо вам потрібно якимось чином виконати специфічну перевірку для окремого рядка.

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

Дивись також:

[Керування шрифтами](#), [Налаштовування поведінки](#), [Максимальна довжина перекладу](#)

Не збіглися \n

Число \n у перекладі не відповідає джерелу

Зазвичай, екрановані символи розриву рядка є важливими для форматування виведених програмою даних. Перевірка вважається не пройденою, якщо кількість записів \n у перекладі не збігається із кількістю у початковому рядку.

Невідповідність двокрапки

Джерело та переклад не закінчуються двокрапкою

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

Джерело і переклад не завершуються трикрапкою

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

Джерело та переклад не закінчуються знаком оклику

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

Джерело і переклад не завершуються кінцевою крапкою

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

Джерело та переклад не закінчуються знаком питання

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

Джерело і переклад не завершуються крапкою з комою

Перевіряє, чи відтворено крапки з комами наприкінці рядків з початкового рядка у перекладі. Це може бути корисним для збереження форматування записів, зокрема у файлах .desktop.

Дивись також:

[Крапка з комою у Вікіпедії](#)

Розбіжність розривів рядків

Кількість знаків нових рядків у перекладі не збігається з джерелом

Зазвичай, символи розриву рядка є важливими для форматування виведених програмою даних. Перевірка вважається не пройденою, якщо кількість записів `\n` у перекладі не збігається із кількістю у початковому рядку.

Бракує множини

Деякі форми множини не перекладено

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

У перекладі відсутні деякі замітники:

Нове в версії 3.9.

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

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 певну синтаксичну конструкцію для заміників, ви можете скористатися формальним виразом:

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

Дивись також:*Налаштовування поведінки***Щільність знаків***Пропущений нерозривний пробіл перед подвійним знаком пунктуації*

Нове в версії 3.9.

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***Переклад не відповідає регулярному виразу:*

Нове в версії 3.9.

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

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

Однакова множина*Деякі форми множини перекладені однаково*

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

Перехід у новий рядок*Джерело і переклад не починаються з переходу у новий рядок*

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***Джерело і переклад не починаються з однакового числа пропусків*

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

Кінцевий знак нового рядка

Джерело і переклад не завершуються переходом у новий рядок

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

Джерело і переклад не завершуються пропуском

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

Джерело і переклад збігаються

Вважається не пройденою, якщо початковий рядок збігається із перекладом, аж до принаймні одної з форм множини. Деякі рядки, які є типовими для усіх мов буде проігноровано, а розмітку при перевірці вилучено. Таким чином Weblate уникає помилок при перевірці.

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

Небезпечний HTML-код

Переклад використовує небезпечну HTML-розмітку

Нове в версії 3.9.

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

Дивись також:

Перевірка HTML виконується бібліотекою [Bleach](#), яку розроблено Mozilla.

URL

Цей переклад не містить URL

Нове в версії 3.5.

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

Теги XML у перекладі не збігаються з тегами XML у джерелі

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.

Синтаксис XML

Цей переклад не є правильним XML

Нове в версії 2.8.

Розмітка XML є некоректною.

Простір нульової ширини

Переклад містить зайві символи пропуску нульової ширини

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](#)

1.5.4 Перевірки джерела

Перевірки початкових рядків можуть допомогти розробникам удосконалити початкові рядки.

Трикрапка

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

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](#)

Тривалий час не перекладено

Цей рядок вже тривалий час не перекладено

Нове в версії 4.1.

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

Кратні невдалі перевірки

Переклади на декількох мовах мають непройдені перевірки

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

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

Нове в версії 4.1.

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

Рядок використовується як множина, але не використовує форми множини

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

1.6 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 полях *Джерело*, *Переклад* і *Контекст*.

Search

Custom Search ▾ Search for... Sort By ▾

Advanced query builder

Source strings ▾ Search for... ☐ Exact Add String has suggestion ▾ Add

String changed after ▾ mm/dd/yyyy ☐ Add

Query examples

Review strings changed by other users	changed:>=2020-11-14 AND NOT changed_by:testuser	Add
Translated strings	state:>=translated	Add
Strings with comments	has:comment	Add
Strings with any failing checks	has:check	Add
Strings with suggestions from others	has:suggestion AND NOT suggestion_author:testuser	Add
Approved strings with suggestions	state:approved AND has:suggestion	Add
All untranslated strings added the past month	added:>=2020-11-14 AND state:<=needs-editing	Add
Translated strings in a certain language	is:translated AND language:cs	Add

Search

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

1.6.1 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'.

1.6.2 Поля

source:ТЕКСТ Пошук у початковому рядку без врахування регістру символів.

target:ТЕКСТ Пошук у рядку перекладу без врахування регістру символів.

context:ТЕКСТ Пошук у рядку контексту без врахування регістру символів.

key:ТЕКСТ Пошук у рядку ключа без врахування регістру символів.

note:ТЕКСТ Пошук у рядках коментарів без врахування регістру символів.

location:ТЕКСТ Пошук у рядку розташування без врахування регістру символів.

priority:ЧИСЛО Пріоритетність рядка.

added:ДАТА_ЧАС Часова позначка додавання рядка до Weblate.

state:ТЕКСТ Пошук за станом (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 (працює лише для рядків початковою мовою).

is:ТЕКСТ Шукати за станом рядка (pending, translated, untranslated).

language:ТЕКСТ Рядок цільової мови.

component:ТЕКСТ Ідентифікатор складника, див. *Ідентифікатор складника*.

project:ТЕКСТ Ідентифікатор проєкту, див. *Ідентифікатор проєкту*.

changed_by:ТЕКСТ Рядок було змінено автором із вказаним ім'ям користувача.

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

change_time:DATEIME 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 замінені дефісами. Див. приклади для *Пошук змін*.

check:ТЕКСТ Рядок не пройшов перевірки.

dismissed_check:ТЕКСТ Перевірку для рядка скасовано.

comment:ТЕКСТ Шукати у коментарях користувачів.

comment_author:ТЕКСТ Фільтрувати за автором коментаря.

suggestion:ТЕКСТ Шукати у пропозиціях.

suggestion_author:ТЕКСТ Фільтрувати за автором пропозиції.

1.6.3 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)

1.6.4 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.

1.6.5 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.

1.6.6 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"`.

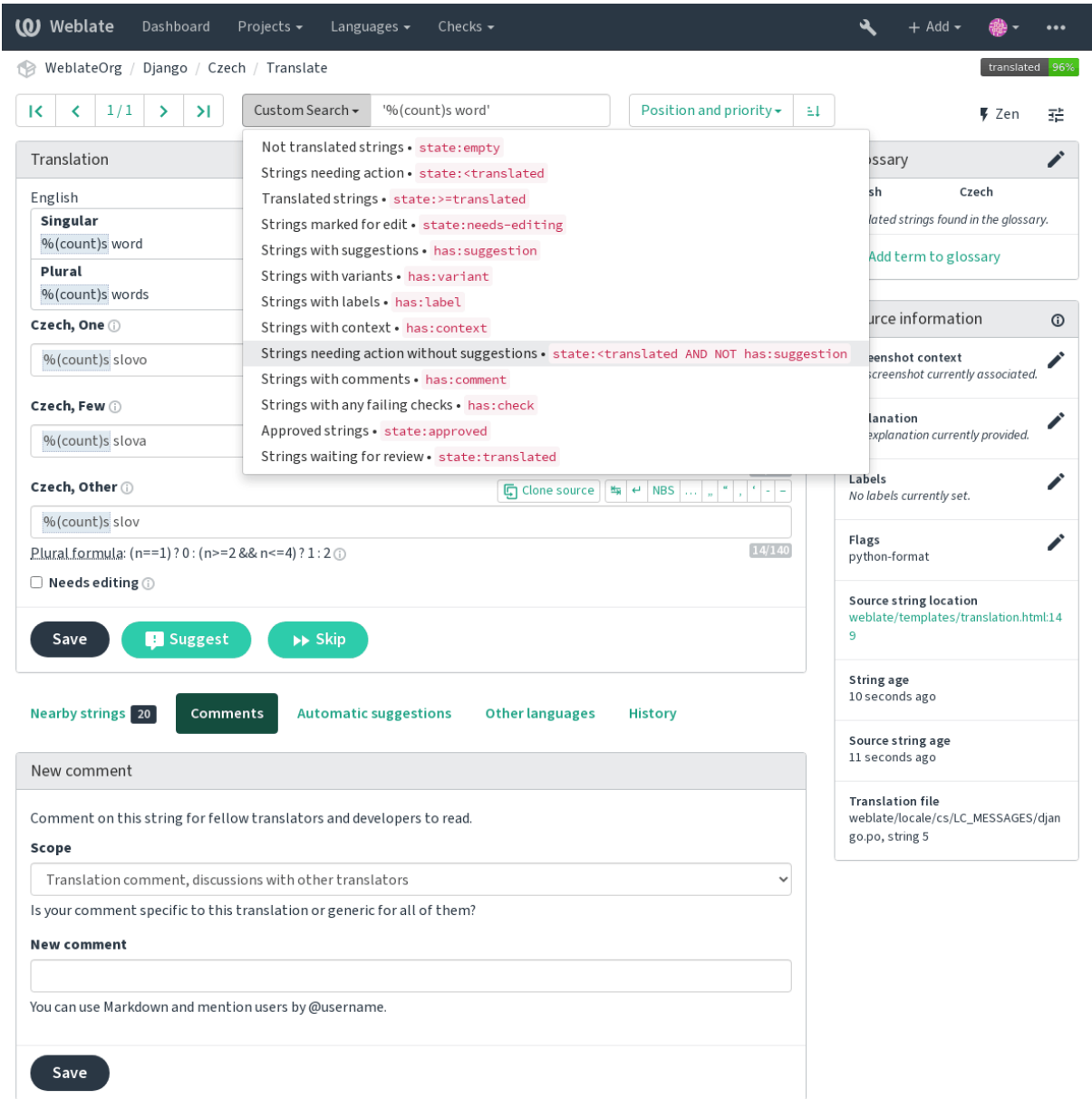
1.6.7 Формальні вирази

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]"`.

1.6.8 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:



Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

1.6.9 Упорядковування результатів

Передбачено багато варіантів упорядкування, відповідно до ваших потреб:

The screenshot displays the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, the breadcrumb 'WeblateOrg / Django / Czech / Translate' is visible. A progress indicator shows 'translated 96%'. The main area features a translation form with 'English' and 'Czech' sections. The English text is 'The string uses three dots (...) instead of an ellipsis character (...)'. The Czech section is empty. A 'Clone source' button is next to the Czech input. Below the form are 'Save', 'Suggest', and 'Skip' buttons. A dropdown menu is open, listing various metadata fields. To the right, a sidebar contains a 'Glossary' section and 'Source information' 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 with a text area and a 'Save' button.

Powered by Weblate 4.4 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.7 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.

1.7.1 Доступ до перекладу

Питання *прав доступу* 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 проект є відкритим, або користувач, який має права доступу до проекту рівня *Translate*.

1.7.2 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. Переклад зберігається у файлі. Залежно від формату файла, його може бути позначено як такий, 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.

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

1.7.3 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 <i>Перекласти</i> із <i>Керування доступом</i> .
Група рецензентів	Н/Д	Не використовується.

1.7.4 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 проєкту.
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>Перекласти</i> із <i>Керування доступом</i> .
Група рецензентів	Н/Д	Не використовується, рецензування усіх перекладачів.

1.7.5 Спеціалізовані рецензенти

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

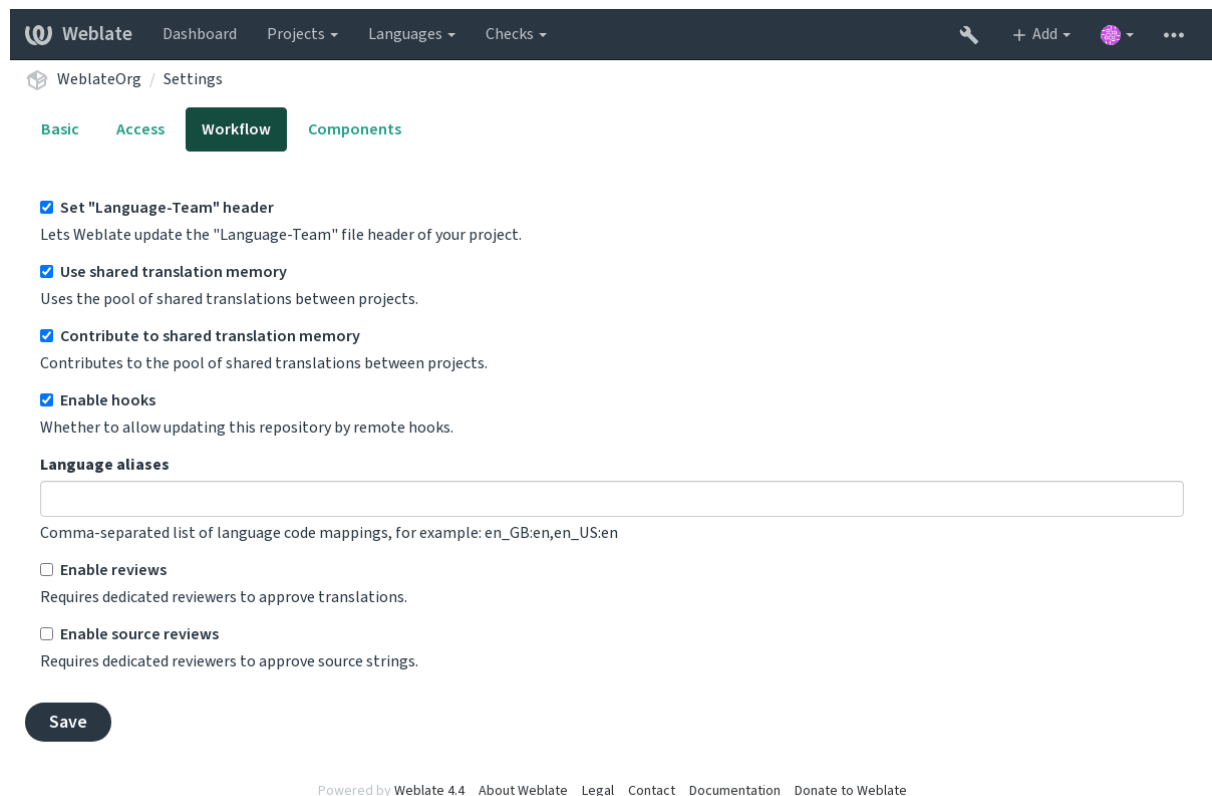
Якщо у проєкті є спеціалізовані рецензенти, ви матимете дві групи користувачів: одна зможе надсилати переклади, а інше рецензувати їх з метою забезпечення коректності та якості.

- *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 проєкту.
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>Перекласти</i> із <i>Керування доступом</i> .
Група рецензентів	<i>Рецензенти</i>	Або <i>Рецензувати</i> із <i>Керування доступом</i> .

1.7.6 Вмикання рецензування

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

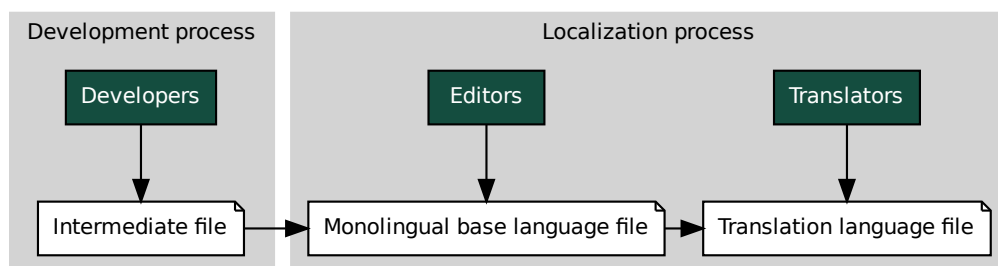


Примітка: Залежно від налаштувань Weblate, цей пункт може виявитися недоступним. Наприклад, на Hosted Weblate він недоступний для проєктів, які працюють на безкоштовній основі.

1.7.7 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

1.7.8 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

1.8 Поширені питання та відповіді на них

1.8.1 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 сховища, переклад відбуватиметься без вашого втручання, якщо не станеться якихось конфліктів об'єднання.

1. Налаштуйте ваше сховище Git так, щоб Weblate отримував сповіщення при кожній зміні, див. *скрипти сповіщення*, щоб дізнатися більше про те, як це зробити.
2. Встановіть адресу запису у розділі складника у Weblate, це надасть змогу Weblate записувати зміни до вашого сховища.
3. Увімкніть запис після внеску у параметрах вашого *проєкту* на Weblate — Weblate записуватиме зміни до вашого сховища, коли вони траплятимуться на Weblate.

Дивись також:

Безперервна локалізація, Уникання конфліктів об'єднання

Як отримувати доступ до сховища за допомогою SSH?

Будь ласка, ознайомтеся із розділом *Доступ до сховищ*, 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 сховищі коду. Зазвичай, цих конфліктів можна уникнути об'єднанням перекладів Weblate зі сховищем до внесення змін до файлів перекладів (наприклад, до запуску msgmerge). Просто накажіть Weblate вносити усі переклади з черги (зробити це можна за допомогою пункту *Супровід сховища* у меню *Управління*) і виконайте об'єднання коду зі сховищем (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 віддалене сховище, об'єднайте його із основною гілкою розробки і виправте усі конфлікти. Щойно ви запишете зміни до основного сховища, Weblate зможе використовувати об'єднану версію без будь-яких інших додаткових дій.

Примітка: Залежно від конфігурації, доступ до сховища 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/master

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

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

# Open Weblate for translation:
wlc unlock
```

Якщо ви використовуєте у Weblate декілька гілок, ви можете зробити те саме для усіх цих гілок:

```
# 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 master branch:
git checkout master
```

(continues on next page)

(continued from previous page)

```
git merge weblates-second/master
... # 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 сховища Git Weblate. Також отримайте другий свіжий локальний клон основного сховища Git (тобто, вам знадобляться дві копії сховища 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/master

# 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 сховище Git, яке використовує Weblate?, Безперервна локалізація, Уникання конфліктів об'єднання

Як налаштувати одночасний переклад у декількох гілках розробки?

У Weblate передбачено підтримку запису змін у перекладах у межах одних *налаштувань проекту*. Для кожних *налаштувань складника*, для якого увімкнено 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 усталеної 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 усіх 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 усіх платформ одночасно.

Дивись також:

Підтримання синхронізації перекладів у різних складниках

Як експортувати сховище Git, яке використовує Weblate?

У цьому сховищі 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 використовувати сховище безпосередньо.

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

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

Якими є варіанти надсилання змін назад до основного сховища?

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 сховища).
- Хтось вручну об'єднує зміни зі сховища git Weblate до основного сховища.
- 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 сховищі.

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 сховище 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 сховищем — йому більше не знадобиться доступ до сховища, яке містить ваш початковий код.
4. Ви можете оновлювати основне сховища 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.

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

Це типове ім'я автора внеску, яке було налаштовано при створенні складника перекладу. Ви можете будь-коли змінити його за допомогою адміністративного інтерфейсу.

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

Дивись також:

Налаштовування складників

1.8.2 Користування

Як рецензувати переклади інших користувачів?

- Ви можете підписатися на усі зміни у розділі *Сповіщення*, а потім перевіряти внески інших учасників перекладу, коли сповіщення надходитимуть 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.

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?

- 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:

Машинний переклад, Автоматичні пропозиції

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.

Для одномовних файлів (див. *Підтримувані формати файлів*) Weblate може додавати нові рядки перекладу, яких немає у *шаблоні*, а не у самих перекладах. Втім, він не виконує ніякого автоматичного вилучення застарілих рядків, як можна було б очікувати. Якщо вам потрібне таке очищення, встановіть відповідний додаток, який виконає чищення перекладу відповідно до ваших вимог.

Крім того, Weblate не намагатиметься оновлювати двомовні файли у будь-який спосіб, тому якщо вам потрібне оновлення файлів `po` на основі `pot`, вам слід виконати оновлення власноруч чи застосувавши додаток.

Дивись також:

Обробка сховища зі скриптами, Очищення файлів перекладу, Вилучити порожні рядки, Оновити файли RESX, Оновити файли po, порівнявши із pot (msgmerge)

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.

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 сховища, 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 програми **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)*.

1.8.3 Вирішення проблем

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

Таке трапляється, якщо ваше сховище Git розростається і у вас стає забагато файлів. Стискання сховища 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 сайта 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 типово 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 `:ref:` початкову мову `<component-source_language>` у налаштуваннях складника.
- Якщо файл перекладу для початкової мови є непотрібним, будь ласка вилучіть його зі сховища.
- Якщо файл перекладу для початкової мови є бажаним, але Weblate має його ігнорувати, будь ласка, скоригуйте *Фільтр мов*, щоб виключити його.

1.8.4 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 *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. Якщо вам потрібно, щоб ваш проєкт перекладався як одне ціле, вам варто об'єднати усі файли PO в один. Так можна спростити життя навіть тих перекладачів, які не використовують Weblate.

Примітка: Якщо відповідна можливість матиме високий рейтинг серед користувачів, її може бути реалізовано у майбутніх версіях.

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

Ці коди мови визначено **RFC 4646** 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`.

1.9 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 вашої програми слід надавати перевагу усталеним форматам на платформі або у наборі інструментів, яким ви користуєтесь. Якщо ви це зробите, перекладачі зможуть кори-

стуватися знайомими їм інструментами, що збільшить ймовірність того, що вони візьмуть участь у вашому проєкті.

1.9.1 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.

1.9.2 Автоматичне виявлення

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

1.9.3 Можливості типів перекладу

Можливості усіх підтримуваних форматів:

Формат	Кількість мов ¹	Форми множини ²	Коментарі ³	Контекст ⁴	Розташування ⁵	Відношення ⁸	Додаткові стани 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>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 ¹⁰	потребує редагування

continues on next page

Table 1 – continued from previous page

Формат	Кількість мов ¹	Форми множини ²	Коментарі ³	Контекст ⁴	Розташування ⁵	Відношення ⁸	Додаткові стани ⁶
<i>Рядкові ресурси Android</i>	одномовний	так	так ⁷	ні	ні	так ¹⁰	
<i>Рядки Apple iOS</i>	двомовний	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	ні	ні	ні	ні	
<i>Файл ARB</i>	одномовний	так	так	ні	ні	ні	
<i>JSON WebExtension</i>	одномовний	так	так	ні	ні	ні	
<i>Файли ресурсів .XML</i>	одномовний	ні	так	ні	ні	так 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>Flat XML</i>	одномовний	ні	ні	ні	ні	так 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 метаданих 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	

continues on next page

Table 1 – continued from previous page

Формат	Кількість мов ¹	Форми множини ²	Коментарі ³	Контекст ⁴	Розташування прапорці ⁸	Додаткові стани ⁶
<i>Переклади INI Inno Setup</i>	одномовний	ні	ні	ні	ні	ні

1.9.4 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"
```

Типовий <i>Налаштовування складників Weblate</i>	
Маска файла	po/* .po
Основний файл для одномовного перекладу	<i>Порожньо</i>
Заготовка 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 mo-файли*, *Оновити файли po*, *порівнявши із pot (msgmerge)*

¹ Див. *Двомовні і одномовні формати*

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

³ Коментарями можна скористатися для передавання додаткових відомостей щодо рядка, який слід перекласти.

⁴ Контекст використовується для розрізнення ідентичних рядків, які використовуються у різних парадигмах (наприклад, *Sun* 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*

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

⁹ Коментарі типу gettext використовуються як прапорці.

¹⁰ Прапорці видобуваються з нестандартного атрибута weblate-flags для усіх 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 коментар розробника.

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

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 файлу	po/* .po
Основний файл для одномовного перекладу	po/en .po
Заготовка для нових перекладів	po/messages .pot
Формат файлу	<i>файл PO Gettext (одномовний)</i>

1.9.5 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.

Дивись також:

Специфікація *XML Localization Interchange File Format (XLIFF)*

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.

Див. *Спеціалізовані рецензенти*.

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

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. *Налаштування поведінки*) 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 до модуля або контекст

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 файла	<code>localizations/*.xliff</code>
Основний файл для одномовного перекладу	<i>Порожньо</i>
Заготовка 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>
Заготовка 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](#)

1.9.6 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 вашої програми.

Типовий <i>Налаштовування складників</i> Weblate	
Маска файла	<code>src/app/Bundle_*.properties</code>
Основний файл для одномовного перекладу	<code>src/app/Bundle.properties</code>
Заготовка 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](#), [Форматує файл властивостей Java](#), [Очищення файлів перекладу](#)

1.9.7 Властивості GWT

Власний формат GWT для перекладів.

Властивості GWT, зазвичай, використовуються як одномовні переклади.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файла	src/app/Bundle_*.properties
Основний файл для одномовного перекладу	src/app/Bundle.properties
Заготовка для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>властивості GWT</i>

Дивись також:

Підручник з локалізації GWT Mozilla and Java properties files, *Форматує файл властивостей Java*, *Очищення файлів перекладу*

1.9.8 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 файла	language/*.ini
Основний файл для одномовного перекладу	language/en.ini
Заготовка для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл INI</i>

Дивись також:

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

1.9.9 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 заміників %n і %t для розриву рядка і табуляції.

Типовий <i>Налаштовування складників Weblate</i>	
Маска файла	language/*.isl
Основний файл для одномовного перекладу	language/en.isl
Заготовка 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](#)

1.9.10 Переклади Joomla

Нове в версії 2.12.

Власний формат Joomla для перекладів.

Переклади Joomla, зазвичай, використовуються як одномовні переклади.

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

Дивись також:

Специфікація мовних файлів Joomla, Mozilla and Java properties files, [Переклади INI](#), [Переклади INI Inno Setup](#)

1.9.11 Qt Linguist .ts

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

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

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

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

Дивись також:

Підручник з Qt Linguist, Qt .ts, [Двомовні і одномовні формати](#)

1.9.12 Рядкові ресурси Android

Специфічний для Android формат файлів для перекладу програм.

Рядкові ресурси Android є одномовними. Файл [Основний файл для одномовного перекладу](#) зберігається у іншому місці, якщо порівнювати із іншими res/values/strings.xml.

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

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 бути збережено у 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>

1.9.13 Рядки Apple iOS

Специфічний формат файлів Apple для програм для перекладу. Використовується у перекладах програм iOS і iPhone/iPad.

Рядки Apple iOS, зазвичай, використовуються як двомовні переклади.

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

Дивись також:

Документація із «рядкових файлів» Apple, Mac OSX strings

1.9.14 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 <i>Налаштовування складників Weblate</i>	
Маска файла	lang/*/texts.php
Основний файл для одномовного перекладу	lang/en/texts.php
Заготовка 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

1.9.15 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*
- *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. Єдиною відмінністю є обробка додавання нових рядків. У варіанті із вкладеністю програма робить спробу обробити ключ і вставити новий рядок у відповідну структуру.

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

Дивись також:

[JSON](#), [Налагодити виведення JSON](#), [Очищення файлів перекладу](#),

1.9.16 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.

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 версії 3 JSON i18next. Версії 2 і 1 є здебільшого сумісними, 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 файла	langs/*.json
Основний файл для одномовного перекладу	langs/en.json
Заготовка для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл i18next JSON</i>

Дивись також:

[JSON](#), [формат JSON i18next](#), [Налагодити виведення JSON](#), [Очищення файлів перекладу](#)

1.9.17 Файли JSON go-i18n

Нове в версії 4.1.

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

Примітка: У Weblate передбачено підтримку формату JSON go-i18n версії 1. Для простих форматів JSON, будь ласка, скористайтеся *файли JSON*. Підтримки формату версії 2 із хешем у поточній версії не передбачено.

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

Дивись також:

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

1.9.18 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 файла	lib/l10n/intl_*.arb
Основний файл для одномовного перекладу	lib/l10n/intl_en.arb
Заготовка для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл ARB</i>

Дивись також:

[JSON](#), [Специфікація пакунків ресурсів програм](#), [Інтернаціоналізація програм Flutter](#), [Налагодити виведення JSON](#), [Очищення файлів перекладу](#)

1.9.19 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": "",
    "description": "Description"
  },
  "try": {
    "message": "",
    "description": "Description"
  },
  "thanks": {
    "message": "",
    "description": "Description"
  }
}
```

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

Дивись також:

[JSON](#), [Google chrome.i18n](#), [Переклад розширень Mozilla](#)

1.9.20 Файли ресурсів .XML

Нове в версії 2.3.

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

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

Дивись також:

[.NET Resource files \(.resx\)](#), [Очищення файлів перекладу](#),

1.9.21 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 (початковий рядок, переклад, розташування...). Цей підхід є рекомендованим, оскільки зменшує ймовірність помилки.
- Файли із двома полями — початковим рядком і перекладом (саме у такому порядку) — виберіть *Простий файл CSV* у полі формату файлів
- Файли із полями, які визначено у translate-toolkit: location, source, target, ID, fuzzy, context, translator_comments, developer_comments
- Не забудьте визначити *Основний файл для одномовного перекладу* коли ваші файли одномовні (гляньте *Двомовні і одномовні формати*)

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.

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

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

Дивись також:

CSV

1.9.22 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 Weblate</i>	
Маска файла	translations/messages.*.yaml
Основний файл для одномовного перекладу	translations/messages.en.yaml
Заготовка для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл YAML</i>

Дивись також:

[YAML](#), [Файли YAML Ruby](#)

1.9.23 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 Weblate</i>	
Маска файла	translations/messages.*.yaml
Основний файл для одномовного перекладу	translations/messages.en.yaml
Заготовка для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>файл YAML Ruby</i>

Дивись також:

[YAML](#), [Файли YAML](#)

1.9.24 Файли DTD

Нове в версії 2.18.

Приклад файла DTD:

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

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

Дивись також:

[Mozilla DTD format](#)

1.9.25 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 <i>Налаштовування складників</i> Weblate	
Маска файла	locale/*.xml
Основний файл для одномовного перекладу	locale/en.xml
Заготовка для нових перекладів	<i>Порожньо</i>
Формат файлу	<i>простий файл XML</i>

Дивись також:

Flat XML

1.9.26 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 <i>Налаштовування складників</i> Weblate	
Маска файла	lang/*.rc
Основний файл для одномовного перекладу	lang/en-US.rc
Заготовка для нових перекладів	lang/en-US.rc
Формат файлу	<i>Файл RC</i>

Дивись також:

Windows RC files

1.9.27 Файли метаданих крамниці застосунків

Нове в версії 3.5.

Метадані, які використовуються для оприлюднення програму різноманітних крамницях програмного забезпечення, 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 файла	fastlane/android/metadata/*
Основний файл для одномовного перекладу	fastlane/android/metadata/en-US
Заготовка для нових перекладів	fastlane/android/metadata/en-US
Формат файлу	„Файли метаданих крамниці програм“

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*). Процесу можна автоматизувати за допомогою *Масове редагування*.

1.9.28 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 файла	path/*.srt
Основний файл для одномовного перекладу	path/en.srt
Заготовка для нових перекладів	path/en.srt
Формат файлу	<i>файл субтитрів SubRip</i>

Дивись також:

[Subtitles](#)

1.9.29 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.

1.9.30 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](#)

1.9.31 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](#)

1.9.32 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.

1.9.33 Інше

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 сховищ даних translate-toolkit.

Дивись також:

[Translation Related File Formats](#)

1.9.34 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](#).

1.10 Налаштування інтеграції із керуванням версіями

У поточній версії Weblate передбачено підтримку модулів керування версіями *Git* (із розширеною підтримкою *GitHub*, *Gerrit* і *Subversion*) та *Mercurial*.

1.10.1 Доступ до сховищ

Сховище системи керування версіями, яким ви хочете скористатися, має бути доступним для Weblate. Якщо сховище є загальнодоступним, вам достатньо ввести правильну адресу (наприклад `https://github.com/WeblateOrg/weblate.git`), але для приватних сховищ або адрес для запису налаштування є складнішим і потребує розпізнавання користувача.

Доступ до сховищ з Hosted Weblate

Для Hosted Weblate на GitHub, Bitbucket, Codeberg і GitLab зареєстровано особливого користувача для запису внесків (назва облікового запису — *weblate*, ім'я користувача *Weblate push user*). Вам слід додати цього користувача як учасника розробки і надати йому відповідні права доступу до вашого сховища (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 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 [сховище початкового коду](#) і [адресу запису до сховища](#) з використання протоколу SSH (наприклад, `git@github.com:WeblateOrg/weblate.git`).

Сховища із доступом за SSH

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

Попередження: На GitHub кожен ключ може бути додано лише до одного сховища, див. *Сховища GitHub* і *Доступ до сховищ 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:

The screenshot shows the Weblate web interface for managing SSH keys. The top navigation bar includes 'Weblate', 'Dashboard', 'Projects', 'Languages', 'Checks', and a '+ Add' button. Below this, a breadcrumb trail shows 'Manage / SSH keys'. The main content area has a sidebar with links: 'Weblate status', 'Backups', 'Translation memory', 'Performance report', 'SSH keys' (highlighted), 'Alerts', 'Repositories', 'Users', and 'Appearance'. Below the sidebar, there are two sections: 'Public SSH key' and 'Known host keys'. The 'Public SSH key' section shows the current SSH key used by Weblate, a text area with the key, and a 'Download private key' button. The 'Known host keys' section is a table with columns 'Hostname', 'Key type', and 'Fingerprint', showing a single entry for 'github.com'. Below this is an 'Add host key' section with a form to enter a 'Hostname' and 'Port', and a 'Submit' button. At the bottom, a footer bar contains links: 'Powered by Weblate 4.4', 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Hostname	Key type	Fingerprint
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6ESSY8

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 запам'ятовує ключів SSH вузла при першому доступі і використовує їх у подальшому.

Якщо вам захочеться перевірити їх до встановлення з'єднання зі сховищем, перевірте ключі SSH вузлів серверів, до яких ви збираєтесь 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 відповідає доданому вами серверу. Дані буде показано у повідомленні підтвердження:

The screenshot shows the Weblate web interface for managing SSH keys. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, a breadcrumb trail shows 'Manage / SSH keys'. A yellow notification box states: 'Added host key for github.com with fingerprint nThbg6kXUpJWGi7E1IGOCspRomTxdCARLviKw6E5SY8 (ssh-rsa), please verify that it is correct.' Below the notification, there's a horizontal menu with 'Weblate status', 'Backups', 'Translation memory', 'Performance report', 'SSH keys' (selected), 'Alerts', 'Repositories', 'Users', and 'Appearance'. Under 'SSH keys', there are links for 'Tools' and 'Billing'.

The main content area is divided into three sections:

- Public SSH key:** A section titled 'Public SSH key' with an information icon. It states 'Weblate currently uses this SSH key:' and shows a text box containing 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDNDVZp127qFR6+GH/NrkrFuVxGntScCH0C+yAC36qXnZ85y+h1Z6j26UeALnjHI63Bu4xQz1XCZUpGsGNwKKdUMmq3WmikXZg'. Below the text box is a button labeled 'Download private key'.
- Known host keys:** A section titled 'Known host keys' with an information icon. It contains a table with columns 'Hostname', 'Key type', and 'Fingerprint'. The table has one row: 'github.com', 'ssh-rsa', and 'nThbg6kXUpJWGi7E1IGOCspRomTxdCARLviKw6E5SY8'.
- Add host key:** A section titled 'Add host key' with an information icon. It contains a message: '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.' Below the message is a form with two input fields: 'Hostname' (containing 'github.com') and 'Port' (empty). Below the form is a button labeled 'Submit'.

Powered by Weblate 4.4 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Сховища GitHub

Можливий доступ з використанням SSH (див. *Сховища із доступом за SSH*), але якщо вам потрібен доступ до декількох сховищ, ви зіткнетесь із обмеженням GitHub на використання ключів SSH (оскільки один ключ можна використовувати лише для одного сховища).

Якщо *гілку запису* не встановлено, буде створено відгалуження проекту, і зміни буде записано до відгалуження. Якщо гілку запису встановлено, зміни буде записано до основного сховища 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 сховищ, які ви хочете перекладати. Цей підхід використано і на Hosted Weblate — маємо особливого користувача *weblate*.

Дивись також:

Доступ до сховищ з Hosted Weblate

Внутрішні адреси Weblate

Щоб скористатися одним спільним сховищем для різних складників ви можете використати особливу адресу, `weblate://проект/складник`. Тоді, складник матиме спільні налаштування сховища системи керування версіями із еталонним складником (проект/складник у нашому випадку).

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

Причини для використання внутрішніх адрес:

- Заощаджує місце на диску сервера — сховище зберігається лише один раз.
- Пришвидшує оновлення — оновлюється лише одне сховище.
- Наявність лише одного експортованого сховища із перекладами Weblate (див. *Засіб експортування Git*).
- Деякі додатки можуть працювати із більшою кількістю складників, які мають єдине спільне сховище, наприклад *Сполучити Git подання*.

Сховища HTTPS

Щоб отримати доступ до захищених сховищ 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 сховища).

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 сховищ 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: Налаштування 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

1.10.2 Git

Дивись також:

Див. розділ *Доступ до сховищ*, щоб дізнатися більше про доступ до сховищ 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 сховища для перекладів.

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

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 сховищах 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 сховища у Weblate.

Клонування проєкту `gnuhello` з Launchpad за допомогою Bazaar:

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

Для сховища `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 сховища.

1.10.3 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 сховища.

Git записує зміни безпосередньо до сховища, а *GitHub* створює запити щодо об'єднання. Останній не потрібен для простого доступу до сховищ Git.

Дивись також:

Записування змін з Weblate

Записування змін до GitHub у формі запитів щодо об'єднання

Якщо ви не хочете записувати переклади до сховища GitHub, їх може бути надіслано до одного або багатьох запитів щодо об'єднання.

Щоб мати змогу цим скористатися, вам слід налаштувати реєстраційні дані до програмного інтерфейсу.

Дивись також:

GITHUB_USERNAME, GITHUB_TOKEN, GITHUB_CREDENTIALS

1.10.4 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 сховища.

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

Дивись також:

Записування змін з Weblate

Записування змін до GitLab у формі запитів щодо злиття

Якщо ви не хочете записувати переклади до сховища GitLab, їх може бути надіслано до одного або багатьох запитів щодо об'єднання.

Щоб мати змогу цим скористатися, вам слід налаштувати реєстраційні дані до програмного інтерфейсу.

Дивись також:

GITLAB_USERNAME, GITLAB_TOKEN, GITLAB_CREDENTIALS

1.10.5 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 сховища.

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

Дивись також:

Записування змін з Weblate

Записування змін до Pagine у формі запитів щодо злиття

Якщо ви не хочете записувати переклади до сховища Pagine, їх може бути надіслано до одного або багатьох запитів щодо об'єднання.

Щоб мати змогу цим скористатися, вам слід налаштувати реєстраційні дані до програмного інтерфейсу.

Дивись також:

`PAGURE_USERNAME`, `PAGURE_TOKEN`, `PAGURE_CREDENTIALS`

1.10.6 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 сховища.

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

1.10.7 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 сховищ*, щоб дізнатися більше про доступ до сховищ різних типів.

1.10.8 Subversion

Нове в версії 2.8.

Weblate використовує для взаємодії зі сховищами *subversion* *git-svn*. Це скрипт мовою Perl, який надає змогу користуватися *subversion* за допомогою клієнта *Git*, уможливлюючи для користувачів супровід повного клону внутрішнього сховища і локальні внески.

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

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

Реєстраційні дані Subversion

Weblate очікує, що вами було прийнято сертифікат і, якщо потрібно, вказано ваші реєстраційні дані. Weblate вставить їх до каталогу DATA_DIR. Прийміть сертифікат за допомогою одноразового запуску `svn` із змінною середовища `$HOME`, яка збігається за значенням із 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

1.10.9 Локальні файли

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 цю можливість реалізовано лише для одномовних перекладів).

За лаштунками, Weblate створює сховище 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 сховища у Weblate, на основі якого ви зможете побудувати інтеграцію.

1.11 Програмний інтерфейс REST Weblate

Нове в версії 2.6: Програмний інтерфейс REST є доступним з версії Weblate 2.6.

Доступ до програмного інтерфейсу можна отримати за допомогою адреси `/api/`. Засіб доступу засновано на бібліотеці REST Django. Ви можете скористатися ним безпосередньо або за допомогою *Клієнт Weblate*.

1.11.1 Розпізнавання і типові параметри

Відкритий програмний інтерфейс проєкту є доступним без розпізнавання користувача, хоча частоту запитів користувачів, які не пройшли розпізнавання, значно зменшено (типово, до 100 запитів на добу). Через це рекомендуємо вам скористатися розпізнаванням. У розпізнаванні використовується ключ, який ви можете отримати у вашому профілі. Скористайтеся ним у 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`. Остання надає інтерфейсу браузера для програмного інтерфейсу.

Заголовки запиту

- `Accept` – тип вмісту відповіді залежить від заголовка `Accept`
- `Authorization` – необов'язковий ключ для розпізнавання

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 помилки (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
- 400 Bad Request – якщо не вказано параметри форми
- 403 Forbidden – якщо заборонено доступ
- 429 Too Many Requests – якщо увімкнено обмеження швидкості

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.

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

1.11.2 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/"
}
```

1.11.3 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 суперкористувачем? (необов'язковий)
- **is_active** (*boolean*) – Чи є користувач активним? (необов'язковий)

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 суперкористувачем
- **is_active** (*boolean*) – чи є користувач активним
- **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,
  "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 суперкористувачем
- **is_active** (*boolean*) – чи є користувач активним
- **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 суперкористувачем
- **is_active** (*boolean*) – чи є користувач активним

- **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

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

1.11.4 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 проєкту за вказаними параметрами
- **language_selection** (*int*) – Група мов, яку вибрано за вказаними параметрами

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 проєктів
- **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 проєкти; див. `GET /api/projects/(string:project)/`
- **components** (*array*) – посилання на пов'язані складники; див. `GET /api/components/(string:project)/(string:component)/`
- **componentlist** (*array*) – посилання на пов'язаний список складників; див. `GET /api/component-lists/(str:slug)/`

Приклад даних JSON:

```
{
  "name": "Guests",
  "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 компонент 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 проект 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

1.11.5 Ролі

GET /api/roles/

Повертає список усіх ролей, які пов'язано із користувачем. Якщо користувач є суперкористувачем, 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

1.11.6 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 писемності мови
- **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

Приклад даних JSON:

```
{
  "code": "en",
  "direction": "ltr",
  "name": "English",
  "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 писемності мови
- **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 писемності мови
- **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 рядків
- **fuzzy_percent** (*int*) – частка неточно перекладених рядків
- **failing** (*int*) – кількість рядків, які не пройшли перевірки
- **failing** – відсоток рядків, які не пройшли перевірки

1.11.7 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 проєкту
- **components_list_url** (*string*) – адреса списку складників, див. *GET /api/projects/(string:project)/components/*
- **repository_url** (*string*) – адреса стану сховища, 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*) – Встановлення заголовка «Language-Team»
- **enable_hooks** (*boolean*) – Дозволити гачки
- **instructions** (*string*) – Вказівки щодо перекладу
- **mail** (*string*) – Список листування
- **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 проєкт розміщенням запиту щодо латання.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту
- **component** (*string*) – Ідентифікатор адрес складника

PUT /api/projects/ (*string: project*) /

Нове в версії 4.3.

Редагувати проєкт розміщенням запиту щодо розташування.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту

DELETE /api/projects/ (*string: project*) /

Нове в версії 3.9.

Вилучає проєкт.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту

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 проєкту

Об'єкт відповіді JSON

- **results** (*array*) – список об'єктів складника, див. *GET* /api/changes/ (*int:id*) /

GET /api/projects/ (*string: project*) /repository/

Повертає відомості щодо стану сховища системи керування версіями. Ця кінцева точка містить лише загальне резюме для усіх сховищ проєкту. Щоб отримати докладні дані щодо стану скористайтеся *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 сховищі 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 сховищем 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

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 компонентів системи керування версіями, див. *Локальні файли*.

Створює складники перекладу у вказаному проєкті.

Підказка: 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

Надіслати запит щодо об'єкта JSON

- **zipfile** (*file*) – Файл ZIP, який слід вивантажити до Weblate для ініціалізації перекладів
- **docfile** (*file*) – Документ для перекладу

Об'єкт відповіді JSON

- **result** (*object*) – Створений об'єкт складника; див. `GET /api/components/(string:project)/(string:component)/`

Приклад для CURL:

```
curl \
  --data-binary '{
    "branch": "master",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "slug": "weblate",
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
    "new_base": "",
    "vcs": "git"
  }' \
  -H "Content-Type: application/json" \
  -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

{
  "branch": "master",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "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": "master",
  "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",
      "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/"
}

```

(continues on next page)

(continued from previous page)

}

GET `/api/projects/(string: project)/languages/`

Повертає поділену на сторінки статистику для усіх мов у проєкті.

Нове в версії 3.8.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту

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 проєкту.

Нове в версії 3.8.

Параметри

- **project** (*string*) – Ідентифікатор адреси проєкту

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

1.11.8 Складники

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 проєкту

- **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*) – Сховище з джерелами
- **git_export** (*string*) – Експортований URL сховища
- **branch** (*string*) – Гілка сховища
- **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*) – Заготовка 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 сховища
- **check_flags** (*string*) – Прапорці перекладу
- **priority** (*string*) – Пріоритет
- **enforced_checks** (*string*) – Примусові перевірки
- **restricted** (*string*) – Обмежений доступ
- **repoweb** (*string*) – Переглядач сховища
- **report_source_bugs** (*string*) – Адреса для повідомлень про помилки у рядках
- **merge_style** (*string*) – Стиль злиття
- **commit_message** (*string*) – Повідомлення щодо внеску, додавання, вилучення, об'єднання та повідомлення додатків
- **add_message** (*string*) – Повідомлення щодо внеску, додавання, вилучення, об'єднання та повідомлення додатків
- **delete_message** (*string*) – Повідомлення щодо внеску, додавання, вилучення, об'єднання та повідомлення додатків
- **merge_message** (*string*) – Повідомлення щодо внеску, додавання, вилучення, об'єднання та повідомлення додатків
- **addon_message** (*string*) – Повідомлення щодо внеску, додавання, вилучення, об'єднання та повідомлення додатків

- **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 сховища, 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": "master",
  "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",
      "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",
    "name": "English",
    "url": "http://example.com/api/languages/en/",
    "web_url": "http://example.com/languages/en/"
  },
}
```

(continues on next page)

(continued from previous page)

```

"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 щодо латання.

Параметри

- **project** (string) – Ідентифікатор адреси проекту
- **component** (string) – Ідентифікатор адрес складника
- **source_language** (string) – Код початкової мови проекту (необов'язковий)

Надіслати запит щодо об'єкта JSON

- **name** (string) – назва складника
- **slug** (string) – ідентифікатор складника
- **repo** (string) – адреса сховища 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

{
  "branch": "master",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",

```

(continues on next page)

(continued from previous page)

```

"license": "",
"license_url": "",
"name": "new name",
"slug": "weblate",
"project": {
  "name": "Hello",
  "slug": "hello",
  "source_language": {
    "code": "en",
    "direction": "ltr",
    "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 щодо розташування.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **component** (*string*) – Ідентифікатор адрес складника

Надіслати запит щодо об'єкта JSON

- **branch** (*string*) – гілка сховища системи керування версіями
- **file_format** (*string*) – формат файлів перекладів
- **filemask** (*string*) – маска файлів перекладів у сховищі
- **name** (*string*) – назва складника
- **slug** (*string*) – ідентифікатор складника
- **repo** (*string*) – адреса сховища 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/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/` Встановлює 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 сховище 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 сховища 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 сховищі для злиття
- **needs_push** (*boolean*) – визначає, чи є якісь локальні зміни для запису
- **remote_commit** (*string*) – Дані віддаленого внеску
- **status** (*string*) – Стан сховища 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 сховищем системи керування версіями.

Див. документацію у `POST /api/projects/(string:project)/repository/`.

Параметри

- **project** (*string*) – Ідентифікатор адреси проекту
- **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

{"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",
    "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/",
```

(continues on next page)

(continued from previous page)

```

"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 усіх перекладів у складнику.

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/`

1.11.9 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 проекту
- **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 файлу перекладу
- **fuzzy** (*int*) – кількість рядків, які позначено для рецензування
- **fuzzy_percent** (*float*) – частка рядків, які позначено для рецензування

- **fuzzy_words** (*int*) – кількість слів, які позначено для рецензування
- **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 сховищі; 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 сховища, 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": "master",
    "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",
        "name": "English",

```

(continues on next page)

(continued from previous page)

```

        "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",
    "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,
"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 одномовний 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
- **value** (*string*) – Значення модуля перекладу

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
- **component** (*string*) – Щоб отримати доступ до додаткових складників, увімкніть спільний внесок у пам'ять перекладів для проекту.
- **engines** (*string*) – Рушії комп'ютерного перекладу
- **threshold** (*string*) – Поріг оцінки

GET `/api/translations/(string: project) /`
string: `component/string: language/file/` Отримати поточний файл перекладу у форматі, у якому він зберігається у системі керування версіями, (без параметра `format`) або у стандартному форматі (у поточній версії підтримуваними форматами є: Gettext PO, MO, XLIFF id TBX).

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`, `csv`, `xlsx`, `json`, `aresource`, `strings`

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`), див. [Способи імпортування](#)
- **string fuzzy** – Обробка неточних 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 сховища 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 сховищем 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 рядків, які потребують рецензування
- **fuzzy_percent** (*float*) – частка рядків, які потребують рецензування
- **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)

1.11.10 Модулі

Нове в версії 2.10.

GET `/api/units/`

Повертає список модулів перекладу.

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
- **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](#)

- **web_url** (*string*) – адреса, за якою можна редагувати модуль
- **source_unit** (*string*) – Посилання на модуль початкового коду; див. [GET /api/units/\(int:id\)/](#)

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 із 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](#)

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 із 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](#)

DELETE /api/units/(int: id) /

Нове в версії 4.3.

Вилучає модуль перекладу.

Параметри

- **id** (*int*) – Ід. модуля

1.11.11 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
- **glossary_term** (*string*) – адреса пов'язаного об'єкта терміна з глосарія
- **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

1.11.12 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 проекту
- **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

1.11.13 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

1.11.14 Глосарій

GET `/api/glossary/`

Повертає список усіх пов'язаних із проектом глосаріїв, доступ до яких має користувач.

Дивись також:

Атрибути об'єкта ролі описано у `GET /api/languages/ (string: language) /`.

GET `/api/glossary/ (int: id) /`

Повертає подробиці про глосарій.

Параметри

- **id** (*int*) – Ідентифікатор глосарія

Об'єкт відповіді JSON

- **name** (*string*) – Код мови
- **color** (*string*) – Напрямок тексту
- **source_language** (*object*) – Об'єкт даних щодо форм множини мови
- **projects** (*array*) – посилання на пов'язані проекти; див. `GET /api/projects/ (string:project) /`

Приклад даних JSON:

```
{
  "name": "Hello",
  "id": 1,
  "color": "silver",
  "source_language": {
    "code": "en",
    "name": "English",
    "plural": {
      "id": 75,
      "source": 0,
      "number": 2,
      "formula": "n != 1",
      "type": 1
    },
  },
  "aliases": [
    "english",
    "en_en",
    "base",
    "source",
  ]
}
```

(continues on next page)

(continued from previous page)

```

        "eng"
    ],
    "direction": "ltr",
    "web_url": "http://example.com/languages/en/",
    "url": "http://example.com/api/languages/en/",
    "statistics_url": "http://example.com/api/languages/en/statistics/"
},
"project": {
    "name": "Hello",
    "slug": "hello",
    "id": 1,
    "source_language": {
        "code": "en",
        "name": "English",
        "plural": {
            "id": 75,
            "source": 0,
            "number": 2,
            "formula": "n != 1",
            "type": 1
        },
        "aliases": [
            "english",
            "en_en",
            "base",
            "source",
            "eng"
        ]
    },
    "direction": "ltr",
    "web_url": "http://example.com/languages/en/",
    "url": "http://example.com/api/languages/en/",
    "statistics_url": "http://example.com/api/languages/en/statistics/"
},
"web_url": "http://example.com/projects/demo1/",
"url": "http://example.com/api/projects/demo1/",
"components_list_url": "http://example.com/api/projects/demo1/
↪components/",
"repository_url": "http://example.com/api/projects/demo1/repository/",
"statistics_url": "http://example.com/api/projects/demo1/statistics/",
"changes_list_url": "http://example.com/api/projects/demo1/changes/",
"languages_url": "http://example.com/api/projects/demo1/languages/"
},
"projects_url": "http://example.com/api/glossary/7/projects/",
"terms_url": "http://example.com/api/glossary/7/terms/",
"url": "http://example.com/api/glossary/7/"
}

```

PUT `/api/glossary/(int: id)/`
Змінює параметри глосарія.

Параметри

- **id** (*int*) – Ідентифікатор глосарія

Надіслати запит щодо об'єкта JSON

- **name** (*string*) – Назва мови
- **color** (*string*) – Напрямок писемності мови
- **source_language** (*object*) – Подробиці щодо форм множини мови

PATCH `/api/glossary/(int: id)/`
Змінює параметри глосарія.

Параметри

- **id** (*int*) – Ідентифікатор глосарія

Надіслати запит щодо об'єкта JSON

- **name** (*string*) – Назва мови
- **color** (*string*) – Напрямок писемності мови
- **source_language** (*object*) – Подобиці щодо форм множини мови

DELETE /api/glossary/ (int: id) /

Вилучає глосарій.

Параметри

- **id** (*int*) – Ідентифікатор глосарія

GET /api/glossary/ (int: id) /projects/

Повертає проєкт, який пов'язано з глосарієм.

Параметри

- **id** (*int*) – Ідентифікатор глосарія

Об'єкт відповіді JSON

- **projects** (*array*) – пов'язані проєкти; див. *GET* /api/projects/ (*string:project*) /

POST /api/glossary/ (int: id) /projects/

Пов'язати проєкт із глосарієм.

Параметри

- **id** (*int*) – Ідентифікатор глосарія

Параметри форми

- **string project_slug** – Ідентифікатор проєкту

DELETE /api/glossary/ (int: id) /projects/

Вилучити прив'язку проєкту до глосарія.

Параметри

- **id** (*int*) – Ідентифікатор глосарія

Параметри форми

- **string project_slug** – Ідентифікатор проєкту

GET /api/glossary/ (int: id) /terms/

Вивести список термінів глосарія.

Параметри

- **id** (*int*) – Ідентифікатор глосарія

POST /api/glossary/ (int: id) /terms/

Пов'язати терміни з глосарієм.

Параметри

- **id** (*int*) – Ідентифікатор глосарія

Надіслати запит щодо об'єкта JSON

- **language** (*object*) – Мова терміна
- **source** (*string*) – Початковий рядок терміна
- **target** (*string*) – Рядок перекладу терміна

GET `/api/glossary/(int: id)/terms/`
int: `term_id`/ Отримати термін, пов'язаний з глосарієм.

Параметри

- **id** (*int*) – Ідентифікатор глосарія
- **term_id** (*int*) – Ідентифікатор терміна

PUT `/api/glossary/(int: id)/terms/`
int: `term_id`/ Змінити термін, який пов'язано із глосарієм.

Параметри

- **id** (*int*) – Ідентифікатор глосарія
- **term_id** (*int*) – Ідентифікатор терміна

Надіслати запит щодо об'єкта JSON

- **language** (*object*) – Мова терміна
- **source** (*string*) – Початковий рядок терміна
- **target** (*string*) – Рядок перекладу терміна

PATCH `/api/glossary/(int: id)/terms/`
int: `term_id`/ Змінити термін, який пов'язано із глосарієм.

Параметри

- **id** (*int*) – Ідентифікатор глосарія
- **term_id** (*int*) – Ідентифікатор терміна

Надіслати запит щодо об'єкта JSON

- **language** (*object*) – Мова терміна
- **source** (*string*) – Початковий рядок терміна
- **target** (*string*) – Рядок перекладу терміна

DELETE `/api/glossary/(int: id)/terms/`
int: `term_id`/ Вилучити термін, який пов'язано з глосарієм.

Параметри

- **id** (*int*) – Ідентифікатор глосарія
- **term_id** (*int*) – Ідентифікатор терміна

1.11.15 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

1.11.16 Обробники сповіщень

Обробники сповіщень надають змогу стороннім програмам сповіщати Weblate про оновлення у сховищі коду системи керування версіями.

Ви можете скористатися кінцевими точками для проєктів, складників та перекладів у сховищі для оновлення окремих сховищ. 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 компонентів.

Примітка: До GitHub включено безпосередню підтримку сповіщень Weblate: увімкніть скрипт перехоплення для служби Weblate у параметрах сховища 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/free-pro-team@latest/github/extending-github/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 компонентів.

Дивись також:

Автоматичне отримання змін з GitLab Настанови щодо налаштування інтеграції із GitLab

<https://docs.gitlab.com/ce/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 компонентів.

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 веб-скриптів 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 компонентів.

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 Repos і автоматичного оновлення відповідних компонентів.

Дивись також:

Автоматичне отримання змін зі сховищ Azure Настанови щодо налаштування інтеграції із Azure

<https://docs.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops> Загальні відомості щодо вебскриптів сховищ Azure

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 компонентів.

Дивись також:

Автоматичне отримання змін зі сховищ 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 компонентів.

Дивись також:

Автоматичне отримання змін зі сховищ 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

1.11.17 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/master/ 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/master/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/",
```

(continues on next page)

(continued from previous page)

```

    "url_translate": "http://hosted.weblate.org/projects/weblate/master/nl/
  },
  {
    "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/master/el/
  }
]

```

1.11.18 RSS-стрічки

Зміни у перекладах експортуються до стрічок RSS.

GET `/exports/rss/(string: project) /`

string: *component* / **string:** *language* / Отримує стрічку RSS із останніми змінами у перекладі.

GET `/exports/rss/(string: project) /`

string: *component* / Отримує стрічку RSS із останніми змінами у складнику.

GET `/exports/rss/(string: project) /`

Отримує стрічку RSS із останніми змінами у проєкті.

GET `/exports/rss/language/(string: language) /`

Отримує стрічку RSS із останніми змінами у мові.

GET `/exports/rss/`

Отримує стрічку RSS із останніми змінами у екземплярі Weblate.

Дивись також:

[RSS у Вікіпедії](#)

1.12 Клієнт Weblate

Нове в версії 2.7: Повну підтримку допоміжного засобу `wlc` реалізовано з версії Weblate 2.7. Якщо ви користуєтеся давнішою версією, ви можете зіткнутися з певними несумісностями у програмному інтерфейсі.

1.12.1 Встановлення

Клієнт Weblate постачається окремо і включає до себе модуль Python. Щоб користуватися наведеними нижче командами, вам слід встановити `wlc`:

```
pip3 install wlc
```

1.12.2 Користування Docker

Крім того, клієнт Weblate доступний у форматі образу для Docker.

Образ оприлюднено на Docker Hub: <https://hub.docker.com/r/weblate/wlc>

Встановлення:

```
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
```

1.12.3 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

1.12.4 Короткий опис

```
wlc [arguments] <command> [options]
```

Команди вказують на те, яку дію має бути виконано.

1.12.5 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 сховищі до об'єкта Weblate (перекладу, складника або проєкту).

push

Записує зміни у об'єкті Weblate до віддаленого сховища (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 сховищі (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 сховищі (переклад, складник або проєкт).

repo

Показує стан у сховищі для вказаного об'єкта Weblate (перекладу, складника або проєкту).

statistics

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 сховищі.

--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.

Підказка: Докладніші відомості щодо виклику окремих команд можна отримати за допомогою параметра `--help`. Приклад: `wlc ls --help`.

1.12.6 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 `XDG_CONFIG_HOME` і `XDG_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/master
```

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 сховищі системи керування версіями так, щоб `wlc` було відомо, з яким сервером слід працювати.

1.12.7 Приклади

Вивести версію поточної програми:

```
$ wlc version
version: 0.1
```

Список усіх проектів:

```
$ 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/
```

Ви також можете позначити проєкт, з яким має працювати wlc:

```
$ cat .weblate
[weblate]
url = https://hosted.weblate.org/api/
translation = weblate/master

$ wlc show
branch: master
file_format: po
source_language: en
filemask: weblate/locale/*/LC_MESSAGES/django.po
git_export: https://hosted.weblate.org/git/weblate/master/
license: GPL-3.0+
license_url: https://spdx.org/licenses/GPL-3.0+
name: master
new_base: weblate/locale/django.pot
project: weblate
repo: git://github.com/WeblateOrg/weblate.git
slug: master
template:
url: https://hosted.weblate.org/api/components/weblate/master/
vcs: git
web_url: https://hosted.weblate.org/projects/weblate/master/
```

З цими налаштуваннями доволі просто надсилати зміни з черги у поточному проєкті:

```
$ wlc commit
```

1.13 Програмний інтерфейс Python Weblate

1.13.1 Встановлення

Програмний інтерфейс Python постачається окремо. Вам слід встановити *Клієнт Weblate*: (wlc), щоб отримати доступ до нього.

```
pip install wlc
```

1.13.2 wlc

WeblateException

exception `wlc.WeblateException`

Основний клас для усіх виключень.

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 типову.
- **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.

1.13.3 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*).

1.13.4 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 Встановлення Weblate

Встановлення 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):

- 2 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 сховищ 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.

Встановлення

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 сховища `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](#)

Контейнер Docker із підтримкою HTTPS

Будь ласка, ознайомтеся із розділом *Встановлення*, 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.

Ви можете досягти цього фіксацією наявного `docker-compose` із простим отриманням найсвіжіших образів і наступним перезапуском:

```
docker-compose stop
docker-compose pull
docker-compose up
```

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 версій за 3.0 у Weblate не передбачено. Якщо ви працюєте із версією 2.x і хочете оновитися до 3.x, спочатку оновіться до найсвіжішої образу версії 3.0.1-x (на час написання цього підручника такою версією була 3.0.1-7), а далі продовжуйте оновлення до новіших версій.

Ймовірно, вам також варто оновити сховище `docker-compose`, хоча, у більшості випадків, потреби у цьому немає. Будь ласка, зверніть увагу на зміни у версії PostgreSQL у цьому випадку, оскільки тут не обійтися простим оновленням бази даних. Докладніше [на GitHub](#), щоб дізнатися більше.

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`

Змінні середовища Docker

Багато з *параметрів налаштування* може бути встановлено у контейнері Docker за допомогою змінних 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_SITE_TITLE

Налаштовує заголовок сайту, який буде показано у верхньому колонтитулі усіх сторінок.

WEBLATE_SITE_DOMAIN

Налаштовує домен сайту.

Підказка: Якщо змінну не було встановлено, буде використано перший запис із `WEBLATE_ALLOWED_HOSTS`.

Дивись також:

Встановіть належний домен сайту, `SITE_DOMAIN`

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Налаштовує ім'я та адресу електронної пошти адміністратора сайту. 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

Встановлює 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_NAME`, `WEBLATE_ADMIN_EMAIL`

WEBLATE_SERVER_EMAIL

WEBLATE_DEFAULT_FROM_EMAIL

Налаштовує адресу для вихідних повідомлень електронної пошти.

Дивись також:

Налаштовування надсилання електронної пошти

WEBLATE_ALLOWED_HOSTS

Налаштовує дозволених 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, Встановіть 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
```

(continues on next page)

(continued from previous page)

WEBLATE_TIME_ZONE

Налаштовує використаний часовий пояс у Weblate, див. [TIME_ZONE](#).

Примітка: Щоб змінити часовий пояс самого контейнера Docker, скористайтеся 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 посиленнях електронної пошти та програмного інтерфейсу та встановлює прапорці захищеності у куках.

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:

Встановіть належний домен сайта

WEBLATE_IP_PROXY_HEADER

Дозволяє Weblate отримувати IP-адресу з будь-якого вказаного заголовка HTTP. Скористайтеся 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

Налаштовує обліковий запис, потрібний для усього встановленого Weblate, за допомогою [LOGIN_REQUIRED_URLS](#).

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 облікового запису, 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

Налаштовує ім'я користувача GitHub для запитів щодо об'єднання GitHub зміною [GITHUB_USERNAME](#).

Дивись також:

[GitHub](#)

WEBLATE_GITHUB_TOKEN

Нове в версії 4.3.

Налаштовує особистий жетон доступу GitHub для запитів щодо об'єднання на GitHub з використанням програмного інтерфейсу шляхом зміни [GITHUB_TOKEN](#).

Дивись також:

[GitHub](#)

WEBLATE_GITLAB_USERNAME

Налаштовує ім'я користувача GitLab для запитів щодо об'єднання GitLab зміною [GITLAB_USERNAME](#)

Дивись також:

[GitLab](#)

WEBLATE_GITLAB_TOKEN

Налаштовує особистий жетон доступу GitLab для запитів щодо об'єднання на GitLab з використанням програмного інтерфейсу шляхом зміни [GITLAB_TOKEN](#)

Дивись також:

[GitLab](#)

WEBLATE_PAGURE_USERNAME

Налаштовує ім'я користувача Pagure для запитів щодо об'єднання Pagure зміною [PAGURE_USERNAME](#)

Дивись також:

[Pagure](#)

WEBLATE_PAGURE_TOKEN

Налаштовує особистий жетон доступу Pagure для запитів щодо об'єднання на Pagure з використанням програмного інтерфейсу шляхом зміни *PAGURE_TOKEN*

Дивись також:

Pagure

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_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 налаштувати 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_HIDE_VERSION

Налаштовує *HIDE_VERSION*.

WEBLATE_BASIC_LANGUAGES

Налаштовує *BASIC_LANGUAGES*.

Параметри комп'ютерного перекладу

WEBLATE_MT_APERTIUM_APY

Вмикає комп'ютерний переклад *Apertium* і встановлює *MT_APERTIUM_APY*

WEBLATE_MT_AWS_REGION

WEBLATE_MT_AWS_ACCESS_KEY_ID

WEBLATE_MT_AWS_SECRET_ACCESS_KEY

Налаштовує комп'ютерний переклад *AWS*.

```
environment:
  WEBLATE_MT_AWS_REGION: us-east-1
  WEBLATE_MT_AWS_ACCESS_KEY_ID: AKIAIOSFODNN7EXAMPLE
  WEBLATE_MT_AWS_SECRET_ACCESS_KEY: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

WEBLATE_MT_DEEPL_KEY

Вмикає комп'ютерний переклад *DeepL* і встановлює *MT_DEEPL_KEY*

WEBLATE_MT_DEEPL_API_VERSION

Налаштовує версію програмного інтерфейсу *DeepL*, якою слід скористатися, див. *MT_DEEPL_API_VERSION*.

WEBLATE_MT_GOOGLE_KEY

Вмикає *Перекладач Google* і встановлює *MT_GOOGLE_KEY*

WEBLATE_MT_MICROSOFT_COGNITIVE_KEY

Вмикає *Когнітивні служби перекладу Microsoft* і встановлює *MT_MICROSOFT_COGNITIVE_KEY*

WEBLATE_MT_MICROSOFT_ENDPOINT_URL

Встановлює *MT_MICROSOFT_ENDPOINT_URL*. Будь ласка, зауважте, що тут має міститися лише назва домену.

WEBLATE_MT_MICROSOFT_REGION

Встановлює *MT_MICROSOFT_REGION*

WEBLATE_MT_MICROSOFT_BASE_URL

Встановлює *MT_MICROSOFT_BASE_URL*

WEBLATE_MT_MODERNMT_KEY

Вмикає *ModernMT* і встановлює *MT_MODERNMT_KEY*.

WEBLATE_MT_MYMEMORY_ENABLED

Вмикає комп'ютерний переклад *MyMemory* і встановлює *MT_MYMEMORY_EMAIL* у значення *WEBLATE_ADMIN_EMAIL*.

Приклад:

```
environment:
  WEBLATE_MT_MYMEMORY_ENABLED: 1
```

WEBLATE_MT_GLOSBE_ENABLED

Вмикає комп'ютерний переклад *Glosbe*.

```
environment:
  WEBLATE_MT_GLOSBE_ENABLED: 1
```

WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED

Вмикає комп'ютерний переклад *Служба термінології Microsoft*.

```
environment:
  WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED: 1
```

WEBLATE_MT_SAP_BASE_URL

WEBLATE_MT_SAP_SANDBOX_APIKEY**WEBLATE_MT_SAP_USERNAME****WEBLATE_MT_SAP_PASSWORD****WEBLATE_MT_SAP_USE_MT**Налаштовує комп'ютерний переклад *Концентратор перекладів SAP*.

```
environment:
  WEBLATE_MT_SAP_BASE_URL: "https://example.hana.ondemand.com/translationhub/
  ↪api/v1/"
  WEBLATE_MT_SAP_USERNAME: "user"
  WEBLATE_MT_SAP_PASSWORD: "password"
  WEBLATE_MT_SAP_USE_MT: 1
```

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_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

Вмикає *Розпізнавання за GitHub*.

Bitbucket

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*.

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`

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_UBUNTU`

Ідентифікатор

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Вмикає розпізнавання за допомогою Slack, див. *Ідентифікатор*.

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

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.

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_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*.

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_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 доступ до електронної пошти.

Дивись також:

`EMAIL_HOST_USER`

WEBLATE_EMAIL_HOST_PASSWORD

Пароль для розпізнавання для доступ до електронної пошти.

Дивись також:

`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`.

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`.

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`

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.

Налаштовування для *CDN локалізації JavaScript*.

WEBLATE_LOCALIZE_CDN_PATH є шляхом у контейнері. Сховище даних має перебувати на постійному томі, а не у тимчасовому сховищі 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: За обслуговування файлів, які буде створено Weblate, відповідає лише ви — програми лише зберігає файли у налаштованому вами місці.

Дивись також:

weblate-cdn, *LOCALIZE_CDN_URL*, *LOCALIZE_CDN_PATH*

Зміна увімкнених програм, перевірок, додатків та автоматичних виправлень

Нове в версії 3.8-5.

Вбудовані 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

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 паралельних обробників збігається із кількістю процесорів. Винятком є обробки резервного копіювання, який запускається лише в одному екземплярі.

Приклад:

```
environment:
  CELERY_MAIN_OPTIONS: --concurrency 16
```

Дивись також:

[Параметри обробника Celery](#), [Фонові завдання з використанням Celery](#)

UWSGI_WORKERS

Налаштувати кількість обробників uWSGI, які має бути виконано.

Типовим значенням є кількість процесорів + 1.

Приклад:

```
environment:
  UWSGI_WORKERS: 32
```

In case you have a lot of CPU cores and hit out of memory issues, try reducing number of workers:

```
environment:
  UWSGI_WORKERS: 4
  CELERY_MAIN_OPTIONS: --concurrency 2
  CELERY_NOTIFY_OPTIONS: --concurrency 1
  CELERY_TRANSLATE_OPTIONS: --concurrency 1
```

Томи контейнера Docker

Існує один том даних, який експортується 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 сховищ, 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/`. У контейнері його змонтовано до `/app/data`.

Дивись також:

[Документація з томів Docker](#)

Подальше налаштування

Ви можете здійснити подальше налаштування встановленого Weblate у томі даних, див. *Томі контейнера Docker*.

Файли нетипових налаштувань

Крім того, ви можете перевизначити налаштування у `/app/data/settings-override.py` (див. *Томі контейнера Docker*). Цей файл виконується після завантаження усіх параметрів середовища, отже, середовище буде повністю налаштовано, і файл може бути використано для додаткового налаштування будь-чого.

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. *Томі 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.

Крім того, ви можете включити власний модуль (див. *Налаштування 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
```

Додавання власних модулів Python

Нове в версії 3.8-5.

Ви можете розташувати власні модулі Python у `/app/data/python/` (див. *Томі контейнера Docker*), і їх потім може бути завантажено Weblate, найімовірніше, за допомогою *Файли нетипових налаштувань*.

Дивись також:

Налаштування Weblate

Виберіть вашу машину — локальні обробники і обробники у «хмарі»

За допомогою машини у Docker ви можете створити вашу розгорнуту систему Weblate або на вашій локальній машині, або на будь-якій із численних систем на основі «хмарних» технологій, наприклад Amazon AWS, Greenhost тощо.

Встановлення 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):

- 2 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 сховищ 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.

Встановлення

Вимоги до системи

Встановіть залежності, потрібні для збирання модулів Python (див. *Вимоги для програмного забезпечення*):

```
apt install \
  libxml2-dev libxslt-dev libfontconfig-dev libjpeg-dev libz-dev libyaml-dev \
  libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev libacl1-dev libssl-dev \
  build-essential python3-gdbm python3-dev python3-pip python3-virtualenv \
  ↪virtualenv git
```

Встановіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (див. *Необов'язкові залежності*):

```
apt install tesseract-ocr libtesseract-dev liblibleptonica-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 nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with `mod_wsgi`
apt install apache2 libapache2-mod-wsgi

# Caching backend: Redis
apt install redis-server
```

(continues on next page)

(continued from previous page)

```
# Database server: PostgreSQL
apt install postgresql postgresql-contrib

# SMTP server
apt install 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 --python=python3 ~/weblate-env
```

2. Активуйте `virtualenv` для Weblate:

```
. ~/weblate-env/bin/activate
```

3. Встановіть Weblate разом із усіма залежностями:

```
pip install Weblate
```

4. Встановіть драйвер бази даних:

```
pip install psycopg2-binary
```

5. Встановіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (деяких з них можуть потребувати додаткових системних бібліотек, див. [Необов'язкові залежності](#)):

```
pip install ruamel.yaml aeidon boto3 zeep chardet tesseract
```

Налаштовування Weblate

Примітка: У наступних кроках ми припускаємо, що віртуальне середовище, яке використовується Weblate, є активним (активувати його можна за допомогою `. ~/weblate-env/bin/activate`). Якщо середовище не активовано, вам доведеться вказати повний шлях до команди **weblate** як `~/weblate-env/bin/weblate`.

1. Скопіюйте файл `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` до `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Скорируйте значення у новому файлі `settings.py` відповідно до ваших уподобань. Ви можете скористатися типовим прикладом для тестування, але вам варто внести зміни для вашої промислової конфігурації, див. [Коригування налаштувань](#).
3. Створіть базу даних і її структуру для Weblate (у прикладі вказано параметри для PostgreSQL, ознайомтеся із розділом [check](#) [Налаштування бази даних для Weblate](#), щоб дізнатися більше про готову до промислової роботи конфігурацію):

```
weblate migrate
```

4. Створіть обліковий запис адміністративного користувача і скопіюйте пароль, який буде виведено, до буфера обміну даними, а також збережіть його для наступного використання:

```
weblate createadmin
```

5. Зберіть статичні файли для 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 рекомендуємо це у всіх інших варіантах. Див. [Фонові завдання з використанням Celery](#), щоб дізнатися більше:

```
~/weblate-env/lib/python3.7/site-packages/weblate/examples/celery start
```

8. Запустіть сервер для розробки (див. [Запуск сервера](#), щоб дізнатися більше про промислові налаштування):

```
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 програми **weblate**, коли віртуальне середовище Weblate є активним, див. [Команди керування](#).
- Припинити роботу тестового сервера можна за допомогою натискання комбінації клавіш Ctrl+C.

Додавання перекладу

1. Відкрийте адміністративний інтефейс (`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 сховище системи керування версіями і визначає, які файли слід перекладати. Див. [Налаштовування складників](#), щоб дізнатися більше.

Важливими полями тут є такі: назва складника, адреса сховища системи керування версіями та маска для пошуку придатних до перекладу файлів. У Weblate передбачено підтримку широкого діапазону форматів, зокрема файлів PO gettext PO, рядків ресурсів Android, рядкових властивостей iOS, властивостей Java та файлів Qt Linguist. Див. [Підтримувані формати файлів](#), щоб дізнатися більше.

3. Щойно буде виконано вказані вище дії (це може бути доволі тривалим процесом, залежно від розмірів вашого сховища системи керування версіями та кількості повідомлень, які слід перекласти), ви можете розпочати переклад.

Встановлення 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):

- 2 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 сховищ 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.

Встановлення

Вимоги до системи

Встановіть залежності, потрібні для збирання модулів Python (див. *Вимоги для програмного забезпечення*):

```
zypper install \
  libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel libyaml-
  <del>devel</del> \
  cairo-devel typelib-1_0-Pango-1_0 gobject-introspection-devel libacl-devel \
  python3-pip python3-virtualenv python3-devel git
```

Встановіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (див. *Необов'язкові залежності*):

```
zypper install tesseract-ocr tesseract-devel leptonica-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
```

(continues on next page)

(continued from previous page)

```
# 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 призвичаєні до такого встановлення, ознайомтеся із [User Guide](#) до `virtualenv`.

1. Створіть `virtualenv` для Weblate:

```
virtualenv --python=python3 ~/weblate-env
```

2. Активуйте `virtualenv` для Weblate:

```
. ~/weblate-env/bin/activate
```

3. Встановіть Weblate разом із усіма залежностями:

```
pip install Weblate
```

4. Встановіть драйвер бази даних:

```
pip install psycopg2-binary
```

5. Встановіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (деяких з них можуть потребувати додаткових системних бібліотек, див. [Необов'язкові залежності](#)):

```
pip install ruamel.yaml aeidon boto3 zeep chardet tesseract
```

Налаштовування Weblate

Примітка: У наступних кроках ми припускаємо, що віртуальне середовище, яке використовується Weblate, є активним (активувати його можна за допомогою `. ~/weblate-env/bin/activate`). Якщо середовище не активовано, вам доведеться вказати повний шлях до команди **weblate** як `~/weblate-env/bin/weblate`.

1. Скопіюйте файл `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` до `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Скорируйте значення у новому файлі `settings.py` відповідно до ваших уподобань. Ви можете скористатися типовим прикладом для тестування, але вам варто внести зміни для вашої промислової конфігурації, див. [Коригування налаштувань](#).
3. Створіть базу даних і її структуру для Weblate (у прикладі вказано параметри для PostgreSQL, ознайомтеся із розділом [check](#) [Налаштування бази даних для Weblate](#), щоб дізнатися більше про готову до промислової роботи конфігурацію):

```
weblate migrate
```

- Створіть обліковий запис адміністративного користувача і скопіюйте пароль, який буде виведено, до буфера обміну даними, а також збережіть його для наступного використання:

```
weblate createadmin
```

- Зберіть статичні файли для 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 рекомендуємо це у всіх інших варіантах. Див. [Фонові завдання з використанням Celery](#), щоб дізнатися більше:

```
~/weblate-env/lib/python3.7/site-packages/weblate/examples/celery start
```

- Запустіть сервер для розробки (див. [Запуск сервера](#), щоб дізнатися більше про промислові налаштування):

```
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 програми **weblate**, коли віртуальне середовище Weblate є активним, див. [Команди керування](#).
- Припинити роботу тестового сервера можна за допомогою натискання комбінації клавіш Ctrl+C.

Додавання перекладу

- Відкрийте адміністративний інтефейс (`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 сховище системи керування версіями і визначає, які файли слід перекладати. Див. [Налаштовування складників](#), щоб дізнатися більше.

Важливими полями тут є такі: назва складника, адреса сховища системи керування версіями та маска для пошуку придатних до перекладу файлів. У Weblate передбачено підтримку широкого діапазону форматів, зокрема файлів PO gettext PO, рядків ресурсів Android, рядкових властивостей iOS, властивостей Java та файлів Qt Linguist. Див. [Підтримувані формати файлів](#), щоб дізнатися більше.

- Щойно буде виконано вказані вище дії (це може бути доволі тривалим процесом, залежно від розмірів вашого сховища системи керування версіями та кількості повідомлень, які слід перекласти), ви можете розпочати переклад.

Встановлення 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):

- 2 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 сховищ 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.

Встановлення

Вимоги до системи

Встановіть 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 \
    cairo-devel pango-devel gobject-introspection-devel libacl-devel \
    python3-pip python3-virtualenv python3-devel git
```

Встановіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (див. *Необов'язкові залежності*):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-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
```

(continues on next page)

(continued from previous page)

```
# 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 призвичаєні до такого встановлення, ознайомтеся із [User Guide](#) до `virtualenv`.

1. Створіть `virtualenv` для Weblate:

```
virtualenv --python=python3 ~/weblate-env
```

2. Активуйте `virtualenv` для Weblate:

```
. ~/weblate-env/bin/activate
```

3. Встановіть Weblate разом із усіма залежностями:

```
pip install Weblate
```

4. Встановіть драйвер бази даних:

```
pip install psycopg2-binary
```

5. Встановіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (деяких з них можуть потребувати додаткових системних бібліотек, див. [Необов'язкові залежності](#)):

```
pip install ruamel.yaml aeidon boto3 zeep chardet tesseract
```

Налаштовування Weblate

Примітка: У наступних кроках ми припускаємо, що віртуальне середовище, яке використовується Weblate, є активним (активувати його можна за допомогою `. ~/weblate-env/bin/activate`). Якщо середовище не активовано, вам доведеться вказати повний шлях до команди **weblate** як `~/weblate-env/bin/weblate`.

1. Скопіюйте файл `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` до `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Скорируйте значення у новому файлі `settings.py` відповідно до ваших уподобань. Ви можете скористатися типовим прикладом для тестування, але вам варто внести зміни для вашої промислової конфігурації, див. [Коригування налаштувань](#).
3. Створіть базу даних і її структуру для Weblate (у прикладі вказано параметри для PostgreSQL, ознайомтеся із розділом [check](#) [Налаштування бази даних для Weblate](#), щоб дізнатися більше про готову до промислової роботи конфігурацію):

```
weblate migrate
```

4. Створіть обліковий запис адміністративного користувача і скопіюйте пароль, який буде виведено, до буфера обміну даними, а також збережіть його для наступного використання:

```
weblate createadmin
```

5. Зберіть статичні файли для 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 рекомендуємо це у всіх інших варіантах. Див. [Фонові завдання з використанням Celery](#), щоб дізнатися більше:

```
~/weblate-env/lib/python3.7/site-packages/weblate/examples/celery start
```

8. Запустіть сервер для розробки (див. [Запуск сервера](#), щоб дізнатися більше про промислові налаштування):

```
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 програми **weblate**, коли віртуальне середовище Weblate є активним, див. [Команди керування](#).
- Припинити роботу тестового сервера можна за допомогою натискання комбінації клавіш Ctrl+C.

Додавання перекладу

1. Відкрийте адміністративний інтефейс (`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 сховище системи керування версіями і визначає, які файли слід перекладати. Див. [Налаштовування складників](#), щоб дізнатися більше.

Важливими полями тут є такі: назва складника, адреса сховища системи керування версіями та маска для пошуку придатних до перекладу файлів. У Weblate передбачено підтримку широкого діапазону форматів, зокрема файлів PO gettext PO, рядків ресурсів Android, рядкових властивостей iOS, властивостей Java та файлів Qt Linguist. Див. [Підтримувані формати файлів](#), щоб дізнатися більше.

3. Щойно буде виконано вказані вище дії (це може бути доволі тривалим процесом, залежно від розмірів вашого сховища системи керування версіями та кількості повідомлень, які слід перекласти), ви можете розпочати переклад.

Встановлення 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):

- 2 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 сховищ 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.

Встановлення

Вимоги до системи

Встановіть залежності, потрібні для збирання модулів Python (див. *Вимоги для програмного забезпечення*):

```
brew install pango libjpeg python git libyaml gobject-introspection
pip3 install virtualenv
```

Переконайтеся, що рір зможе знайти версію libffi, яка надається homebrew — це потрібно буде на кроці збирання встановленої версії.

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig"
```

Встановіть 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

# Caching backend: Redis
```

(continues on next page)

(continued from previous page)

```
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 --python=python3 ~/weblate-env
```

2. Активуйте `virtualenv` для Weblate:

```
. ~/weblate-env/bin/activate
```

3. Встановіть Weblate разом із усіма залежностями:

```
pip install Weblate
```

4. Встановіть драйвер бази даних:

```
pip install psycopg2-binary
```

5. Встановіть бажані додаткові залежності, залежно від можливостей, якими ви бажаєте користуватися (деяких з них можуть потребувати додаткових системних бібліотек, див. [Необов'язкові залежності](#)):

```
pip install ruamel.yaml aeidon boto3 zeep chardet tesseract
```

Налаштовування Weblate

Примітка: У наступних кроках ми припускаємо, що віртуальне середовище, яке використовується Weblate, є активним (активувати його можна за допомогою `. ~/weblate-env/bin/activate`). Якщо середовище не активовано, вам доведеться вказати повний шлях до команди **weblate** як `~/weblate-env/bin/weblate`.

1. Скопіюйте файл `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` до `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Скоригуйте значення у новому файлі `settings.py` відповідно до ваших уподобань. Ви можете скористатися типовим прикладом для тестування, але вам варто внести зміни для вашої промислової конфігурації, див. [Коригування налаштувань](#).
3. Створіть базу даних і її структуру для Weblate (у прикладі вказано параметри для PostgreSQL, ознайомтеся із розділом [check](#) [Налаштування бази даних для Weblate](#), щоб дізнатися більше про готову до промислової роботи конфігурацію):

```
weblate migrate
```

4. Створіть обліковий запис адміністративного користувача і скопіюйте пароль, який буде виведено, до буфера обміну даними, а також збережіть його для наступного використання:

```
weblate createadmin
```

- Зберіть статичні файли для 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 рекомендуємо це у всіх інших варіантах. Див. *Фонові завдання з використанням Celery*, щоб дізнатися більше:

```
~/weblate-env/lib/python3.7/site-packages/weblate/examples/celery start
```

- Запустіть сервер для розробки (див. *Запуск сервера*, щоб дізнатися більше про промислові налаштування):

```
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 програми **weblate**, коли віртуальне середовище Weblate є активним, див. *Команди керування*.
- Припинити роботу тестового сервера можна за допомогою натискання комбінації клавіш Ctrl+C.

Додавання перекладу

- Відкрийте адміністративний інтерфейс (`http://localhost:8000/create/project/`) і створіть проект, який ви хочете перекладати. Див. *Налаштування проекту*, щоб дізнатися більше.

Усе, що вам слід тут вказати, це назва проекту і адреса його сайту.

- Створіть складник, який є справжнім об'єктом для перекладу — він вказує на сховище системи керування версіями і визначає, які файли слід перекладати. Див. *Налаштовування складників*, щоб дізнатися більше.

Важливими полями тут є такі: назва складника, адреса сховища системи керування версіями та маска для пошуку придатних до перекладу файлів. У Weblate передбачено підтримку широкого діапазону форматів, зокрема файлів PO gettext PO, рядків ресурсів Android, рядкових властивостей iOS, властивостей Java та файлів Qt Linguist. Див. *Підтримувані формати файлів*, щоб дізнатися більше.

- Щойно буде виконано вказані вище дії (це може бути доволі тривалим процесом, залежно від розмірів вашого сховища системи керування версіями та кількості повідомлень, які слід перекласти), ви можете розпочати переклад.

Встановлення з початкового коду

1. Будь ласка, спочатку виконайте настанови зі встановлення для вашої системи:
 - *Встановлення на Debian і Ubuntu*
 - *Встановлення у SUSE та openSUSE*
 - *Встановлення 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. Встановіть 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. Скоригуйте значення у новому файлі settings.py відповідно до ваших уподобань. Ви можете скористатися типовим прикладом для тестування, але вам варто внести 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  
weblate compilemessages
```

Примітка: Цей крок слід повторювати кожного разу, коли ви оновлюватимете сховище.

Встановлення 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/>>.

Встановлення

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.

Вебконсоль

Скопіюйте вміст файла `template.yml` та імпортуйте його до вашого проєкту, а потім скористайтесь кнопкою **Create** у вебконсолі OpenShift для створення вашої програми. У вебконсолі програма попросить вас надати значення для усіх 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

The parameters that you can override are listed in the parameters section of the template. You can list them with the CLI by using the following command and specifying the file to be used:

```
$ 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.

Встановлення 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>>.

Встановлення

```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

Залежно від налаштувань та вашого досвіду, виберіть для себе відповідний спосіб встановлення:

- install/docker, рекомендовано для промислових налаштувань.
- Встановлення у віртуальному середовищі, рекомендовано для промислових конфігурацій:
 - *Встановлення на Debian і Ubuntu*
 - *Встановлення у SUSE та openSUSE*
 - *Встановлення у RedHat, Fedora і CentOS*
 - *Встановлення на macOS*
- install/source, рекомендовано для розробки.
- *Встановлення на OpenShift*
- *Встановлення 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.celeryproject.org/>

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'язкова, призначено для підтримки PHP) <https://github.com/viraptor/phply>

tesseract (необов'язкова, для підтримки оптичного розпізнавання тексту на знімках вікон) <https://github.com/sirfz/tesseract>

akismet (необов'язкова, призначено для захисту пропозицій від спаму) <https://github.com/ubernostrum/akismet>

ruamel.yaml (необов'язковий, для *Файли YAML*) <https://pypi.org/project/ruamel.yaml/>

Zeep (необов'язковий, для *Служба термінології Microsoft*) <https://docs.python-zeep.org/>

aeidon (необов'язковий, для *Файли субтитрів*) <https://pypi.org/project/aeidon/>

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 gir <https://cairographics.org/>, <https://pango.gnome.org/>, see *Pango і 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 і Cairo

Змінено в версії 3.7.

Weblate використовує Pango і 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 і Cairo, і 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 додаткову інформацію щодо ідентифікації можна знайти тут: [<https://keybase.io/nijel>](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@braiiins.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@braiiins.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 ключа 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
```

(continues on next page)

(continued from previous page)

```
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 (Brains) <michal.cihar@brains.cz>" [ultimate]
→ [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 Weblate 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 вебсервера 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 -O weblate weblate
```

Підказка: Якщо ви не хочете робити користувача Weblate суперкористувачем у PostgreSQL, ви можете пропустити це. У такому випадку вам доведеться виконати деякі з кроків з перенесення вручну від імені суперкористувача PostgreSQL у схемі, яку використовуватиме Weblate:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm 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": "",
    }
}
```

The database migration performs `ALTER ROLE` on database role used by Weblate. In most cases the name of the role matches username. In more complex setups the role name is different than username and you will get error about non-existing role during the database migration (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). This is known to happen with Azure Database for PostgreSQL, but it's not limited to this environment. Please set `ALTER_ROLE` to change name of the role Weblate should alter during the database migration.

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:

- Скористайтесь 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.
- Встановіть рівень ізоляції `READ COMMITTED`.
- Режим SQL має бути встановлено у значення `STRICT_TRANS_TABLES`.

Підказка: Якщо ви бачите повідомлення про помилку `#1071 - Specified key was too long; max key length is 767 bytes`, будь ласка, встановіть `Innodb_large_prefix`, як це описано вище.

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_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`.

Примітка: Ви можете перевірити, чи працює вихідна пошта належними чином, за допомогою команди керування `sendtestemail` (див. [Виклик команд керування](#), щоб ознайомитися із настановами щодо того, яка викликати її у різних середовищах).

Дивись також:

[Налаштування сервера електронної пошти](#) щодо налаштування вихідної електронної 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`.

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 сайту, які отримуватимуть сповіщення при помилках, зокрема сповіщення щодо невдалого об'єднання або помилок Django.

Дивись також:

`ADMINS`

`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 вилучення 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`

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`.

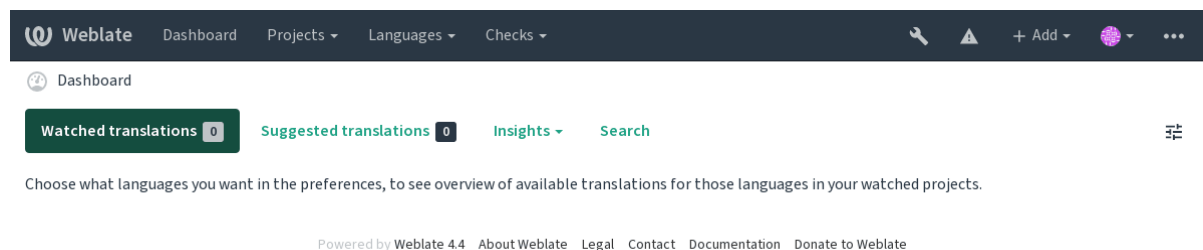
Щойно потрібні дії буде виконано, вам слід також ознайомитися зі *Звітом щодо швидкодії* в інтерфейсі адміністратора, там ви зможете знайти підказки щодо потенційних недоліків у налаштуваннях на вашому сайті.

Дивись також:

Налаштування, Керування доступом

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
```

Дивись також:

Deployment checklist

Вимкнути режим діагностики

Вимикання режиму діагностики Django (*DEBUG*):

```
DEBUG = False
```

З увімкненим режимом діагностики Django зберігає усі виконані запити і показує користувачеві зворотні трасування помилок. Така надмірна інформація є небажаною у промисловій конфігурації.

Дивись також:

Коригування налаштувань

Належне налаштування записів адміністраторів

Встановіть 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

Встановіть 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, Встановіть належний домен сайта

Встановіть 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: Встановлення 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, скористайтеся PostgreSQL для промислових середовищ. Див. *Налаштування бази даних для Weblate*, щоб дізнатися більше.

Дивись також:

Налаштування бази даних для Weblate, Перенесення даних з інших баз даних до PostgreSQL, Коригування налаштувань, Databases

Увімкніть кешування

Якщо можна, скористайтеся 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",
        },
    },
}
```

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"),
        "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 помилку, 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, Встановіть 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

Домашній каталог

Змінено в версії 2.1: У поточній версії потреби у цьому немає, оскільки нові версії Weblate зберігають усі дані у `DATA_DIR`.

Домашній каталог користувача, від імені якого запущено Weblate, має існувати і бути придатним до запису від імені цього користувача. Особливо важливим, якщо ви хочете скористатися SSH для доступу до приватних сховищ коду, може бути забезпечення доступу Git до цього каталогу (це залежить від використаної вами версії Git).

Змінити каталог, який використовується Weblate можна за допомогою файла `settings.py`. Наприклад, щоб використати каталог `configuration` у ієрархії тек Weblate, зробіть так:

```
os.environ["HOME"] = os.path.join(BASE_DIR, "configuration")
```

Примітка: У Linux та інших системах UNIX шлях до домашнього каталогу користувача визначається у `/etc/passwd`. У багатьох дистрибутивах типовим є непридатний до запису каталог для користувачів, записи яких призначено для обслуговування інтернет-даних (зокрема `apache`, `www-data` або `wwwrun`). Тому вам слід або запустити Weblate від імені іншого користувача, або змінити значення цього параметра.

Дивись також:

Доступ до сховищ

Завантаження шаблонів

Рекомендуємо скористатися засобом завантаження кешованих шаблонів для Django. Цей засіб кешує оброблені шаблони і запобігає потребі у повторній обробці для кожного окремого запису. Ви можете налаштувати його за допомогою наведеного нижче фрагмента коду (у ньому є важливим параметр `loaders`):

```
TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [
            os.path.join(BASE_DIR, "templates"),
        ],
        "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",
            ],
            "loaders": [
                (
                    "django.template.loaders.cached.Loader",
                    [
                        "django.template.loaders.filesystem.Loader",
                        "django.template.loaders.app_directories.Loader",
                    ],
                ),
            ],
        },
    ],
]
```

Дивись також:

```
django.template.loaders.cached.Loader
```

Запуск завдань щодо супроводу

Для забезпечення оптимальної швидкодії варто запускати деякі завдання із 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. *«Лінійні» 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 усього тексту та 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. Наприклад, при використанні Apache вам, можливо, слід встановити їх у файлі `/etc/apache2/envvars`:

```
export LANG='en_US.UTF-8'
export LC_ALL='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 коді 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

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`.

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:

Стиснення клієнтських даних, Deploying Django, Deploying 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/python3.7/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`):

```
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
↪env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your
↪setup.
server {
```

(continues on next page)

(continued from previous page)

```

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/ {
    # 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`):

```

# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
↪env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your_
↪setup.
[uwsgi]
plugins      = python3
master       = true
protocol     = uwsgi
socket       = 127.0.0.1:8080
wsgi-file    = /home/weblate/weblate-env/lib/python3.7/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

```

(continues on next page)

(continued from previous page)

```
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
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
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
↪env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your_
↪setup.
#
<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/
```

(continues on next page)

(continued from previous page)

```

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
WSGIProcessGroup weblate
WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias / /home/weblate/weblate-env/lib/python3.7/site-packages/
↪weblate/wsgi.py process-group=weblate request-timeout=600
WSGIPassAuthorization On

<Directory /home/weblate/weblate-env/lib/python3.7/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 mod_wsgi 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
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
↪env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your_
↪setup.
#
<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/

```

(continues on next page)

(continued from previous page)

```

<Directory /home/weblate/data/media/>
    Require all granted
</Directory>

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**

Змінено в версії 1.3: Підтримку передбачено з версії Weblate 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
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
→env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your
→setup.
#
<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
    WSGIProcessGroup weblate

```

(continues on next page)

(continued from previous page)

```

WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.7/site-packages/
↳weblate/wsgi.py process-group=weblate request-timeout=600
WSGIPassAuthorization On

<Directory /home/weblate/weblate-env/lib/python3.7/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.

Для обробки фонових завдань Weblate використовує Celery. Параметри у прикладі призначено для енергійної конфігурації, яка обробляє усі завдання на місці. Втім, ви можете змінити параметри обробки на щось помірніше для вашої промислової конфігурації.

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

```

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

```

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

```

(continues on next page)

(continued from previous page)

```

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"

# 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="/var/run/celery/weblate-%n.pid"
CELERYD_LOG_FILE="/var/log/celery/weblate-%n%I.log"
CELERYD_LOG_LEVEL="INFO"

# Internal Weblate variable to indicate we're running inside Celery
CELERY_WORKER_RUNNING="1"

```

Налаштування logrotate, які слід зберегти як `/etc/logrotate.d/celery`:

```

/var/log/celery/*.log {
    weekly
    missingok
    rotate 12
    compress
    notifempty
}

```

Примітка: Процес Celery має бути запущено від імені того самого користувача, що і Weblate і процес WSGI, інакше файли у `DATA_DIR` буде збережено від імені різних власників, що призведе до проблем під час роботи сервера.

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

Ви можете скористатися `celery_queues` для перегляду поточної довжини черг завдань Celery. 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:

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.

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": "master",
    "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 створення дампу бази даних на одному сервері з наступним імпортуванням його на іншому сервері. Ви також можете скористатися реплікацією, якщо у базі даних передбачено її підтримку.

Найкращим підходом є використання природних інструментів бази даних, оскільки вони, зазвичай, є найефективнішими (наприклад, `mysqldump` або `pg_dump`). Якщо ви хочете перенести дані між різними базами даних, єдиним варіантом може бути використання засобів керування Django для створення дампів та імпортування бази даних:

```
# Export current data
weblate dumpdata > /tmp/weblate.dump
# Import dump
weblate loaddata /tmp/weblate.dump
```

Перенесення сховищ системи керування версіями

Сховища 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:

- *Встановлення за допомогою Docker*
- *Встановлення на OpenShift*
- *Встановлення на Kubernetes*

2.2.1 Діаграма Гельма

Ви можете встановити Weblate на Kubernetes за допомогою Helm. Див. <<https://github.com/WeblateOrg/helm/tree/master/charts/weblate>>, щоб дізнатися більше.

2.2.2 Стос Weblate Bitnami

У Bitnami передбачено стек Weblate для багатьох платформ у <<https://bitnami.com/stack/weblate>>. Конфігурацію буде скориговано під час встановлення, документацію наведено у <<https://bitnami.com/stack/weblate/README.txt>>.

2.2.3 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.

2.3.2 Загальні настанови щодо оновлення

Перш ніж оновлювати програми, 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 із *настановами щодо специфіки версії* до оновлення. Якщо ви пропустили декілька попередніх версій, будь ласка, виконайте настанови для усіх версій, які ви пропускаєте при оновленні. Іноді краще оновитися до проміжної версії, щоб забезпечити безпроблемне перенесення даних. Оновлення із пропусканням декількох випусків мають працювати, але їхня перевірка на коректність виконується набагато менш ретельно, ніж перевірка оновлень без пропускання 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
```

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
```

3. Оновіть файл налаштувань. Зверніться до файла `settings_example.py` або *специфічних для випуску настанов*, щоб ознайомитися із потрібними для цього діями.
4. Оновіть структуру бази даних:

```
weblate migrate --noinput
```

5. Зберіть оновлені статичні файли (див. [Запуск сервера](#) і [Обслуговування статичних файлів](#)):

```
weblate collectstatic --noinput
```

6. Стисніть файли JavaScript і CSS (необов'язково, див. [Стиснення клієнтських даних](#)):

```
weblate compress
```

7. 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
```

8. Перевірте, чи є працездатною ваша конфігурація (див. також *Промислові налаштування*):

```
weblate check --deploy
```

9. 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 невідтримуваними — 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 невідтримуваними — 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`.
- У комп'ютерному перекладі *DeepL* тепер використовується версія 2 програмного інтерфейсу. У вас може виникнути потреба у коригуванні `MT_DEEPL_API_VERSION`, якщо ваша поточна передплата DeepL не передбачає використання нової версії.

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 програм.
- Ключі для заснованих на JSON форматів більше не включають початкової крапки. Рядки скориговано під час перенесення бази даних, але зовнішні компоненти 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 не-повторювані 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 перевіряках 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 у 4.3.1:** налаштування Celery змінено так, щоб додавалася черга `memory`. Змініть ваші сценарії запуску та значення параметра `CELERY_TASK_ROUTES`.
- **Змінено у 4.3.2:** метод `post_update` у додатках у новій версії 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`.
- Тепер потрібен Django 3.1.
- Якщо ви використовуєте MySQL або MariaDB, підвищено вимоги щодо мінімальних версій, див. *MySQL і MariaDB*.

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 3.5. 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 це єдина підтримувана база даних, починаючи з випуску 4.0. 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 -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",
```

(continues on next page)

(continued from previous page)

```

# 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",
    # 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 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*.

Попередження: До автоматизованих резервних копій буде включено лише базу даних PostgreSQL. Резервні копії даних інших рушіїв керування базами даних має бути збережено вручну. Рекомендуємо вам перейти на PostgreSQL, див. *Налаштування бази даних для Weblate* і *Перенесення даних з інших баз даних до PostgreSQL*.

Резервні копії з використанням Borg є нарощувальними, а Weblate налаштовування на збереження таких резервних копій:

- резервний копій із 14-денним період створення
- резервних копій із 8-тижневим періодом створення
- резервних копій із 6-місячним періодом створення

 Weblate
 Dashboard Projects Languages Checks
 + Add

 Manage / Backups

Backup process triggered

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

[Tools](#)
[Billing](#)

Backup service: /tmp/tmpd6i6hug9weblate

Backup service credentials

Backup repository

/tmp/tmpd6i6hug9weblate

Passphrase

qVnji6y*fsRcY0mt5Op7&bo3(^^VamUTBtG\$10I3ACcbwXqjWo

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

Backup performed

Repository initialization

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 for remote SSH backups.

Add

Ключ шифрування Borg

BorgBackup створює зашифровані резервні копії і без пароля ви не зможете відновити дані з резервної копії. Пароль створюється при додаванні нової служби резервного копіювання, і вам слід скопіювати його і зберігати його у безпечному місці.

Якщо ви користуєтеся *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`

2.4.2 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 вручну додано з метою убезпечення ваших даних. Без вашої згоди під час процедури реєстрації до сховища резервних копій не буде надіслано жодних даних.

2.4.3 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 файлової 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, будь ласка, переконайтеся, що місце зберігання резервних копій відкрито як том з контейнера Weblate. Якщо ви цього не зробите, резервні копії буде втрачено при перезапуску контейнера Docker.

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 композиції Docker і 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

Передбачено підтримку резервних копій з використанням SSH. На сервері SSH має бути встановлено [BorgBackup](#). Weblate встановлює з'єднання з використанням ключа SSH. Будь ласка, переконайтеся, що ключ SSH Weblate буде прийнято сервером (див. [Ключ SSH Weblate](#)).

Підказка: *Передбачене у Weblate сховище резервних копій* надає вам доступ до автоматичного створення резервних копій на віддалених серверах.

2.4.4 Відновлення з BorgBackup

1. Відновлення доступу до вашого сховища резервних копій і приготування вашого пароля до резервних копій.
2. Отримайте список резервних копій на сервері за допомогою команди `borg list СХОВИЩЕ`.
3. Відновіть бажану резервну копію до поточного каталогу за допомогою команди `borg extract СХО–ВИЩЕ : :ARCHIV`.
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](#).
6. Скопіюйте увесь каталог відновлених даних до місця, яке налаштовано змінною `DATA_DIR`.

Сесія 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`

2.4.5 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.

Найважливішим сховищем даних є база даних. Налаштуйте регулярне резервне копіювання вашої бази даних — якщо її буде втрачено, буде втрачено усі ваші налаштування перекладу.

Власна система резервного копіювання бази даних

Рекомендованим способом створення дампу бази даних є використання власних інструментів засобу керування базою даних, зокрема `pg_dump` або `mysqldump`. Такий інструмент, зазвичай, працює краще за засіб резервного копіювання Django і може відновлювати таблиці повністю із 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 дані, 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

Зберігається у `DATA_DIR/backups`.

Weblate створює тут дампи різноманітних даних, і ви можете включити ці файли для отримання повної 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`.

Типово, резервні копії зберігаються у форматі простого тексту. Втім, її можна стиснути або повністю пропустити за допомогою параметра `DATABASE_BACKUP`.

Сховища систем керування версіями

Зберігається у `DATA_DIR/vcs`.

У сховищах систем керування версіями зберігаються копії базових сховищ зі змінами, які внесено Weblate. Якщо для усіх ваших складників перекладу увімкнено негайний запис до сховища після внеску, усі внесені Weblate буде включено до основного сховища, і вам не потрібно буде створювати резервні копії сховищ на боці Weblate. Сховища можна буде просто знову клонувати із основного сховища 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 сховищами 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
```

2.4.6 Відновлення зі створеної вручну резервної копії

1. Відновлення усіх даних зі створеної вами резервної копії.
2. Оновлення усіх сховищ за допомогою `updategit`.

```
weblate updategit --all
```

2.4.7 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, скористайтесь примусовою перевіркою адреси електронної пошти на боці Weblate, налаштувавши для таких служб параметр `FORCE_EMAIL_VALIDATION`. Приклад:

```
SOCIAL_AUTH_OPENSUSE_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.

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 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
```

(continues on next page)

(continued from previous page)

```
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/`.

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. *Встановіть належний домен сайту*.

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.BitbucketOAuth",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_BITBUCKET_KEY = "Bitbucket Client ID"
SOCIAL_AUTH_BITBUCKET_SECRET = "Bitbucket Client Secret"
SOCIAL_AUTH_BITBUCKET_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. *Встановіть належний домен сайту*.

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://СЕРВЕР 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
```

(continues on next page)

(continued from previous page)

```
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. *Встановіть 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. *Встановіть 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"
```

(continues on next page)

(continued from previous page)

```
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. *Встановіть 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. *Встановіть 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 переспрямовування — `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. *Встановіть належний домен сайта*.

Дивись також:

[Slack](#)

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.

Порада: Ви все ще можете скористатися для створених тут вручну записів користувачів розпізнаванням за паролями для адміністративного інтерфейсу. Просто перейдіть до `/admin/`.

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_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",
    }
}
```

Дивись також:

Налаштування 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
```

Попередження: Якщо використано `django-auth-ldap` давнішої за версію 1.3.0, *Автоматичні призначення груп* не працюватиме належним чином для новостворених записів користувачів.

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",
)

# 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 `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(
```

(continues on next page)

(continued from previous page)

```
)
    "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 файлу `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",
)
```

(continues on next page)

(continued from previous page)

```
# 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 вашої програми
- 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 програми 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

Змінено в версії 3.0: До версії Weblate 3.0 систему прав доступу було засновано на Django, але тепер її побудовано окремо для Weblate. Якщо ви користуєтеся застарілою версією, будь ласка, зверніться до документації з вашої версії — наведені тут дані є незастосовними до неї.

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.

Систему прав доступу засновано на групах і ролях, де ролі визначають набір прав доступу, а групи пов'язують їх із користувачами та перекладами, див. *Користувачі, ролі, групи і права доступу*, щоб дізнатися більше.

Після встановлення буде створено типовий набір груп, і ви зможете скористатися ними для визначення ролей користувачів для усього встановленого екземпляра (див. *Типові групи і ролі*). Крім того, якщо увімкнено *Керування доступом на рівні проєкту*, ви зможете пов'язувати користувачів із специфічними проєктами перекладу. Прискіпливіше налаштування можна виконати за допомогою розділу *Нетипове керування доступом*.

2.6.1 Типові налаштування

Блокування Weblate

Щоб повністю заблокувати встановлення Weblate, ви можете скористатися *REQUIRE_LOGIN*, щоб змусити користувачів входити до системи та *REGISTRATION_OPEN*, щоб запобігти новим реєстраціям користувачів.

Права доступу на рівні сайта

Щоб виконати керування правами доступу для усього екземпляра, просто додайте користувачів до груп *Users* (ця дія виконується типово за допомогою *Автоматичні призначення груп*), *Reviewers* та *Managers*. Усі проєкти налаштуйте як *Public* (див. *Керування доступом на рівні проєкту*).

Права доступу на рівні проєктів

Примітка: Ця функція недоступна для проєктів, які використовують план Hosted Libre.

Налаштуйте ваші проєкти як *Protected* або *Private* і керуйте користувачами окремих проєктів за допомогою інтерфейсу Weblate.

Додавання прав доступу до мов, компонентів або проєктів

Примітка: Ця функція недоступна для проєктів, які використовують план Hosted Libre.

Крім того, ви можете надати будь-якому користувачу права доступу на основі проєкту, складника або мови. Для цього створіть групу (наприклад, *Czech translators*) і налаштуйте її для заданого ресурсу. Усі призначені права доступу для вибраних ресурсів буде пов'язано із учасниками цієї групи.

Це працюватиме без додаткових налаштувань, якщо використовується обмеження прав доступу на рівні проєкту. Щоб визначити права доступу на рівні усього екземпляра Weblate, вам, ймовірно, слід вилучити ці права доступу з групи *Users* або змінити автоматичне пов'язування усіх користувачів із цією групою (див. *Автоматичні призначення груп*).

Дивись також:

Перевірка прав доступу

2.6.2 Керування доступом на рівні проєкту

Примітка: Якщо увімкнено ACL, усім користувачам буде заборонено доступ до усіх даних у вказаному проєкті, якщо ви явним чином не дозволите такий доступ.

Примітка: Ця функція недоступна для проєктів, які використовують план Hosted Libre.

Ви можете обмежити доступ користувача до окремих проєктів. Ця можливість вмикається пунктом *Керування доступом* у налаштуваннях відповідного проєкту. У відповідь буде автоматично створено декілька груп для цього проєкту, див. *Попередньо визначені групи*.

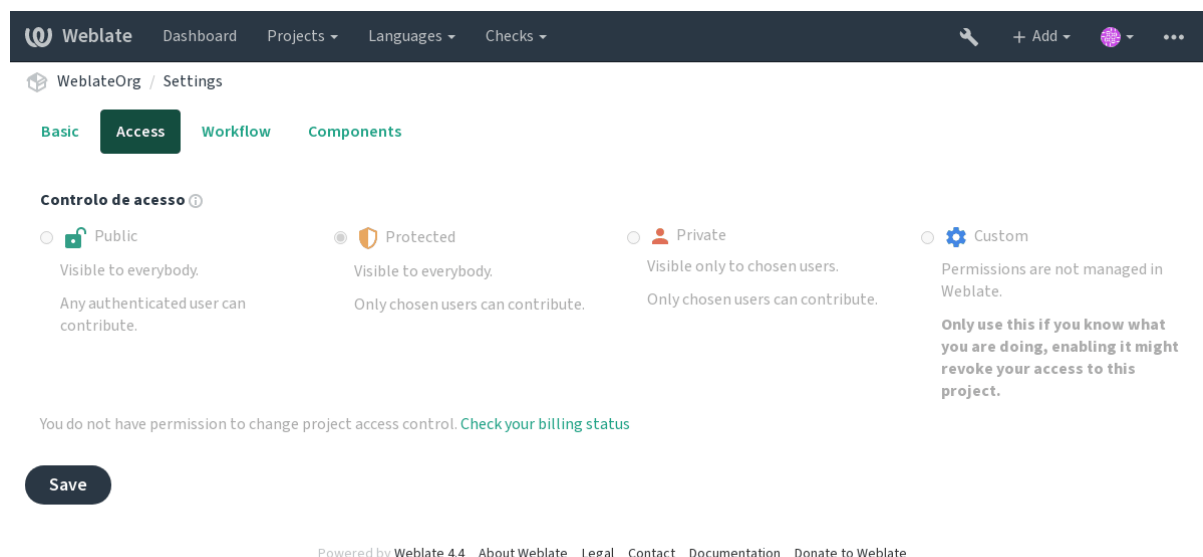
Для *Керування доступом* передбачено такі варіанти:

Відкритий Усі бачать і можуть перекладати

Захищений Доступний до перегляду усіма, але придатний до перекладу лише вибраними користувачами

Приватний Доступний до перегляду і перекладу лише вибраним користувачам

Власний Сам Weblate не керує записами користувачів, див. *Нетипове керування доступом*.



Щоб дозволити доступ до цього проєкту, вам слід додати права доступу або безпосередньо для запису певного користувача або групи користувачів у адміністративному інтерфейсі Django або за допомогою керування записами користувачів на сторінці проєкту, як це описано у розділі *Керування правами доступу на рівні проєкту*.

Примітка: Навіть якщо ACL увімкнено, буде доступною деяка загальна інформація щодо вашого проєкту:

- Статистика щодо усього екземпляра Weblate включено із обліковими даними для усіх проєктів.
- Резюме щодо мов для усього екземпляра Weblate, включно із числовими даними для усіх проєктів.

2.6.3 Автоматичні призначення груп

Ви можете налаштувати Weblate на автоматичне додавання користувачів до груп на основі їхніх адрес електронної пошти. Ця автоматична прив'язка виконуватиметься лише під час створення облікових записів.

Виконати налаштування можна за допомогою адміністративного інтерфейсу Django для кожної з груп (у розділі *Authentication*).

Примітка: Автоматична прив'язка до груп для груп *Users* і *Viewers* виконуватиметься Weblate завжди при перенесенні даних. Якщо ви хочете вимкнути її, просто встановіть для формального виразу прив'язки значення `^$`, якому не відповідає жоден рядок.

2.6.4 Користувачі, ролі, групи і права доступу

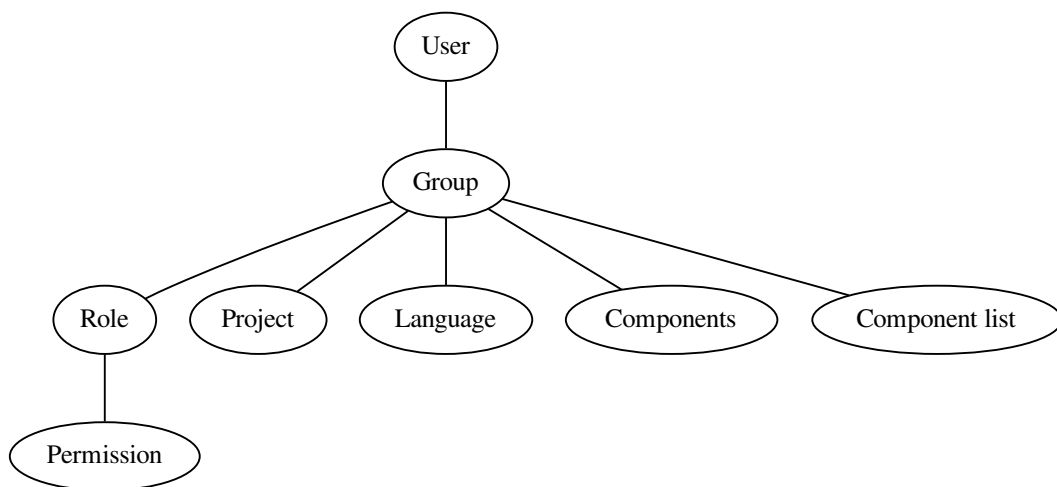
Моделі розпізнавання складаються з декількох об'єктів:

Права доступу Персональні права доступу, які визначаються Weblate. Ви не можете визначати персональні права доступу — 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 груп одночасно.

Група Групи поєднують об'єкти ролей, користувачів та розпізнавання (проекти, мови і списки складників).



Перевірка прав доступу

Кожного разу, коли виконується перевірка прав доступу для визначення того, чи може певний користувач виконувати вказану дію, враховується область дії, а перевірки застосовуються у такому порядку:

1. Встановлюється відповідність групи *Список складників* складнику або проекту (для доступу на рівні проекту).
2. Встановлюється відповідність групи *Складники* складнику або проекту для доступу (на рівні доступу до проекту).
3. Встановлюється відповідність групи *Проекти* проекту для доступу.

Як бачите, надання доступу до складника автоматично надає користувачеві доступу до проекту, у якому він міститься.

Примітка: Використовується лише перше правило. Отже, якщо встановлено значення усіх параметрів — *Список складників*, *Складники* і *Проект*, — застосовано буде лише параметр *Список складників*.

Під час перевірки прав доступу до перекладу виконується додатковий крок:

4. Group *Languages* are matched against accessed translations, it is ignored for component or project level access.

Підказка: Для автоматизації включення усіх мов або проектів ви можете скористатися пунктами *Вибір мови* та *Вибір проекту*.

Перевірка доступу до проекту

Користувач має бути учасником групи, пов'язаної із проектом або будь-яким складником у проекті. Участь достатньо — для доступу до проекту не потрібні якісь особливі права доступу (це використано у типовій групі *Переглядачі*, див. *Типові групи і ролі*).

Перевірка доступу до компонента

Користувач отримує доступ до необмеженого складника, щойно він отримує доступ до проекту, у якому він міститься. Із увімкненим параметром *Обмежений доступ* доступ до складника потребує явного доступу до складника (або списку складників, який його містить).

2.6.5 Керування користувачами і групами

Керувати усіма користувачами та групами можна за допомогою адміністративного інтерфейсу Django, доступного за адресою `/admin/`.

Керування правами доступу на рівні проекту

Примітка: Ця можливість працює лише для керованих ACL проектів, див. *Керування доступом на рівні проекту*.

Користувачі з правами доступу *Керування доступом до проекту* (див. *Керування доступом*) можуть також керувати записами користувачів у проектах із увімкненим за допомогою сторінки проекту керуванням доступом. За допомогою інтерфейсу ви можете виконувати такі дії:

- Додавання наявних користувачів до проекту
- Запрошення нових користувачів до проекту
- Зміна прав доступу користувачів
- Відкликання доступу користувачів

Нове в версії 3.11.

- Повторно надіслати запрошення користувачам електронною поштою, скасовує будь-які раніше надіслані запрошення

Керувати користувачами можна за допомогою меню *Керувати* проєкту:

Users

Username	Full name	E-mail	Last login	Administration	Billing	Glossary	Languages	Memory	Screenshots	Sources	Translate	VCS
testuser	Weblate Test	weblate@example.org	22 seconds ago	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

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

Invite new user

E-mail

Username

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

Full name

Invite

Powered by Weblate 4.4 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Дивись також:

Керування доступом на рівні проєкту

Попередньо визначені групи

Weblate постачається із попередньо визначеним набором груп для проєкту. Групи з цього набору можна пов'язувати із записами користувачів.

Administration

Має усі можливі права доступу до проєкту.

Glossary

Може змінювати глосарій (додавати або вилучати записи, а також вивантажувати дані).

Languages

Може керувати мовами перекладу (додавати або вилучати переклади).

Screenshots

Може керувати знімками вікон (додавати і вилучати знімки, а також пов'язувати зніми із початковими рядками).

Sources

Може редагувати початкові рядки у *Одномовні складники* і дані щодо початкових рядків.

Translate

Може перекладати проєкт і вивантажувати виконані поза межами вебредактора переклади.

VCS

Може керувати системою керування версіями і має доступ до експортованого сховища початкового коду.

Review

Може затверджувати переклади у процесі рецензування.

Billing

Має доступ до даних щодо фінансових рахунків (див. [Оплата](#)).

2.6.6 Нетипове керування доступом

Якщо вибрати варіант *Нетиповий* для *Керування доступом*, Weblate припинить керування доступом для вказаного проєкту — керування усіма записами користувачів і груп може бути здійснене за допомогою адміністративного інтерфейсу Django. За допомогою цього інтерфейсу можна здійснювати складніше керування доступом або налаштувати правила спільного доступу для усіх проєктів у межах окремого екземпляра Weblate. Якщо ви хочете типово увімкнути такий варіант для усіх проєктів, будь ласка, налаштуйте `DEFAULT_ACCESS_CONTROL`.

Попередження: Якщо буде увімкнено цей варіант, Weblate вилучить усі правила *Керування доступом на рівні проєкту*, які було створено для цього проєкту. Якщо ви виконаєте цю дію без адміністративних прав доступу до екземпляра, ви негайно втратите доступ до керування проєктом.

2.6.7 Типові групи і ролі

Ці ролі і групи створюються під час встановлення. Актуальність вбудованих ролей завжди підтримується перенесенням бази даних при оновленні, а усі нетипові зміни буде втрачено. Якщо ви хочете визначити власний набір прав доступу, будь ласка, визначте для нього нову роль.

Список привілеїв

Фінансові рахунки (див. [Оплата](#)) Переглядати фінансові дані [*Administration, Billing*]

Зміни Отримувати зміни [*Administration*]

Коментарі Залишати коментарі [*Administration, Edit source, Power user, Review strings, Translate*]

Вилучити коментар [*Administration*]

Складник Змінити параметри складника [*Administration*]

Блокувати складник, запобігаючи перекладу [*Administration*]

Глосарій Додавати записи глосарія [*Administration, Manage glossary, Power user*]

Редагувати записи глосарія [*Administration, Manage glossary, Power user*]

Вилучати записи глосарія [*Administration, Manage glossary, Power user*]

Вивантажувати записи глосарія [*Administration, Manage glossary, Power user*]

Автоматичні пропозиції Використовувати автоматичні пропозиції [*Administration, Power user*]

Проекти Змінити параметри проєкту [*Administration*]

Керувати доступом до проєкту [*Administration*]

Звіти Звіти щодо отримання даних [*Administration*]

Знімки екрана Додати знімок вікна [*Administration, Manage screenshots*]

Змінити знімок вікна [*Administration, Manage screenshots*]

Вилучити знімок вікна [*Administration, Manage screenshots*]

Джерельні рядки Редагувати дані початкового рядка [*Administration, Edit source*]

Рядки Додати нові рядки [*Administration*]

Ігнорувати виявлені під час перевірки помилки [*Administration, Edit source, Power user, Review strings, Translate*]

Редагувати рядки [*Administration, Edit source, Power user, Review strings, Translate*]

Рецензувати рядки [*Administration, Review strings*]

Змінювати рядок з примусовими пропозиціями [*Administration, Review strings*]

Редагувати початкові рядки [*Administration, Edit source, Power user*]

Пропозиції Приймати пропозиції [*Administration, Edit source, Power user, Review strings, Translate*]

Додавати пропозиції [*Add suggestion, Administration, Edit source, Power user, Review strings, Translate*]

Вилучити пропозицію [*Administration*]

Голосувати щодо пропозицій [*Administration, Edit source, Power user, Review strings, Translate*]

Переклади Розпочати новий переклад [*Administration, Manage languages, Power user*]

Виконувати автоматичний переклад [*Administration, Manage languages*]

Вилучати наявні переклади [*Administration, Manage languages*]

Розпочати переклад новою мовою [*Administration, Manage languages*]

Вивантаження Визначити автора вивантаженого перекладу [*Administration*]

Перезаписувати наявні рядки вивантаженими даними [*Administration, Edit source, Power user, Review strings, Translate*]

Вивантажувати рядки перекладу [*Administration, Edit source, Power user, Review strings, Translate*]

VCS Користуватися доступом до внутрішнього сховища [*Access repository, Administration, Manage repository, Power user*]

Внести зміни у внутрішнє сховище [*Administration, Manage repository*]

Витягнути зміни з внутрішнього сховища [*Administration, Manage repository*]

Очистити усі зміни у внутрішньому сховищі [*Administration, Manage repository*]

Бачити розташування основного сховища [*Access repository, Administration, Manage repository, Power user*]

Оновити внутрішнє сховище [*Administration, Manage repository*]

Привілеї на рівні сайту Використовувати інтерфейс керування

Додавати нові проекти

Додати визначення мов

Керувати визначеннями мов

Керувати групами

Управління користувачами

Керувати ролями

Керувати оголошеннями

Керувати пам'яттю перекладів

Керувати списками складників

Примітка: Привілеї на рівні сайту не надаються будь-якій типовій ролі. Ці привілеї є доволі широкими і дуже близькими до стану суперкористувача — більшість з них стосуються усіх проектів у встановленому вами Weblate.

Список груп

Під час встановлення (або після виконання *setupgroups*) буде створено вказані нижче групи. 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 групі містяться записи лише анонімних користувачів (див. *ANONYMOUS_USER_NAME*).

Ви можете вилучати ролі з цієї групи для обмеження прав доступу користувачів, які не пройшли розпізнавання.

Типові ролі: *Add suggestion*, *Access repository*

Переглядачі Ця роль забезпечує видимість відкритих проєктів для усіх користувачів. Типово, учасниками цієї групи є усі користувачі.

Типово, учасниками цієї групи є усі користувачі, використовується *Автоматичні призначення груп*.

Типові ролі: немає

Users Типова група для усіх користувачів.

Типово, учасниками цієї групи є усі користувачі, використовується *Автоматичні призначення груп*.

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 групи і користувачів Weblate, оскільки це може призвести до неочікуваних проблем. Якщо ви не хочете скористатися цими можливостями, просто вилучіть усі привілеї з цих записів.

2.7 Проєкти перекладу

2.7.1 Упорядкування перекладів

Weblate упорядковує придатні до перекладу дані системи керування версіями проєкту і складників у деревоподібну структуру.

- Об'єктом нижнього рівня є *проєкт*. Проєкт має містити усі переклади, які йому належать, разом (наприклад, переклади декількох версій програми та/або відповідної документації).
- Рівнем вище розташовано *складник*, який є, власне, складником, який перекладають. Ви визначаєте сховище 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, заданій у *складнику*) у сховищі 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 сховище для декількох складників за допомогою можливості *Внутрішні адреси Weblate*. Наполегливо рекомендуємо скористатися цією можливістю, якщо у вашому проєкті декілька складників використовують те саме сховище

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 компонентів 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:

The screenshot shows the Weblate Billing page. The top navigation bar includes 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below the navigation bar, there is a section for 'Your profile / Billing'. The main content area is divided into two columns. The left column contains a table for the 'Billing plan' with the following details:

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	2020-12-14 - 2020-12-16
Projects limit	Used 0 of 1
Projects	No projects currently assigned!

At the bottom of the left column, there is a button labeled 'Add new translation project' and a red button labeled 'Terminate billing plan'.

The right column contains a table for 'Invoices' with the following details:

Invoice period	Invoice amount	Download invoice
12/14/2020 - 12/16/2020	19.0 EUR	Not available

At the bottom of the page, there is a footer with the text: 'Powered by Weblate 4.4 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:

Weblate
Dashboard
Projects
Languages
Checks
+ Add

Create project

Add new 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.

Mailing list

weblate@lists.cihar.com

Mailing list for translators.

Translation instructions

https://weblate.org/contribute/

You can use Markdown and mention users by @username.

Billing

Weblate Test (Basic plan)

Save

Powered by Weblate 4.4
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Після створення проекту ви перейдете безпосередньо до сторінки проекту:

Weblate
Dashboard
Projects
Languages
Checks
+ Add

WeblateOrg

translated 100%

Components
Languages
Info
Search
Glossaries
Insights
Files
Tools
Manage
Share
Watching

Nothing to list here.

Add new translation component

Powered by Weblate 4.4
About Weblate
Legal
Contact
Documentation
Donate to Weblate

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 сховища 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 сховищі.

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 допо-

могою вебінтерфейсу або *програмного інтерфейсу*.

Перекласти документ Вивантажити один документ 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 сховища.

Спочатку вам слід заповнити назву і розташування сховища:

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.

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 with translations.

Source code repository ⓘ

https://github.com/WeblateOrg/demo.git

URL of a repository, use weblate://project/component for sharing with other component.



Repository branch ⓘ

Repository branch to translate

Continue

Powered by Weblate 4.4 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 + Add  ...

Create component

Add new translation component ?

Choose translation files to import ?

☐ Specify configuration manually

☐ File format `Android String Resource`, Filemask `app/src/main/res/values-*/strings.xml`

☐ File format `gettext PO file`, Filemask `weblate/langdata/locale/*/LC_MESSAGES/django.po`

☐ File format `gettext PO file`, Filemask `weblate/locale/*/LC_MESSAGES/django.po`

☐ File format `gettext PO file`, Filemask `weblate/locale/*/LC_MESSAGES/djangojs.po`

Continue

Powered by Weblate 4.4 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

На останньому кроці ви можете переглянути дані щодо складника перекладу і вкажіть додаткові подробиці:

Powered by [Weblate 4.4](#) [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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. [Memory Management](#).

Дивись також:

/devel/integration

Ці базові атрибути допомагають перекладачам проєкту та інформують їх:

Назва проєкту

Докладна назва проєкту, використовується для показу назви проєкту.

Ідентифікатор проєкту

Назва проєкту, яку можна включати до адрес.

Вебсайт проєкту

Адреса, за якою перекладачі можуть знайти додаткові відомості щодо проєкту.

Список листування

Список листування, у якому перекладачі можуть обговорити або прокоментувати переклади.

Вказівки щодо перекладу

Адреса до додаткового сайту з докладними настановами для перекладачів.

Встановлення 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*.

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*.

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*.

Дозволити гачки

Визначає, чи буде використано *скрипти слідування* для цього сховища.

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 сховищах 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:

Обробка кодів мов

2.7.4 Налаштовування складників

Складник групує дані для перекладу. Ви вводите адресу сховища 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 програми або додатки, глави 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*.

Сховище з джерелами

Сховище системи керування версіями, яке буде використано для записування змін.

Дивись також:

Див. *Доступ до сховищ*, 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 сховище має бути спільним із іншим складником. Див. *Внутрішні адреси Weblate*, щоб дізнатися більше.

Адреса для записування до сховища

Адреса сховища, яка використовуватиметься для запису даних до сховища. Цей параметр використовується лише для *Git* і *Mercurial*, а підтримку запису вимкнено для цих систем керування версіями, якщо значення цього параметра є порожнім.

Дивись також:

Див. *Доступ до сховищ*, щоб дізнатися більше про те, як вказати адресу сховища, і *Записування змін з Weblate*, щоб дізнатися більше про те, як записувати зміни до сховища з Weblate.

Переглядач сховища

Адреса навігатора сховищем, 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, 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 сховища

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 сховища Git.

Гілка сховища

Яку гілку слід отримати зі сховища керування версіями, і де слід шукати переклади.

Гілка для запису

Гілка, до якої слід записувати зміни. Не заповнюйте, якщо слід використовувати *Гілка сховища*.

Примітка: У поточній версії передбачено підтримку лише для Git, GitLab і GitHub. Буде проігноровано для усіх інших інтеграцій із системами керування версіями.

Маска файлів

Маска файлів для перекладу, включно із шляхом. У ній має бути один символ «*», який замінюватиме код мови (див. *Визначення мов*, щоб дізнатися про принципи обробки). Якщо ваше сховище 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, переклад джерела засновано на цьому файлі, але усі інші засновано на *Основний файл для одномовного перекладу*. Якщо рядок не перекладено у перекладі 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)»?

Заготовка 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. Скористайтесь цим варіантом, якщо ви хочете при створенні перекладу усі рядки було показано як порожні значення.

Дивись також:

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 коді. 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`.

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*.

Примусові перевірки

Список перевірок, які не можуть бути знехтувані, див. *Примусові перевірки*.

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 проєкту отримають про це сповіщення. Супровідники зможуть вирішувати, чи слід додавати мову до сховища.

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.

Дивись також:

adding-translation.

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 зі сховища основного коду. Підтримка варіантів оновлення залежить від системи керування версіями. Див. *Злиття чи перенесення*, щоб дізнатися більше.

Типове значення можна змінити за допомогою `DEFAULT_MERGE_STYLE`.

Повідомлення щодо внеску, додавання, вилучення, об'єднання та повідомлення додатків

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`.

Ім'я подавача

Ім'я автора внеску, яке буде використано у внесках Weblate. Автором завжди буде реальний перекладач. У деяких системах керування версіями підтримки визначення автора внеску не передбачено.

Типове значення можна змінити за допомогою `DEFAULT_COMMITER_NAME`.

Ел. пошта подавача

Адреса електронної пошти автора внеску, яке буде використано у внесках Weblate. Автором завжди буде реальний перекладач. У деяких системах керування версіями підтримки визначення адреси електронної пошти автора внеску не передбачено. Типове значення може бути змінено за допомогою параметра `DEFAULT_COMMITER_EMAIL`.

Відправляти при поданні

Визначає, чи слід записані зміни автоматично записувати до основного сховища коду. Якщо увімкнено, запис буде ініційовано одразу після того, як Weblate запише зміни до внутрішнього сховища (див. *«Лінії» внески*). Щоб насправді увімкнути запис, слід також налаштувати *Адресу запису до сховища*.

Вік змін для подання

Встановлює, 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`.

Блокувати при помилці

Вмикає блокування складника при помилці у сховищі (помилки отримання даних зі сховища, запису даних до сховища або об'єднання даних зі сховищем). Блокування у цьому випадку запобігає додаванню конфліктів, які доведеться вирішувати вручну.

Складник буде автоматично розблоковано, щойно у сховищі 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, вам варто вибрати *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 керівники 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.

Обмежений доступ

Типово, складник є видимим для усіх, хто має доступу до проекту, навіть якщо особа не може вносити зміни до складника. Це спрощує підтримання однорідності перекладу у межах проекту.

Вмикайте це, якщо ви хочете надати доступ до цього складника явним чином — права доступу на рівні проекту не застосовуватимуться, і вам слід вказати рівень прав доступу до складника або списку складників для надання доступу.

Типове значення можна змінити за допомогою `DEFAULT_RESTRICTED_COMPONENT`.

Підказка: Це стосується і керівників проекту — будь ласка, переконайтеся, що ви не втратите доступ до складника після перемикання його стану.

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

Доступними у шаблонах складників є такі змінні:

```
{{ language_code }} Код мови
{{ language_name }} Назва мови
{{ component_name }} Назва складника
{{ component_slug }} Ідентифікатор складника
{{ project_name }} Назва проекту
{{ 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.

Доступними у навігаторі сховища або шаблонах редактора є такі змінні:

{{branch}} поточна гілка
{{line}} рядок у файлі
{{filename}} назва файлу, ви також можете вилучити початкові частини за допомогою фільтра `parentdir`; приклад: `{{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 сховища 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 швидкість. 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 сховищ, 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 швидкість.
- 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 швидкість 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 десятків файлів перекладу (наприклад, для різних доменів gettext або частин програм 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 Обробка кодів мов

Під час обробки перекладів Weblate намагається прив'язати код мови (зазвичай, код зі списку ISO 639-1) 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 (наприклад, за допомогою ігнорування типового коду країни для вибраної мови — вибираємо `cs` замість `cs_CZ`).

Should that also fail, a new language definition will be created using the defaults (left to right text direction, one plural). The automatically created language with code `xx_XX` will be named as `xx_XX (generated)`. You might want to change this in the admin interface later, (see *Зміна визначень мов*) and report it to the issue tracker (see *Участь у розробці Weblate*), so that the proper definition can be added to the upcoming Weblate release.

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

2.8.2 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 напрямку запису тексту). Якщо ви цього не зробите, перекладачі не зможуть належним чином редагувати відповідні переклади.

2.8.3 Вбудовані визначення мов

До пакунка Weblate включено понад 550 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`.

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, передбачено 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](#)

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 усіх 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 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 *Заготовка 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:

Усталення 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 Android Використовується лише у програмах Android. Створює коди мов, подібні до `pt-rBR`.

Стиль Java Використовується у VCP із переважним використанням Java з застарілими кодами для китайської мови.

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, Обробка кодів мов

2.9 Безперервна локалізація

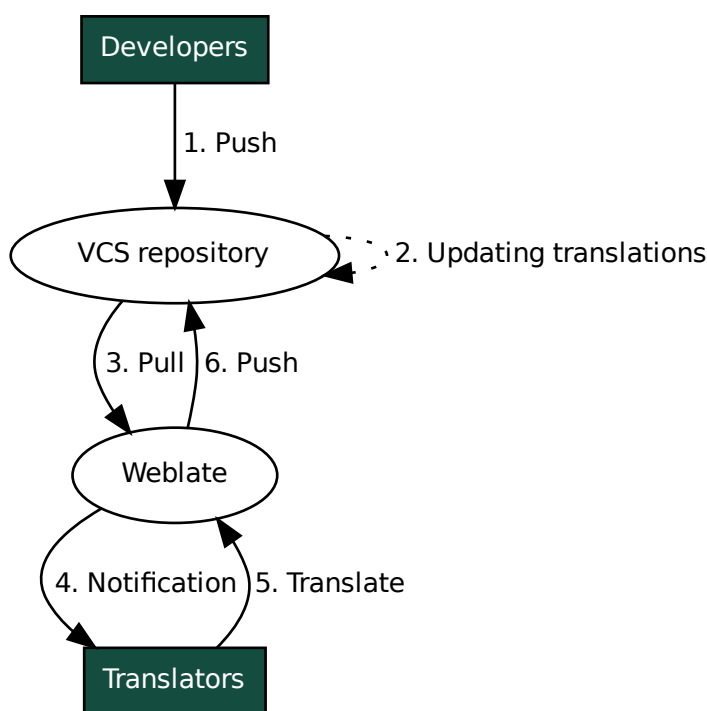
Готовою до використання є інфраструктура, за допомогою якої ви можете точно слідувати за розробкою проєкту. Перекладачі можуть працювати над перекладами неперервно, а не працювати над величезними обсягами перекладів нового тексту безпосередньо перед випуском.

Дивись також:

У `/devel/integration` описано базові способи інтеграції вашої розробки із Weblate.

Ось процедура:

1. Розробники змінюють код і записують його до сховища 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 зі сховища системи керування версіями, див. [Оновлення сховищ](#).
4. Щойно Weblate виявить зміни у перекладах, перекладачів буде сповіщено, залежно від встановлених параметрів підписки на сповіщення.
5. Перекладачі подають переклади за допомогою вебінтерфейсу Weblate або вивантажують зроблені поза інтернетом переклади.
6. Щойно переклад буде завершено, Weblate надсилає зміни до локального сховища (див. [«Лінійні» внески](#)) і записує їх до основного сховища, якщо має право на виконання цієї дії (див. [Записування змін з Weblate](#)).



2.9.1 Оновлення сховищ

Вам слід налаштувати певний спосіб оновлення сховищ з початкового коду.

- Скористайтеся [Обробники сповіщень](#) для інтеграції із більшістю типових служб зберігання коду
- Увімкніть оновлення вручну або у керуванні сховищем, або за допомогою [API](#) або [Клієнт Weblate](#)
- Увімкніть [AUTO_UPDATE](#) для автоматичного оновлення усіх складників у вашому екземплярі Weblate
- Виконайте `updategit` (із вибором проєкту або додаванням `-all` для оновлення усього)

Кожного разу, коли оновлюватиме Weblate, буде увімкнено додатки остаточної обробки, див. [Доповнення](#).

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 [API](#) — 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 сховищі 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. [API](#).

Автоматичне отримання змін з GitHub

Weblate постачається із вбудованою підтримкою GitHub.

Якщо ви користуєтесь Hosted Weblate, рекомендованим підходом є встановлення програми [Weblate](#). У такий спосіб ви отримаєте належну конфігурацію без зайвої метушні із налаштуванням. Нею також можна використати для записування змін назад до сховища.

Щоб отримувати сповіщення щодо кожного запису до сховища GitHub, додайте вебскрипт Weblate до параметрів сховища (*Webhooks*), як це показано на наведеному нижче зображенні:

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type

Secret

By default, we verify SSL certificates when delivering payloads. [Disable SSL verification](#)

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

☐ Let me select individual events.

☒ Active

We will deliver event details when this hook is triggered.

[Add webhook](#)

Щоб отримати адресу даних, допишіть `/hooks/github/` до адреси сайту вашого Weblate. Наприклад, для служби Hosted Weblate це `https://hosted.weblate.org/hooks/github/`.

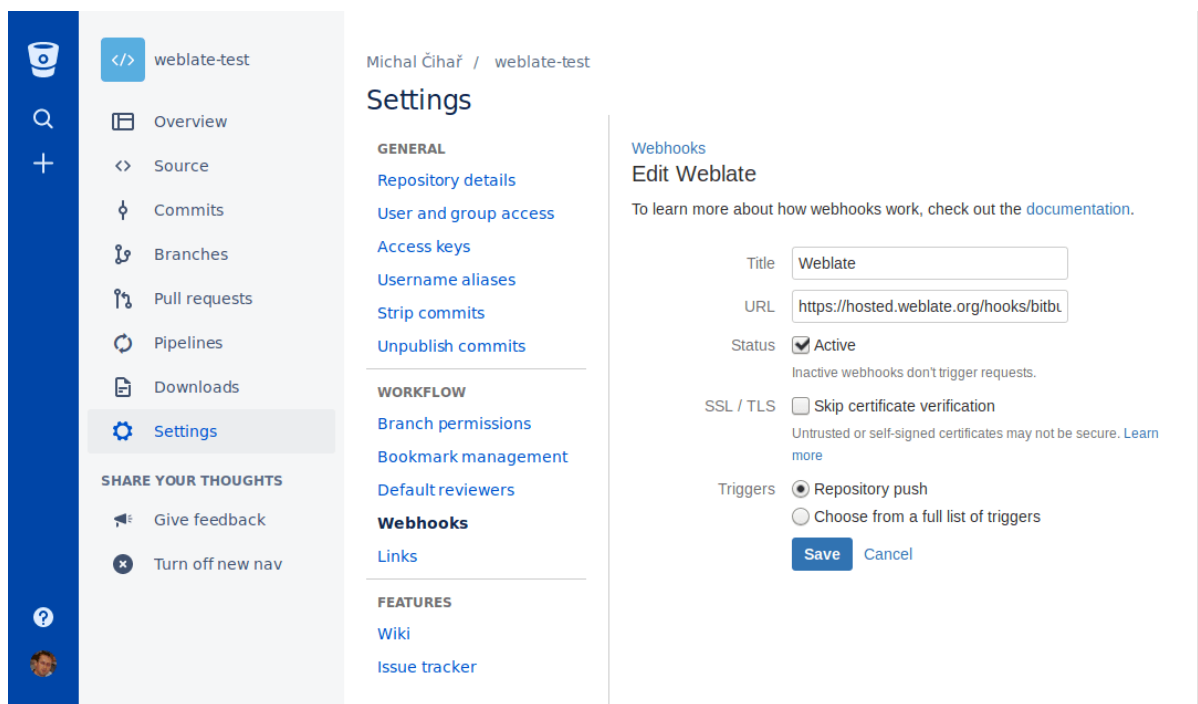
Ви можете лишити типові значення для решти параметрів (Weblate може одночасно обробляти обидва типи даних і споживатиметься лише подію *push*).

Дивись також:

POST /hooks/github/, Доступ до сховищ з Hosted Weblate

Автоматичне отримання змін з Bitbucket

У Weblate передбачено підтримку вебскриптів Bitbucket. Додайте вебскрипт, який вмикає запис до сховища із адресою призначення `/hooks/bitbucket/` у вашому встановленому Weblate (наприклад, `https://hosted.weblate.org/hooks/bitbucket/`).

**Дивись також:**

POST /hooks/bitbucket/, Доступ до сховищ 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 сховищ 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 Weblate interface for a project named 'nijel-test'. The top navigation bar includes 'Browse', 'Create', and a user profile icon. Below the project name, there are links for 'New Issue', 'Open PR', 'Fork', and 'Clone'. The main navigation menu on the left lists various settings categories, with 'Project Options' currently selected. The 'Project Options' section contains several checkboxes for enabling features like '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' (set to -1), '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', and 'Activate stomp notifications'. There is also a section for 'Activate Web-hooks' with a text input field containing 'https://hosted.weblate.org/hooks/pagure/' and buttons for 'Update' and 'Test web-hook'. A 'Learn more about' section lists links for 'Flags', 'Tracker read-only', 'Pull-request access only', 'Roadmap on Issue page', and 'fedmsg notifications'.

Дивись також:

POST /hooks/pagure/, Доступ до сховищ з Hosted Weblate

Автоматичне отримання змін зі сховищ Azure

Нове в версії 3.8.

У Weblate передбачено підтримку веб скриптів сховищ 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 сховищ з Hosted Weblate*

Автоматичне отримання змін зі сховищ 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* сховища.

Дивись також:

Вебскрипти у підручнику з Gitea, *POST /hooks/gitea/*, *Доступ до сховищ з Hosted Weblate*

Автоматичне отримання змін зі сховищ 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* сховища.

Дивись також:

Підручник щодо вебскриптів у Gitee, *POST /hooks/gitee/*, *Доступ до сховищ з Hosted Weblate*

Автоматичне щоденне оновлення сховищ

Weblate автоматично отримує вміст віддалених сховищ 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 сховища*). Якщо таку назву налаштовано, Weblate зможе записувати зміни до віддаленого сховища. 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 сховища* або за допомогою програмного інтерфейсу: *wlc push*.

Параметри запису до сховища є різним для різних *Налаштування інтеграції із керуванням версіями*. Подробиці можна знайти у цьому розділі.

Якщо вам не потрібне безпосереднє записування до сховища від Weblate, передбачено підтримку запитів щодо об'єднання *GitHub*, *GitLab*, *Pagure* або рецензувань *Gerrit*. Ви можете активувати їх вибором *GitHub*, *GitLab*, *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 сховища	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>GitHub</i>	<i>empty</i>	<i>empty</i>
Запит щодо об'єднання у GitHub з гілки	<i>GitHub</i>	SSH URL ¹	Назва гілки
Запит щодо злиття на GitLab з відгалуження	<i>GitLab</i>	<i>empty</i>	<i>empty</i>
Запит щодо злиття на GitLab з гілки	<i>GitLab</i>	SSH URL ¹	Назва гілки
Запит щодо злиття на Pagure з відгалуження	<i>Pagure</i>	<i>empty</i>	<i>empty</i>
Запит щодо злиття на Pagure з гілки	<i>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 сховищ*. 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 сховища.

Наприклад, у GitHub це можна зробити у налаштуваннях сховища:

¹ Може бути порожнім, якщо у *сховищі* 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.

Search for people or teams

People and teams that can dismiss reviews.

 **Organization and repository administrators**
These members can always dismiss.

 **weblate**
Weblate push user

×

2.9.3 Злиття чи перенесення

Типово, Weblate зливає основне сховище до власного. Цей найбезпечніший спосіб, якщо у вас є альтернативні способи доступу до базового сховища. Якщо це вам не потрібно, ви можете увімкнути перенесення (rebase) змін до основного сховища, що створюватиме журнал із меншою кількістю внесків об'єднання.

Примітка: Перенесення може призвести до проблем, якщо злиття є складним. Тому вам слід бути обережним із його вмиканням.

2.9.4 Взаємодія із іншими

Weblate спрощує взаємодію із іншими учасниками проєкту за допомогою програмного інтерфейсу.

Дивись також:

[API](#)

2.9.5 «Ліниві» 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 зі сховищем системи керування версіями, наприклад, для злиття (таке злиття типово увімкнено для групи *Managers*, див. [Керування доступом](#)).

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.
- 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 завдання зі створення внеску:

```
CELERY_BEAT_SCHEDULE = {
    # Unconditionally commit all changes every 2 minutes
    "commit": {
        "task": "weblate.trans.tasks.commit_pending",
        # Ommiting hours will honor per component settings,
        # otherwise components with no changes older than this
        # won't be committed
        "kwargs": {"hours": 0},
        # How frequently to execute the job in seconds
        "schedule": 120,
    }
}
```

2.9.6 Обробка сховища зі скриптами

Спосіб взаємодії Weblate зі сховищем коду можна змінити за допомогою [Доповнення](#). Зверніться до розділу [Виконання скриптів з додатка](#), щоб дізнатися про те, як виконувати зовнішні скрипти з додатків.

2.9.7 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 поширення перекладів (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 сховища за допомогою додатка (див. *Автоматичний переклад*).

2.10 Ліцензування перекладів

Ви можете вказати, за якими умовами ліцензування будуть розповсюджуватися переклади. Це особливо важливо, якщо переклади є відкритими для усіх, 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), цю інформацію буде показано у розділі даних щодо перекладу відповідного *компонента*.

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:

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 / Language names' is visible, along with a 'translated 95%' indicator. A yellow banner contains the text 'Contribution to this translation requires you to agree with a contributor agreement.' and a 'View contributor agreement' button. The main content area has tabs for 'Translations', 'Info', 'Alerts', 'Search', 'Glossaries', 'Insights', 'Files', 'Tools', 'Manage', and 'Share'. The 'Translations' tab is active, showing a table with the following data:

Language	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Czech	✓					
Hebrew	✓					
Hungarian	81%	4	5			
English	✓					

At the bottom left, there is a 'Start new translation' button. At the bottom of the page, a footer reads 'Powered by Weblate 4.4' followed by links for 'About Weblate', 'Legal', 'Contact', 'Documentation', and '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:

Webblate Dashboard Projects Languages Checks

Your profile

Languages Preferences Notifications Account Profile Licenses Audit log API access

Licenses

Please pay attention to the licensing info, as this specifies how translations can be used.

By registering you agree to use your name and e-mail in the commits, and provide your contribution under the license defined by each localization project.

You have agreed to the following as a contributor:

- [WeblateOrg/Language names](#)

Licenses for individual translations

GNU General Public License v3.0 or later [GPL-3.0](#)

[WeblateOrg/Djangojs](#) [WeblateOrg/Django](#) [WeblateOrg/Language names](#)

MIT License [MIT](#)

[WeblateOrg/Android](#)

Powered by Weblate 4.4 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

2.11 Процес перекладу

2.11.1 Голосування за пропозицію

Everyone can add suggestions by default, to be accepted by signed in users. Suggestion voting can be used to make use of a string when more than one signed-in user agrees, by setting up the *Налаштовування складників* configuration with *Suggestion voting* to turn on voting, and *Autoaccept suggestions* to set a threshold for accepted suggestions (this includes a vote from the user making the suggestion if it is cast).

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. *Керування доступом*).

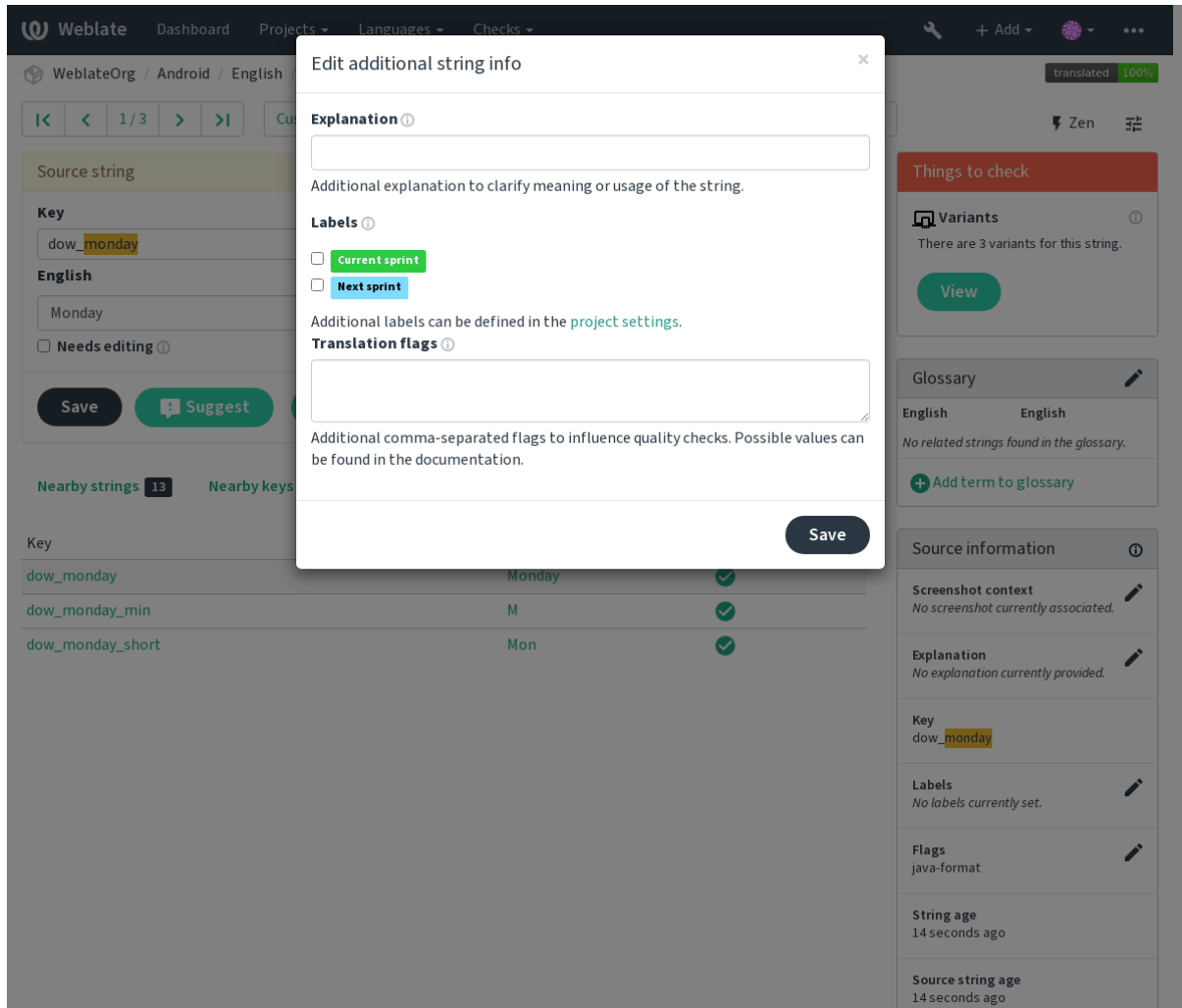
Ви можете поєднувати ці можливості за допомогою *прав доступу* у одну за допомогою таких конфігурацій:

- 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. - Встановіть 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, які будуть доступними у файлах, які перекладають. Ці дані включають пояснення, пріоритетність рядка, прапорці та візуальний контекст. Усі ці можливості можна налаштувати 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
Dashboard
Projects ▾
Languages ▾
Checks ▾
 + Add ▾ ⋮

WeblateOrg / Django / Czech / Translate
translated 96%

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

[Clone source](#)
 NBS ...

Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.

☐ Needs editing 169/1570

Save

Suggest

Skip

Nearby strings 26

Comments
Automatic suggestions
Other languages
History

Context	English	Czech	State
	Files	Soubory	✔
	Automatic translation	Automatický překlad	✔
	Add new translation string	Add new translation string	⚠
	Translation status	Stav překladu	✔
	%{count}s word	%{count}s slovo	✔
	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	Kontrola	✔
	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	WeblateOrg
project	projekt	WeblateOrg

+ Add term to glossary

Source information

Screenshot context

No screenshot currently associated.

Explanation

Help text for automatic translation tool

Labels

No labels currently set.

Flags

No flags currently set.

Source string location
weblate/templates/translation.html:212

String age
2 seconds ago

Source string age
3 seconds ago

Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 11

Powered by Weblate 4.4 · About Weblate · Legal · Contact · Documentation · Donate to Weblate
© 2023 Weblate.org

Встановлення пріоритетності рядків

Нове в версії 2.0.

String priority can be changed to offer higher priority strings for translation earlier by using the `priority` flag.

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.

Типовий набір прапорців перекладу визначається за *складниками* перекладу і файлом перекладу. Втім, ви можете скористатися ним для налаштування на рівні початкових рядків.

Дивись також:

Перевірки якості

Пояснення

Змінено в версії 4.1: У попередній версії це називалося додатковим контекстом.

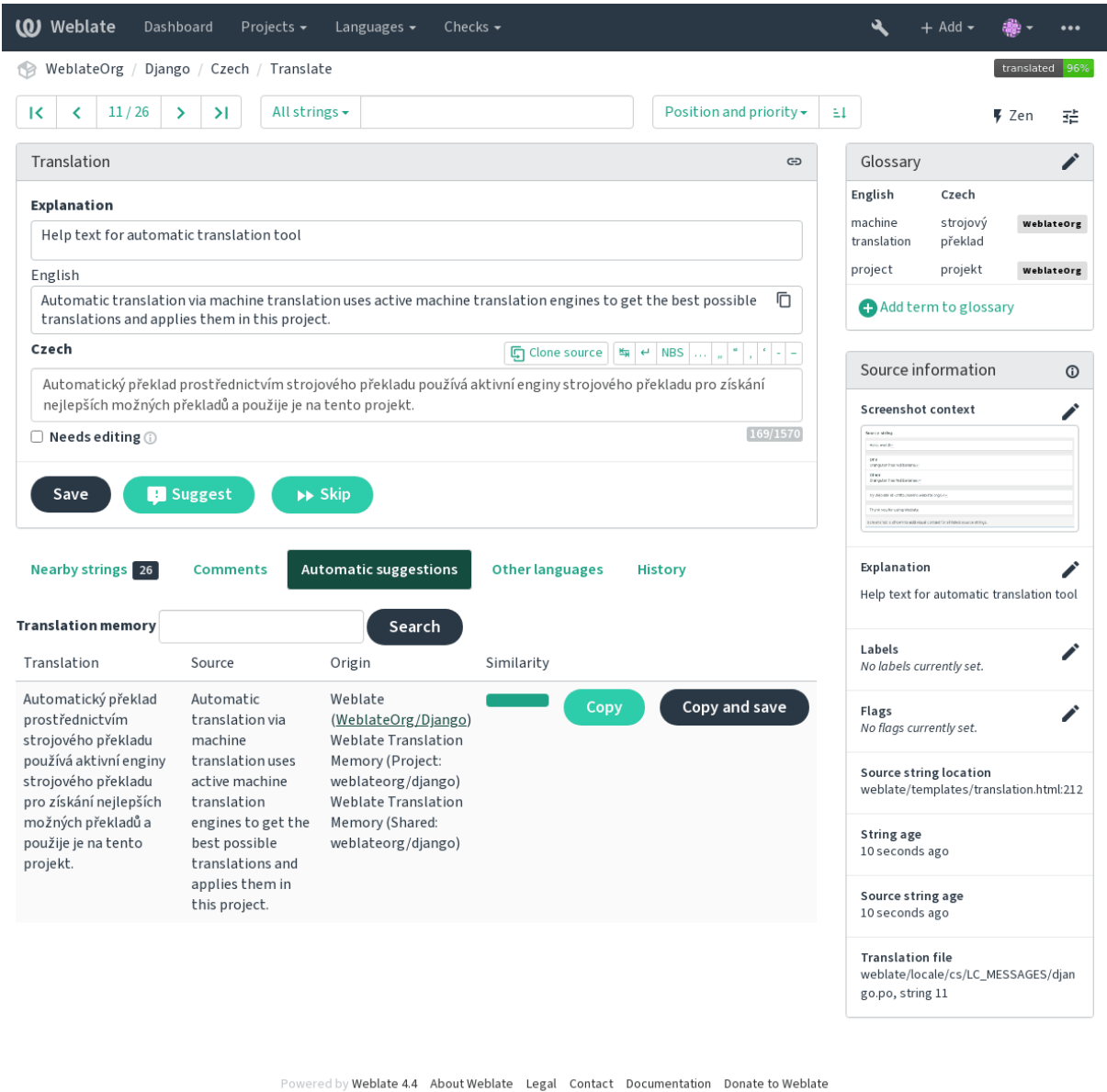
Скористайтесь поясненням для того, що зробити зрозумілішою область або використання перекладу. Ви можете скористатися кодом Markdown для включення посилань та визначення розмітки.

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:



Окрім *Додаткові відомості щодо початкових рядків*, знімки вікон маю окремий інтерфейс керування за допомогою меню *Знаряддя*. Вивантажуйте знімки вікон, пов'язуйте 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:

Web

late

Dashboard

Projects

Languages

Checks

+

Add

Web

lateOrg

/

Django

/

Screenshots

/

Automatic translation

Screenshot has been uploaded, you can now assign it to source strings.

Assigned source strings

Source string	Context	Location	Assigned screenshots	Actions
No source strings are currently assigned!				
Screenshot is shown to add visual context for all listed source strings.				

Assign source strings

Source string	Context	Location	Assigned screenshots	Actions
No new matching source strings found.				

Source string search

Search

Automatically recognize

Image

Source string

Hello, world!

One
Orangutan has %d banana.

Other
Orangutan 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	<div></div> testuser
Language	English

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

Delete

Powered by Weblate 4.4 About Weblate Legal Contact Documentation Donate to Weblate

2.11. Процес перекладу

255

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 © 2012 - 2020 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#

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. *Нетипові перевірки якості, додатки та автоматичні виправлення*.

2.12.2 Налаштовування поведінки

Ви можете скоригувати поведінку Weblate (здебільшого, перевірок) для кожного початкового рядка (під час рецензування початкових рядків, див. *Додаткові відомості щодо початкових рядків*) або на рівні *Налаштування 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:.*
```

Ось список прапорців, які приймаються у поточній версії:

rst-text Вважати текст документом reStructuredText, впливає на *Переклад не змінено*.

md-text Вважати текст документом Markdown.

dos-eol Використовує позначки кінців рядків DOS замість позначок кінців рядків Unix (`\r\n` замість `\n`).

url Рядок має складатися лише з адреси URL.

safe-html Рядок має бути безпечним кодом HTML, див. *Небезпечний HTML-код*.

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:HA3BA Визначає сімейство шрифтів для перевірок обробки, див. *Керування шрифтами*.

font-weight:ВАГА Визначити вагу шрифту для перевірок обробки, див. *Керування шрифтами*.

font-size:PO3MIP Визначити розмір шрифту для перевірок обробки, див. *Керування шрифтами*.

font-spacing: ІНТЕРВАЛ Визначити інтервал шрифту для перевірок обробки, див. *Керування шрифтами*.

placeholders:НАЗВА 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 заміників для забезпечення відповідності тексту, навіть за використання довгих імен. Приклад: **replacements:%s:"John Doe"**.

regex: ФОРМАЛЬНИЙ ВИРАЗ Формальний вираз для зіставлення файлів перекладу, див. *Регулярний вираз*.

python-format, c-format, php-format, python-brace-format, javascript-format, c-sharp-format, ja

Обробляти усі рядки як рядки форматування, впливає на Форматовані рядки, Форматовані рядки, Форматовані рядки, Форматовані рядки, Форматовані рядки, Форматовані рядки, Форматовані ряд- ки, Форматовані рядки, Форматовані рядки, Форматовані рядки, Форматовані рядки, Форматовані рядки, Переклад не змінено.

strict-same Наказати перевіріці «Незмінений переклад» уникати використання «чорного» списку вбудованих слів, див. *Переклад не змінено*.

ignore-bbcode Пропустити перевірку «Розмітка BBcode».

ignore-duplicate Пропустити перевірку якості «Послідовне дублювання слів».

ignore-double-space Пропустити перевірку якості «Подвійний пробіл».

ignore-angularjs-format Пропускати перевірку якості «Рядок інтерполяції AngularJS».

ignore-c-format Пропустити перевірку якості «Форматування C».

ignore-c-sharp-format Пропустити перевірку якості «Форматування C#».

ignore-es-format Пропустити перевірку якості «Літерали шаблонів ECMAScript».

ignore-i18next-interpolation Пропускати перевірку якості «Інтерполяція i18next».

ignore-java-format Пропустити перевірку якості «Форматування Java».

ignore-java-messageformat Пропустити перевірку якості «Форматування повідомлень Java».

ignore-javascript-format Пропустити перевірку якості «Форматування JavaScript».

ignore-percent-placeholders Пропустити перевірку якості «Замінники із символом відсотків».

ignore-perl-format Пропустити перевірку якості «Форматування Perl».

ignore-php-format Пропустити перевірку якості «Форматування PHP».

ignore-python-brace-format Пропустити перевірку якості «Форматування дужок Python».

ignore-python-format Пропустити перевірку якості «Форматування Python».

ignore-qt-format Пропустити перевірку якості «Форматування Qt».

ignore-qt-plural-format Пропустити перевірку якості «Форматування форм множини Qt».

ignore-ruby-format Пропустити перевірку якості «Форматування Ruby».

ignore-vue-format Пропустити перевірку якості «Форматування I18n Vue».

ignore-translated Пропустити перевірку якості «Було перекладено».

ignore-inconsistent Пропустити перевірку якості «Несумісність».

ignore-kashida Пропустити перевірку якості «Використання літери кашида».

ignore-md-link Пропустити перевірку якості «Посилання Markdown».

ignore-md-reflink Пропустити перевірку якості «Внутрішні посилання Markdown».

ignore-md-syntax Пропустити перевірку якості «Синтаксис Markdown».

ignore-max-length Пропустити перевірку якості «Максимальна довжина перекладу».

ignore-max-size Пропустити перевірку якості «Максимальний розмір перекладу».

ignore-escaped-newline Пропустити перевірку якості «Невідповідність n».

ignore-end-colon Пропустити перевірку якості «Невідповідність двокрапки».

ignore-end-ellipsis Пропустити перевірку якості «Невідповідність багатокрапки».

ignore-end-exclamation Пропустити перевірку якості «Невідповідність знаків оклику».

ignore-end-stop Пропустити перевірку якості «Невідповідність кінцевих крапок».

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 «Невідповідність розривів рядків».

ignore-plurals Пропустити перевірку якості «Пропущено форми множини».

ignore-placeholders Пропустити перевірку якості «Замінники».

ignore-punctuation-spacing Пропустити перевірку якості «Інтервали при пунктуації».

ignore-regex Пропустити перевірку якості «Формальний вираз».

ignore-same-plurals Пропустити перевірку якості «Однакові форми множини».

ignore-begin-newline Пропустити перевірку якості «Початковий символ розриву рядка».

ignore-begin-space Пропустити перевірку якості «Початковий пробіл».

ignore-end-newline Пропустити перевірку якості «Кінцева символ розриву рядка».

ignore-end-space Пропустити перевірку якості «Кінцевий пробіл».

ignore-same Пропустити перевірку якості «Незмінений переклад».

ignore-safe-html Пропустити перевірку якості «Небезпечний HTML».

ignore-url Пропустити перевірку якості «Адреса URL».

ignore-xml-tags Пропустити перевірку «Розмітка XML».

ignore-xml-invalid Пропустити перевірку якості «Синтаксис XML».

ignore-zero-width-space Пропустити перевірку якості «Пробіл нульової ширини».

ignore-ellipsis Пропустити перевірку якості «Багатокрапка».

ignore-long-untranslated Пропустити перевірку якості «Довге неперекладене».

ignore-multiple-failures Пропустити перевірку якості «Кратні невдалі перевірки».

ignore-unnamed-format Пропустити перевірку якості «Кратні змінні 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 *Налаштовування 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 *Налаштовування 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*).

2.12.4 Керування шрифтами

Нове в версії 3.7.

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. *Налаштовування поведінки*).

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
+ Add

WeblateOrg / Font groups / default-font

Font group

Name	default-font		
Default font	Source Sans Pro 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 Pro Bold

Default font is used unless per language override matches.

Save

Powered by Weblate 4.4 [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 Pro 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 4.4 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Гарнітура шрифту та стиль автоматично визначаються після вивантаження шрифтів:

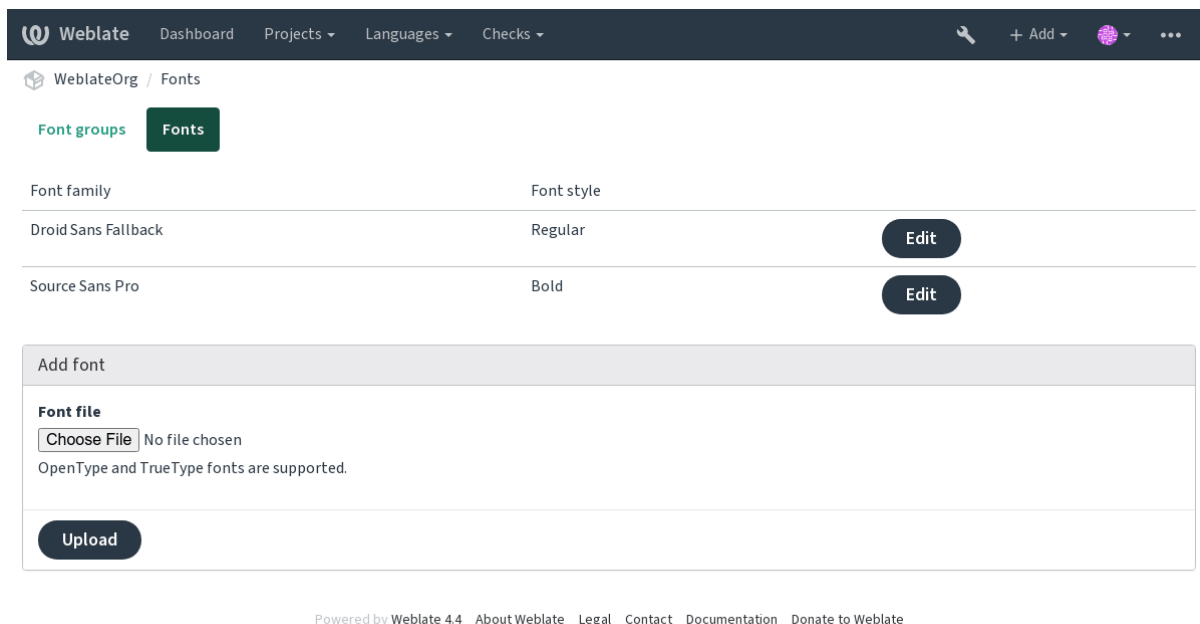
Weblate
 Dashboard Projects Languages Checks

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 4.4 [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](#)). Вам, ймовірно, потрібні такі:

max-size:500 Визначає максимальну ширину.

font-family:ubuntu Визначає групу шрифтів для використання її визначенням ідентифікатора.

font-size:22 Визначає розмір шрифту.

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. [Немуніві 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 © 2012 – 2020 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
```

(continues on next page)

(continued from previous page)

```
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""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 © 2012 - 2020 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""Quality check example for Czech plurals."""

from django.utils.translation import gettext_lazy as _

from weblate.checks.base import TargetCheck
```

(continues on next page)

```

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

    def check_single(self, source, target, unit):
        """We don't check target strings here."""
        return False

```

2.13 Машинний переклад

Передбачено вбудовану підтримку декількох служб комп'ютерного перекладу. Цю підтримку може бути увімкнено адміністратором за допомогою параметра `MT_SERVICES` для кожної із служб. Переклади регламентуються відповідними умовами використання, тому переконайтеся, що ви зможете використовувати їх у бажаний для вас спосіб.

Початкову мову можна налаштувати у розділі [Налаштування проєкту](#).

2.13.1 amaGama

Особливий екземпляр `tmserver`, яким керують розробники Virtaal.

Увімкніть цю службу додаванням `weblate.machinery.tmserver.AmagamaTranslation` до `MT_SERVICES`.

Дивись також:

[Installing amaGama, Amagama, пам'ять перекладів amaGama](#)

2.13.2 Apertium

Платформа перекладу вільного програмного забезпечення, яка забезпечує переклади обмеженим набором мов.

Рекомендованим способом використання Apertium є запуск вашого власного сервера Apertium-APY.

Увімкніть цю службу додаванням `weblate.machinery.apertium.ApertiumAPYTranslation` до `MT_SERVICES` і встановіть `MT_APERTIUM_APY`.

Дивись також:

[MT_APERTIUM_APY](#), сайт Apertium, Документація з Apertium APY

2.13.3 AWS

Нове в версії 3.1.

Amazon Translate — служба перекладу на основі нейронних машин для перекладу тексту англійською мовою і з англійської мови для декількох підтримуваних мов.

1. Turn on this service by adding `weblate.machinery.aws.AWSTranslation` to `MT_SERVICES`.
2. Встановіть модуль `boto3`.
3. Налаштуйте Weblate.

Дивись також:

`MT_AWS_REGION`, `MT_AWS_ACCESS_KEY_ID`, `MT_AWS_SECRET_ACCESS_KEY` Документація до Amazon Translate

2.13.4 Програмний інтерфейс комп'ютерного перекладу Baidu

Нове в версії 3.2.

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.

Увімкніть цю службу додаванням запису `weblate.machinery.baidu.BaiduTranslation` до `MT_SERVICES` і встановленням `MT_BAIDU_ID` та `MT_BAIDU_SECRET`.

Дивись також:

`MT_BAIDU_ID`, `MT_BAIDU_SECRET` Програмний інтерфейс перекладу Baidu

2.13.5 DeepL

Нове в версії 2.20.

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)*.

Увімкніть цю службу додаванням `weblate.machinery.deepl.DeepLTranslation` до `MT_SERVICES` і встановіть `MT_DEEPL_KEY`.

Підказка: Якщо у вас є передплата на інструменти комп'ютерного перекладу, буде зроблено припущення, що ви користуєтеся «v1 API», а не типовим «v2» у Weblate (це, насправді, не версія програмного інтерфейсу у цьому випадку). Ви можете переключити версію програмного інтерфейсу за допомогою параметра `MT_DEEPL_API_VERSION`.

Дивись також:

`MT_DEEPL_KEY`, `MT_DEEPL_API_VERSION`, DeepL website, Ціни DeepL, Документація з програмного інтерфейсу DeepL

2.13.6 Glosbe

Вільний словник та пам'ять перекладів для майже будь-якої із актуальних мов.

Доступ до програмного інтерфейсу є безкоштовним, але його прив'язано до використаного ліцензування початкових даних. Кількість викликів з однієї адреси IP обмежено протягом встановленого періоду часу для запобігання зловмисному використанню.

Увімкніть цю службу додаванням `weblate.machinery.glosbe.GlosbeTranslation` до `MT_SERVICES`.

Дивись також:

[Сайт Glosbe](#)

2.13.7 Перекладач Google

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.

Щоб увімкнути цю службу, додайте `weblate.machinery.google.GoogleTranslation` до `MT_SERVICES` і встановіть `MT_GOOGLE_KEY`.

Дивись також:

`MT_GOOGLE_KEY`, [Документація з Перекладача Google](#)

2.13.8 Google Translate API V3 (розширений)

Служба комп'ютерного перекладу, яка надається хмарними службами Google.

Ця служба відрізняється від попередньої тим, що використовує інший спосіб розпізнавання. Щоб увімкнути службу, додайте `weblate.machinery.googlev3.GoogleV3Translation` до `MT_SERVICES` і встановіть

- `MT_GOOGLE_CREDENTIALS`
- `MT_GOOGLE_PROJECT`

Якщо не працюватиме `location`, вам, ймовірно, також слід вказати `MT_GOOGLE_LOCATION`.

Дивись також:

`MT_GOOGLE_CREDENTIALS`, `MT_GOOGLE_PROJECT`, `MT_GOOGLE_LOCATION` [Документація з програмного інтерфейсу Google](#)

2.13.9 Когнітивні служби перекладу Microsoft

Нове в версії 2.10.

Служба комп'ютерного перекладу, яка надається Microsoft на порталі Azure як одна з когнітивних служб.

У Weblate реалізовано програмний інтерфейс перекладача версії 3.

Щоб увімкнути цю службу, додайте `weblate.machinery.microsoft.MicrosoftCognitiveTranslation` до `MT_SERVICES` і встановіть `MT_MICROSOFT_COGNITIVE_KEY`.

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 `MT_MICROSOFT_REGION` у значення локалі вашої служби.

Дивись також:

`MT_MICROSOFT_COGNITIVE_KEY`, `MT_MICROSOFT_REGION`, Когнітивні служби — програмний інтерфейс текстового перекладу, Портал Microsoft Azure

2.13.10 Служба термінології Microsoft

Нове в версії 2.19.

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.

Увімкніть цю службу додаванням `weblate.machinery.microsoftterminology.MicrosoftTerminologyService` до `MT_SERVICES`.

Дивись також:

Програмний інтерфейс служби термінології Microsoft

2.13.11 ModernMT

Нове в версії 4.2.

Увімкніть цю службу додаванням `weblate.machinery.modernmt.ModernMTTranslation` до `MT_SERVICES` і налаштуйте `MT_MODERNMT_KEY`.

Дивись також:

Програмний інтерфейс ModernMT, `MT_MODERNMT_KEY`, `MT_MODERNMT_URL`

2.13.12 MyMemory

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 `MT_MYMEMORY_EMAIL`. Ви також можете попросити власників служби збільшити кількість запитів.

Увімкніть цю службу додаванням `weblate.machinery.mymemory.MyMemoryTranslation` до `MT_SERVICES` і встановіть `MT_MYMEMORY_EMAIL`.

Дивись також:

`MT_MYMEMORY_EMAIL`, `MT_MYMEMORY_USER`, `MT_MYMEMORY_KEY`, сайт MyMemory

2.13.13 Програмний інтерфейс комп'ютерного перекладу NetEase Sight

Нове в версії 3.3.

Служба комп'ютерного перекладу, яка надається NetEase.

Ця служба використовує програмний інтерфейс — вам слід отримати ключ і пароль від NetEase.

Увімкніть цю службу додаванням `weblate.machinery.youdao.NeteaseSightTranslation` до `MT_SERVICES` і встановіть `MT_NETEASE_KEY` і `MT_NETEASE_SECRET`.

Дивись також:

`MT_NETEASE_KEY`, `MT_NETEASE_SECRET` Платформа перекладу Netease Sight

2.13.14 tmserver

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:
2. Turn on this service by adding `weblate.machinery.tmserver.TMServerTranslation` to `MT_SERVICES`.

```
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
```

3. Запустіть tmserver для очікування на ваші запити:

```
tmserver -d /var/lib/tm/db
```

4. Налаштуйте Weblate на обмін даними зі службою:

```
MT_TMSERVER = "http://localhost:8888/tmserver/"
```

Дивись також:

`MT_TMSERVER`, tmserver Installing amaGama, Amagama, пам'ять перекладів Amagama

2.13.15 Переклад Yandex

Служба комп'ютерного перекладу, яка надається Yandex.

Ця служба використовує програмний інтерфейс перекладу — вам слід отримати ключ до програмного інтерфейсу від Yandex.

Увімкніть цю службу додаванням `weblate.machinery.yandex.YandexTranslation` до `MT_SERVICES` і встановіть `MT_YANDEX_KEY`.

Дивись також:

`MT_YANDEX_KEY`, Програмний інтерфейс перекладу Yandex, Yandex.Translate

2.13.16 Програмний інтерфейс комп'ютерного перекладу Youdao Zhiyun

Нове в версії 3.2.

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.

Увімкніть цю службу додаванням `weblate.machinery.youdao.YoudaoTranslation` до `MT_SERVICES` і встановіть `MT_YOUDAO_ID` і `MT_YOUDAO_SECRET`.

Дивись також:

`MT_YOUDAO_ID`, `MT_YOUDAO_SECRET` Служба перекладу природними мовами Youdao Zhiyun

2.13.17 Weblate

Weblate також може бути джерелом для комп'ютерного перекладу. Його засновано на повнотекстовому рушії Woosh, який здатен надавати точні і неточні відповідники.

Увімкніть ці служби додаванням `weblate.machinery.weblatetm.WeblateTranslation` до `MT_SERVICES`.

2.13.18 Пам'ять перекладів Weblate

Нове в версії 2.20.

Ви також можете скористатися *Пам'ять перекладів* як джерелом для пропозицій комп'ютерного перекладу.

Увімкніть ці служби додаванням `weblate.memory.machine.WeblateMemory` до `MT_SERVICES`. Типово, цю службу увімкнено.

2.13.19 Концентратор перекладів SAP

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.

Увімкніть цю службу додаванням `weblate.machinery.saptranslationhub.SAPTranslationHub` до `MT_SERVICES` і встановіть відповідний доступ або до програмного інтерфейсу пісочниці, або до промислового програмного інтерфейсу.

Примітка: Щоб отримати доступ до програмного інтерфейсу пісочниці, вам слід встановити `MT_SAP_BASE_URL` і `MT_SAP_SANDBOX_APIKEY`.

Щоб отримати доступ до програмного інтерфейсу, вам слід встановити `MT_SAP_BASE_URL`, `MT_SAP_USERNAME` і `MT_SAP_PASSWORD`.

Дивись також:

`MT_SAP_BASE_URL`, `MT_SAP_SANDBOX_APIKEY`, `MT_SAP_USERNAME`, `MT_SAP_PASSWORD`, `MT_SAP_USE_MT` Програмний інтерфейс вузла перекладу SAP

2.13.20 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 © 2012 - 2020 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""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 `MT_SERVICES` і Weblate розпочне використовувати його.

2.14 Доповнення

Нове в версії 2.19.

Додатки надають змогу налаштувати процедуру перекладу. Їх можна встановити за допомогою панелі складника перекладу, вони працюють за лаштунками. Керувати додатками можна за допомогою меню *Керувати* ↓ *Додатки* для кожного відповідного складника перекладу для адміністраторів.

Installed addons ⓘ

There are no addons currently installed.

Available addons ⓘ

⚙️ Automatic translation ⓘ		Install
⚙️ Language consistency ⓘ	project wide	Install
🔍 Component discovery ⓘ	repository wide	Install
⚙️ Bulk edit ⓘ		Install
📊 Statistics generator ⓘ		Install
⚙️ Contributors in comment ⓘ		Install
🔧 Customize gettext output ⓘ		Install
⚙️ Generate MO files ⓘ		Install
⚙️ Update PO files to match POT (msgmerge) ⓘ		Install
✂️ Squash Git commits ⓘ	repository wide	Install
🗑️ Stale comment removal ⓘ	project wide	Install
🗑️ Stale suggestion removal ⓘ	project wide	Install

Some addons will ask for additional configuration during installation.

Powered by Weblate 4.4 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

2.14.1 Вбудовані додатки

Автоматичний переклад

Нове в версії 3.9.

Автоматично перекладає рядки за допомогою машинного перекладу або інших складників.

Цей додаток автоматично вмикається, коли у складнику з'являються нові рядки.

Дивись також:

Автоматичний переклад, Підтримання синхронізації перекладів у різних складниках

CDN локалізації JavaScript

Нове в версії 4.2.

Додає CDN локалізації для JavaScript або HTML.

Ним можна скористатися для локалізації статичних сторінок HTML або завантаження локалізації у коді JavaScript.

Після встановлення додаток створює унікальну адресу для вашого складника, яку ви можете включати до документів HTML для налаштування 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.

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 порожніми рядками, а не повернено до джерельного рядка).

Дивись також:

Чи оновлює Weblate файли перекладів між перекладами?

Очищення файлів перекладу

Оновить 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?

Мовна однорідність

Гарантує, що всі складники в межах одного проєкту містять однаковий переклад однакових рядків.

Створює порожні переклади мовами, у яких є недодані складники.

Пошук пропущених мов відбувається кожні 24 години і при додаванні нової мови у Weblate.

На відміну від інших, цей додаток стосується усього проєкту.

Підказка: Автоматичний переклад нових доданих рядків за допомогою [Автоматичний переклад](#).

Виявлення складників

Автоматично додає або вилучає складники до проєкту 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.

Створіть один основний складник, ймовірність зникнення якого у майбутньому є найнижчою, а інші складники використовуватимуть *Внутрішні адреси Weblate* до нього як налаштування системи керування версіями. Налаштуйте систему керування версіями на пошук усіх складників у ній.

Відповідність встановлюватиметься за формальними виразами, перевагою яких є потужність, а недоліком — складність у визначенні. 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

+ Add

WeblateOrg

/

Language names

/

Addons

/

Component discovery

Configure addon

Please review and confirm the matched components.

Component

Matched files

Following components would be created

Djangojs

weblate/locale/cs/LC_MESSAGES/djangojs.po (cs)
weblate/locale/hu/LC_MESSAGES/djangojs.po (hu)
weblate/locale/he/LC_MESSAGES/djangojs.po (he)

Django

weblate/locale/hu/LC_MESSAGES/django.po (hu)
weblate/locale/he/LC_MESSAGES/django.po (he)
weblate/locale/cs/LC_MESSAGES/django.po (cs)

☐ 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.

Language filter

^(cs|he|hu)\$

Regular expression to filter translation files against when scanning for filemask.

☒ Clone addons from the main component to the newly created ones

☐ Remove components for inexistant 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>	cs/application.po cs/website.po de/application.po de/website.po	One folder per language containing translation files for components.
<code>locale/(?P<language>[^\s]*)/LC_MESSAGES/(?P<component>[^\s]*)\.po</code>	locale/cs/LC_MESSAGES/application.po locale/cs/LC_MESSAGES/website.po locale/de/LC_MESSAGES/application.po locale/de/LC_MESSAGES/website.po	Usual structure for storing gettext PO files.
<code>src/locale/(?P<component>[^\s]*)\. (?P<language>[^\s]*)\.po</code>	src/locale/application.cs.po src/locale/website.cs.po src/locale/application.de.po src/locale/website.de.po	Using both component and language name within filename.
<code>locale/(?P<language>[^\s]*)/(?P<component>[^\s]*)/(?P<language>[^\s]*)\.po</code>	locale/cs/application/cs.po locale/cs/website/cs.po locale/de/application/de.po locale/de/website/de.po	Using language in both path and filename.
<code>res/values-(?P<language>[^\s]*)/strings-(?P<component>[^\s]*)\.xml</code>	res/values-cs/strings-about.xml res/values-cs/strings-help.xml res/values-de/strings-about.xml res/values-de/strings-help.xml	Android resource strings, split into several files.

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

Дивись також:*Розмітка шаблонів***Масове редагування**

Нове в версії 3.11.

Надає можливість пакетного редагування прапорців, міток або стану.

Автоматизація встановлення міток для нових рядків може бути корисним (почніть з пошукового запиту `NOT has:label` і додавайте бажані мітки, аж доки для усіх рядків буде встановлено мітки). Ви також можете виконувати будь-які інші автоматизовані дії для метаданих Weblate.

Приклади:

Table 1: Автоматично додавати мітку до нових рядків

Запит щодо пошуку	<code>NOT has:label</code>
Мітки для додавання	<code>recent</code>

Table 2: Позначення усіх записів журналу змін Файли метаданих
кратниці застосунків як придатних лише для читання

Запит щодо пошуку	<code>language:en AND key:changelogs/</code>
Прапорці перекладу для додавання	<code>read-only</code>

Дивись також:*Масове редагування***Позначити незмінені переклади як такі, що потребують редагування**

Нове в версії 3.1.

Кожного разу, коли новий перекладний рядок імпортується з VCS, і він збігається із початковим рядком, його буде позначено як такий, що потребує редагування в Weblate. Це корисно для форматів файлів, які включають усі рядки, в тому числі й такі, що не перекладаються.

Позначити нові джерельні рядки як „потребують редагування“

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.

Позначити нові переклади як такі, що потребують редагування

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.

Генератор статистики

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

Перекладачі в коментарі

Оновлює коментар у заголовку файла PO, щоб включити імена перекладачів та роки перекладу.

Заголовок файла PO міститиме список учасників і роки внесків:

```
# Michal Čihař <michal@cihar.com>, 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»

Оновлює змінну ALL_LINGUAS у `configure`, `configure.in` або будь-яких файлах `configure.ac` при додаванні нового перекладу.

Налаштувати виведення gettext

Уможливорює налаштування виведення gettext, наприклад, перенесення рядків.

Пропонує такі варіанти дій:

- Перенесення рядків на 77 символів на символах перенесення рядків
- Перенесення рядків лише після символу нового рядку
- Без перенесення рядків

Примітка: Типово, gettext переносить рядки на 77 символів і на символах перенесення рядків. Якщо використано параметр `--no-wrap`, перенесення відбуватиметься лише на символах перенесення рядків.

Оновити файл LINGUAS

Оновлює файл LINGUAS, коли додано новий переклад.

Генерувати мо-файли

Автоматично створює мо-файл для кожного зміненого ро-файла.

The location of the generated MO file can be customized and the field for it uses *Розмітка шаблонів*.

Оновити файли po, порівнявши із pot (msgmerge)

Updates all PO files (as configured by *Маска файлів*) to match the POT file (as configured by *Заготовка для нових перекладів*) using **msgmerge**.

Цей додаток спрацьовує щоразу, коли нові зміни витягуються із основного сховища 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

Об'єднати Git подання перед відправленням змін.

Ви можете вибрати один з таких режимів:

Нове в версії 3.4.

- Всі подання в одне
- Для мови
- Для файлу

Нове в версії 3.5.

- 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.

Нове в версії 4.1.

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

Дозволяє налагодити виведення JSON, наприклад відступи та упорядкування.

Форматує файл властивостей Java

Впорядковує файл властивостей Java.

Видалення застарілого коментаря

Нове в версії 3.7.

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.

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.

Оновити усі файли перекладу, щоб вони відповідали одномовному основному файлу. Невикористані рядки вилучаються, а нові додаються як копії початкового рядка.

Підказка: Скористайтесь [Очищення файлів перекладу](#), якщо ви хочете лише вилучити застарілі переклади.

Дивись також:

Чи оновлює Weblate файли перекладів між перекладами?

Налагодити виведення YAML

Нове в версії 3.10.2.

Дозволяє налаштовувати 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 створити власні додатки — усе, що потрібно, створити підклас `weblate.addons.base.BaseAddon`, 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 © 2012 - 2020 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""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"
    # Verbose name and long description
    verbose = _("Execute script before commit")
    description = _("This addon 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 сховища системи керування версіями для будь-якого заданого компонента.

Крім того, можна скористатися такими змінними середовища:

WL_VCS

Використана система керування версіями.

WL_REPO

Адреса основного сховища із кодом.

WL_PATH

Абсолютний шлях до сховища системи керування версіями.

WL_BRANCH

Нове в версії 2.11.

Гілка сховища, 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 проектом.

Дивись також:

Налаштування складників

Обробка сховища після оновлення

Обробкою сховища після оновлення можна скористатися для оновлення файлів перекладу, коли міняється джерело у основному сховищі системи керування версіями. Слід пам'ятати, що Weblate «бачить» лише файли, які внесено до системи керування версіями, тому вам доведеться внести зміни до сховища 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 сховища

Скористайтеся скриптом внесення даних для автоматичного внесення змін до перекладу перед тим, як його буде внесено до сховища.

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 усіх 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 є доступною для усіх 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 таємної інформації.

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:

- Облікові записи. Вилучає пароль до облікового запису, забороняючи користувачеві входити до системи без запиту щодо введення нового пароля.
- Скидання паролів. 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 сховища щоденно.

Підказка: Корисно, якщо ви не використовуєте *Обробники сповіщень* для автоматичного оновлення сховищ 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 сховища.

"full" також True Оновити віддалені сховища 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 якої можна використовувати один і той самий пароль.

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 багатокрапка (...).

`weblate.trans.autofixes.chars.RemoveZeroSpace` Вилучає пробільні символи нульової ширини, якщо у початковому коді містяться ці символи.

`weblate.trans.autofixes.chars.RemoveControlChars` Вилучає керівні символи, якщо у початковому коді містяться 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 BASE_DIR

Базовий каталог, у якому зберігаються початкові коди Weblate. Типово, використовується для визначення декількох інших похідних шляхів:

- `DATA_DIR`

Типове значення: каталог верхнього рівня для початкових кодів Weblate.

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 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 усіх цих змінних типових значенням є порожній список.

Приклад:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Дивись також:

Правила щодо безпеки даних, Content Security Policy (CSP)

2.16.13 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

2.16.14 COMMENT_CLEANUP_DAYS

Нове в версії 3.6.

Вилучати коментарі за вказану кількість днів. Типове значення — None, тобто не вилучати коментарі.

2.16.15 COMMIT_PENDING_HOURS

Нове в версії 2.10.

Кількість годин, протягом яких слід надіслати до сховища зміни з черги разом із фоновим завданням.

Дивись також:

Налаштовування складників, Вік змін для подання, Запуск завдань щодо супроводу, commit_pending

2.16.16 DATA_DIR

Тека, у якій Weblate зберігає дані. Тека містить посилання на сховища 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`.

media Типове розташування мультимедійних файлів Django, визначається на основі `MEDIA_ROOT`.

vcs Сховища систем керування версіями.

backups Дані щоденної резервної копії. Докладніше про це у розділі [Дампи даних для резервних копій](#).

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 — `$BASE_DIR/data`.

Дивись також:

`BASE_DIR`, Резервне копіювання і пересування Weblate

2.16.17 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

2.16.18 DEFAULT_ACCESS_CONTROL

Нове в версії 3.3.

Типовий параметр керування доступом для нових проєктів:

0 Відкритий

1 Захищений

100 Приватний

200 Нетиповий

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, Керування доступом

2.16.19 DEFAULT_RESTRICTED_COMPONENT

Нове в версії 4.1.

Типове значення для обмеження на доступ до складника.

Дивись також:

Керування доступом на рівні проєкту, Обмежений доступ, Керування доступом

2.16.20 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DEFAULT_DELETE_MESSAGE, DEFAULT_COMMIT_MESSAGE, DEFAULT_MERGE_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, об'єднання та повідомлення додатків

2.16.21 DEFAULT_ADDONS

Типові додатки, які слід встановлювати для усіх створених складників.

Примітка: Цей параметр стосується лише новостворених складників.

Приклад:

```
DEFAULT_ADDONS = {
    # Addon with no parameters
    "weblate.flags.target_edit": {},
    # Addon with parameters
    "weblate.autotranslate.autotranslate": {
        "mode": "suggest",
        "filter_type": "todo",
        "auto_source": "mt",
        "component": "",
        "engines": ["weblate-translation-memory"],
        "threshold": "80",
    },
}
```

Дивись також:

install_addon, WEBLATE_ADDONS

2.16.22 DEFAULT_COMMITER_EMAIL

Нове в версії 2.4.

Адреса електронної пошти автора внеску для створених складників перекладу. Типове значення — `noreply@weblate.org`.

Дивись також:

DEFAULT_COMMITER_NAME, Налаштовування складників, Ел. пошта подавача

2.16.23 DEFAULT_COMMITER_NAME

Нове в версії 2.4.

Ім'я автора внеску для створених складників перекладу. Типове значення — `Weblate`.

Дивись також:

DEFAULT_COMMITER_EMAIL, Налаштовування складників, Ім'я подавача

2.16.24 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

2.16.25 DEFAULT_MERGE_STYLE

Нове в версії 3.4.

Стиль злиття для усіх нових складників.

- *rebase* — типовий
- *merge*

Дивись також:

Налаштовування складників, Стиль злиття

2.16.26 DEFAULT_TRANSLATION_PROPAGATION

Нове в версії 2.5.

Типовий параметр для надсилання перекладу, типове значення — `True`.

Дивись також:

Налаштовування складників, Дозволити поширення перекладу

2.16.27 DEFAULT_PULL_MESSAGE

Заголовок для нових запитів щодо об'єднання, типове значення — 'Update from Weblate'.

2.16.28 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

2.16.29 ENABLE_HOOKS

Чи слід вмикати доступу до анонімних віддалених скриптів обробки.

Дивись також:

Обробники сповіщень

2.16.30 ENABLE_HTTPS

Визначає, надсилати посилання Weblate як HTTPS або як HTTP. Цей параметр впливає на надіслані повідомлення електронної пошти і створення абсолютних адрес.

Підказка: У типових налаштуваннях цей параметр також використовується для декількох параметрів Django, які пов'язано із HTTPS.

Дивись також:

SESSION_COOKIE_SECURE, CSRF_COOKIE_SECURE, SECURE_SSL_REDIRECT, *Встановіть належний домен сайту*

2.16.31 ENABLE_SHARING

Увімкнути або вимкнути меню *Поділитися*, щоб користувачі могли оприлюднювати поступ перекладу у соціальних мережах.

2.16.32 GITLAB_CREDENTIALS

Нове в версії 4.3.

Список реєстраційних даних для серверів GitLab.

Підказка: Скористайтесь цією змінною, якщо Weblate має взаємодіяти з кількома серверами. Для одинарної кінцевої точки GitLab скористайтесь *GITLAB_USERNAME* і *GITLAB_TOKEN*.

```
GITLAB_CREDENTIALS = {
    "gitlab.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitlab.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.33 GITLAB_USERNAME

Ім'я користувача GitLab, яке буде використано для надсилання запитів щодо об'єднання для оновлень перекладів.

Дивись також:

GITLAB_CREDENTIALS, *GitLab*

2.16.34 GITLAB_TOKEN

Нове в версії 4.3.

Особистий жетон доступу GitLab, який використовується для створення викликів програмного інтерфейсу для оновлень перекладу.

Дивись також:

GITLAB_CREDENTIALS, *GitLab*, GitLab: особистий жетон доступу

2.16.35 GITHUB_CREDENTIALS

Нове в версії 4.3.

Список реєстраційних даних для серверів GitHub.

Підказка: Скористайтесь цією змінною, якщо Weblate має взаємодіяти із декількома серверами. Для оди-
нарної кінцевої точки GitHub скористайтесь *GITHUB_USERNAME* і *GITHUB_TOKEN*.

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "github.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.36 GITHUB_USERNAME

Ім'я користувача GitHub, яке буде використано для надсилання запитів щодо об'єднання для оновлень перекладів.

Дивись також:

GITHUB_CREDENTIALS, *GitHub*

2.16.37 GITHUB_TOKEN

Нове в версії 4.3.

Особистий жетон доступу GitHub, який використовується для створення викликів програмного інтерфейсу для надсилання запитів щодо об'єднання для оновлень перекладу.

Дивись також:

GITHUB_CREDENTIALS, *GitHub*, Створення особистого жетона доступу

2.16.38 GOOGLE_ANALYTICS_ID

Ідентифікатор Google Analytics для вмикання спостереження за Weblate за допомогою Google Analytics.

2.16.39 HIDE_REPO_CREDENTIALS

Приховати реєстраційні дані сховища від показу у вебінтерфейсі. Якщо у адресі сховища 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.

2.16.40 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.

2.16.41 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

2.16.42 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

2.16.43 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 встановленої програми. Визначати його слід, лише якщо для визначення IP-адреси має бути використано довірені проксі-сервери.

Типове значення — `0`.

Дивись також:

Робота за реверсивним проксі-сервером, Обмеження частоти, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_HEADER

2.16.44 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 *Юридичні відомості*.

Приклад:

```
LEGAL_URL = "https://weblate.org/terms/"
```

2.16.45 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",
    ),
]
```

2.16.46 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

2.16.47 LICENSE_REQUIRED

Визначає, чи є обов'язковим атрибут ліцензування у записі *Налаштовування складників*.

Примітка: Типово вимкнено.

2.16.48 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Визначає, чи має бути обмежено довжину вказаного перекладу. Обмеження довжини — довжина початкового рядка * 10 символів.

Підказка: Встановіть значення `False`, щоб уможливити довші переклади (аж до 10000 символів), незалежно від довжини початкового рядка.

Примітка: Типове значення — `True`.

2.16.49 LOCALIZE_CDN_URL і LOCALIZE_CDN_PATH

Ці параметри призначено для налаштування додатка *CDN локалізації JavaScript*. `LOCALIZE_CDN_URL` визначає кореневу адресу, за якою доступним є CDN локалізації, а `LOCALIZE_CDN_PATH` визначає шлях, за яким Weblate має зберігати створені файли, які будуть обслуговуватися за адресою `LOCALIZE_CDN_URL`.

Підказка: На базовому Weblate використано `https://weblate-cdn.com/`.

Дивись також:

CDN локалізації JavaScript

2.16.50 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.

2.16.51 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
)
```

2.16.52 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`

2.16.53 MATOMO_URL

Повна адреса (разом із кінцевою похилою ризикою) 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`

2.16.54 MT_SERVICES

Змінено в версії 3.0: Параметр було перейменовано з `MACHINE_TRANSLATION_SERVICES` на `MT_SERVICES` для забезпечення сумісності назв із іншими параметрами комп'ютерного перекладу.

Список увімкнених служб комп'ютерного перекладу, якими слід користуватися.

Примітка: Багато зі служб потребують додаткових налаштувань, зокрема визначення ключів до програмного інтерфейсу. Будь ласка, ознайомтеся із відповідною документацією до *Машинний переклад*, щоб дізнатися більше.

```
MT_SERVICES = (  
    "weblate.machinery.apertium.ApertiumAPYTranslation",  
    "weblate.machinery.deepl.DeepLTranslation",  
    "weblate.machinery.glosbe.GlosbeTranslation",  
    "weblate.machinery.google.GoogleTranslation",  
    "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",  
    "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",  
    "weblate.machinery.mymemory.MyMemoryTranslation",  
    "weblate.machinery.tmserver.AmagamaTranslation",  
    "weblate.machinery.tmserver.TMServerTranslation",  
    "weblate.machinery.yandex.YandexTranslation",  
    "weblate.machinery.weblatetm.WeblateTranslation",  
    "weblate.machinery.saptranslationhub.SAPTranslationHub",  
    "weblate.memory.machine.WeblateMemory",  
)
```

Дивись також:

Машинний переклад, Автоматичні пропозиції

2.16.55 MT_APERTIUM_APY

Адреса сервера Apertium-APY, <https://wiki.apertium.org/wiki/Apertium-apy>

Дивись також:

Apertium, Машинний переклад, Автоматичні пропозиції

2.16.56 MT_AWS_ACCESS_KEY_ID

Ідентифікатор ключа доступу для Amazon Translate.

Дивись також:

AWS, Машинний переклад, Автоматичні пропозиції

2.16.57 MT_AWS_SECRET_ACCESS_KEY

Реєстраційний ключ до програмного інтерфейсу Amazon Translate.

Дивись також:

AWS, Машинний переклад, Автоматичні пропозиції

2.16.58 MT_AWS_REGION

Назва регіону, якою слід скористатися у Amazon Translate.

Дивись також:

AWS, Машинний переклад, Автоматичні пропозиції

2.16.59 MT_Baidu_ID

Ідентифікатор клієнта програмного інтерфейсу Baidu Zhiyun. Ви можете зареєструватися на <https://api.fanyi.baidu.com/api/trans/product/index>

Дивись також:

Програмний інтерфейс комп'ютерного перекладу Baidu, Машинний переклад, Автоматичні пропозиції

2.16.60 MT_Baidu_SECRET

Ключ клієнта до програмного інтерфейсу Baidu Zhiyun. Ви можете зареєструватися на <https://api.fanyi.baidu.com/api/trans/product/index>

Дивись також:

Програмний інтерфейс комп'ютерного перекладу Baidu, Машинний переклад, Автоматичні пропозиції

2.16.61 MT_DEEPL_API_VERSION

Нове в версії 4.1.1.

Версія програмного інтерфейсу для використання у службі DeepL. Версія обмежує область використання:

v1 Призначено для інструментів комп'ютерного перекладу. Придатна до використання із передплатою на основі користувачів.

v2 Призначено для використання програмного інтерфейсу. Передплата є заснованою на використанні.

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.

Дивись також:

DeepL, Машинний переклад, Автоматичні пропозиції

2.16.62 MT_DEEPL_KEY

Ключ до програмного інтерфейсу для програмного інтерфейсу DeepL. Ви можете зареєструвати такий на <https://www.deepl.com/pro.html>

Дивись також:

DeepL, Машинний переклад, Автоматичні пропозиції

2.16.63 MT_GOOGLE_KEY

Ключ до програмного інтерфейсу Перекладача Google, версія 2. Ви можете зареєструвати такий ключ на сторінці <https://cloud.google.com/translate/docs>

Дивись також:

Перекладач Google, Машинний переклад, Автоматичні пропозиції

2.16.64 MT_GOOGLE_CREDENTIALS

Файл реєстраційних даних JSON програмного інтерфейсу версії 3, отримати у консолі «хмари» Google. Вам слід вказати повний шлях в операційній системі. Реєстраційні дані прив'язуються до окремої служби і облікового запису, який пов'язано із певним проектом. Будь ласка, ознайомтеся із вмістом <https://cloud.google.com/docs/authentication/getting-started>, щоб дізнатися більше.

2.16.65 MT_GOOGLE_PROJECT

Ідентифікатор проекту «хмари» програмного інтерфейсу версії 3 Google із активованою службою перекладу та обліком. Будь ласка, ознайомтеся із вмістом сторінки <https://cloud.google.com/appengine/docs/standard/nodejs/building-app/creating-project>, щоб дізнатися більше

2.16.66 MT_GOOGLE_LOCATION

Рушій програм «хмари» програмного інтерфейсу версії 3 Google може бути специфічним до вашого місця розташування. Змініть значення відповідним чином, якщо типовий резервний варіант `global` вас не влаштовує.

Докладніші відомості можна знайти на <https://cloud.google.com/appengine/docs/locations>

Дивись також:

Google Translate API V3 (розширений)

2.16.67 MT_MICROSOFT_BASE_URL

Домен базової адреси регіону, як його визначено у розділі «Base URLs».

Типовим значенням є `api.cognitive.microsofttranslator.com` для Azure Global.

Для Azure China, будь ласка, скористайтеся `api.translator.azure.cn`.

2.16.68 MT_MICROSOFT_COGNITIVE_KEY

Ключ клієнта до програмного інтерфейсу когнітивних служб перекладу Microsoft.

Дивись також:

Когнітивні служби перекладу Microsoft, Машинний переклад, Автоматичні пропозиції, Когнітивні служби — програмний інтерфейс для перекладу текстів, Портал Microsoft Azure

2.16.69 MT_MICROSOFT_REGION

Префікс регіону, як його визначено у «Authenticating with a Multi-service resource».

2.16.70 MT_MICROSOFT_ENDPOINT_URL

Адреса домену кінцевої точки регіону для ключа доступу, як її визначено у розділі «[Authenticating with an access token](#)».

Типовим значенням є `api.cognitive.microsoft.com` для Azure Global.

Для Azure China, будь ласка, скористайтеся вашою кінцевою точкою з порталу Azure.

2.16.71 MT_MODERNMT_KEY

Ключ до програмного інтерфейсу для рушія комп'ютерного перекладу ModernMT.

Дивись також:

ModernMT `MT_MODERNMT_URL`

2.16.72 MT_MODERNMT_URL

Адреса ModernMT. Типовою для «хмарної» служби є адреса `https://api.modernmt.com/`.

Дивись також:

ModernMT `MT_MODERNMT_KEY`

2.16.73 MT_MYMEMORY_EMAIL

Ідентифікаційна адреса електронної пошти MyMemory. Приймає 1000 запитів на день.

Дивись також:

MyMemory, *Машинний переклад*, *Автоматичні пропозиції*, *MyMemory*: технічна специфікація програмного інтерфейсу

2.16.74 MT_MYMEMORY_KEY

Ключ доступу MyMemory для приватної пам'яті перекладів. Слід використовувати разом із `MT_MYMEMORY_USER`.

Дивись також:

MyMemory, *Машинний переклад*, *Автоматичні пропозиції*, *MyMemory*: генератор ключів до програмного інтерфейсу

2.16.75 MT_MYMEMORY_USER

Ідентифікатор користувача MyMemory для приватної пам'яті перекладів. Слід використовувати разом із `MT_MYMEMORY_KEY`.

Дивись також:

MyMemory, *Машинний переклад*, *Автоматичні пропозиції*, *MyMemory*: генератор ключів до програмного інтерфейсу

2.16.76 MT_NETEASE_KEY

Програмний ключ до програмного інтерфейсу NetEase Sight. Зареєструватися можна на <https://sight.youdao.com/>

Дивись також:

Програмний інтерфейс комп'ютерного перекладу NetEase Sight, Машинний переклад, Автоматичні пропозиції

2.16.77 MT_NETEASE_SECRET

Програмний пароль до програмного інтерфейсу NetEase Sight. Зареєструватися можна на <https://sight.youdao.com/>

Дивись також:

Програмний інтерфейс комп'ютерного перекладу NetEase Sight, Машинний переклад, Автоматичні пропозиції

2.16.78 MT_TMSERVER

Адреса, за якою працює tmserver.

Дивись також:

tmserver, Машинний переклад, Автоматичні пропозиції, tmserver

2.16.79 MT_YANDEX_KEY

Ключ до програмного інтерфейсу Yandex Translate. Зареєструватися можна на <https://yandex.com/dev/translate/>

Дивись також:

Переклад Yandex, Машинний переклад, Автоматичні пропозиції

2.16.80 MT_YOUDAO_ID

Ідентифікатор клієнта для програмного інтерфейсу Youdao Zhiyun. Зареєструватися можна на <https://ai.youdao.com/product-fanyi-text.s>.

Дивись також:

Програмний інтерфейс комп'ютерного перекладу Youdao Zhiyun, Машинний переклад, Автоматичні пропозиції

2.16.81 MT_YOUDAO_SECRET

Пароль клієнта для програмного інтерфейсу Youdao Zhiyun. Зареєструватися можна на <https://ai.youdao.com/product-fanyi-text.s>.

Дивись також:

Програмний інтерфейс комп'ютерного перекладу Youdao Zhiyun, Машинний переклад, Автоматичні пропозиції

2.16.82 MT_SAP_BASE_URL

Адреса програмного інтерфейсу до служби вузла перекладу SAP.

Дивись також:

Концентратор перекладів SAP, Машинний переклад, Автоматичні пропозиції

2.16.83 MT_SAP_SANDBOX_APIKEY

Ключ до програмного інтерфейсу для використання програмного інтерфейсу пісочниці

Дивись також:

Концентратор перекладів SAP, Машинний переклад, Автоматичні пропозиції

2.16.84 MT_SAP_USERNAME

Ваше ім'я користувача SAP

Дивись також:

Концентратор перекладів SAP, Машинний переклад, Автоматичні пропозиції

2.16.85 MT_SAP_PASSWORD

Ваш пароль SAP

Дивись також:

Концентратор перекладів SAP, Машинний переклад, Автоматичні пропозиції

2.16.86 MT_SAP_USE_MT

Визначає, чи слід також використовувати служби комп'ютерного перекладу на додачу до бази даних термінів.
Можливі значення: True / False

Дивись також:

Концентратор перекладів SAP, Машинний переклад, Автоматичні пропозиції

2.16.87 NEARBY_MESSAGES

Кількість показаних рядків, які розташовано поруч із поточним рядком перекладу. Це лише типові значення, користувачі можуть змінювати його за допомогою *Профіль користувача*.

2.16.88 PAGURE_CREDENTIALS

Нове в версії 4.3.2.

Список реєстраційних даних для серверів Pagure.

Підказка: Скористайтесь цією змінною, якщо Weblate має взаємодіяти із декількома серверами. Для оди-
нарної кінцевої точки Pagure скористайтесь `PAGURE_USERNAME` і `PAGURE_TOKEN`.

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.89 PAGURE_USERNAME

Нове в версії 4.3.2.

Ім'я користувача Pagure, яке буде використано для надсилання запитів щодо об'єднання для оновлень перекладів.

Дивись також:

PAGURE_CREDENTIALS, Pagure

2.16.90 PAGURE_TOKEN

Нове в версії 4.3.2.

Особистий жетон доступу Pagure, який використовується для створення викликів програмного інтерфейсу для оновлень перекладу.

Дивись також:

PAGURE_CREDENTIALS, Pagure, програмний інтерфейс Pagure

2.16.91 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

2.16.92 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

2.16.93 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

2.16.94 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

2.16.95 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.

2.16.96 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$"
```

2.16.97 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*

2.16.98 REPOSITORY_ALERT_THRESHOLD

Нове в версії 4.0.2.

Порогове значення для вмикання попередження щодо застарілих сховищ або сховищ, у яких міститься надто багато змін. Типове значення — 25.

Дивись також:

alerts

2.16.99 REQUIRE_LOGIN

Нове в версії 4.1.

Це увімкне `LOGIN_REQUIRED_URLS` та налаштовує набір бібліотек REST вимагати входу для всіх кінцевих точок програмного інтерфейсу.

Примітка: Це реалізовано у *Зразок налаштувань*. Для Docker скористайтесь `WEBLATE_REQUIRE_LOGIN`.

2.16.100 SENTRY_DSN

Нове в версії 3.9.

Sentry DSN, яким слід скористатися для *Збирання звітів щодо помилок*.

Дивись також:

Інтеграція Django із Sentry

2.16.101 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`

2.16.102 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.

2.16.103 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) скористайтесь `ENABLE_HTTPS`, а для зміни адреси скористайтесь `URL_PREFIX`.

Підказка: У контейнері Docker домен сайта можна налаштувати за допомогою `WEBLATE_ALLOWED_HOSTS`.

Дивись також:

`faq-site`, *Налаштовування дозволених вузлів*, *Належне налаштування HTTPS* `ENABLE_HTTPS`, `changesite`, The “sites” framework

2.16.104 SITE_TITLE

Назва сайта, яку буде використано для сайта і надсилання повідомлень електронною поштою.

2.16.105 SPECIAL_CHARS

Додаткові символи, які слід включити на візуальну клавіатуру, *Візуальна клавіатура*.

Типове значення:

```
SPECIAL_CHARS = ("\t", "\n", "...")
```

2.16.106 SINGLE_PROJECT

Нове в версії 3.8.

Переспрямовує користувачів безпосередньо до проєкту або складника замість показу панелі приладів. Ви можете встановити значення `True` — у цьому випадку це працюватиме, лише якщо на Weblate є лише один проєкт. У протилежному випадку встановіть ідентифікатор проєкту, і він безумовно переспрямовуватиме до цього проєкту.

Змінено в версії 3.11: У поточній версії цього параметра також приймається ідентифікатор проєкту для примусового показу лише цього проєкту.

Приклад:

```
SINGLE_PROJECT = "test"
```

2.16.107 STATUS_URL

Адреса, на яку екземпляр Weblate повідомляє про свій стан.

2.16.108 SUGGESTION_CLEANUP_DAYS

Нове в версії 3.2.1.

Автоматично вилучає пропозицій після вказаної кількості днів. Типовим є значення `None`, тобто пропозиції не вилучаються.

2.16.109 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*.

Дивись також:

Вбудовані визначення мов

2.16.110 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.

2.16.111 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

2.16.112 VCS_CLONE_DEPTH

Нове в версії 3.10.2.

Визначає, наскільки глибоким має бути клонування сховищ у Weblate.

Примітка: У поточній версії підтримку передбачено лише у *Git*. Типово, Weblate створює поверхневі клони сховищ, щоб пришвидшити клонування і заощадити місце на диску. Залежно від ваших підходів до користування (наприклад, використання нетипових *Доповнення*), 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
```

2.16.113 WEBLATE_ADDONS

Список додатків, якими можна скористатися. Щоб скористатися додатками, їх слід увімкнути для певного складника перекладу. Типово, до списку буде включено усі вбудовані додатки. При розширенні списку вам, ймовірно, слід зберегти наявні додатки увімкненими. Приклад:

```
WEBLATE_ADDONS = (  
    # Built-in 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.consistency.LanguaugeConsistencyAddon",  
    "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.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",  
    # Addon 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:

Доповнення, DEFAULT_ADDONS

2.16.114 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

2.16.115 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

2.16.116 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 сховищі 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

2.17 Зразок налаштувань

Наведений нижче приклад постачається як `weblate/settings_example.py` разом із Weblate:

```
#
# Copyright © 2012 - 2020 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#

import os
import platform
from logging.handlers import SysLogHandler

#
# Django settings for Weblate project.
#
```

(continues on next page)

(continued from previous page)

```

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",
            # 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,
        },
    },
}

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Data directory
DATA_DIR = os.path.join(BASE_DIR, "data")

# 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"),

```

(continues on next page)

(continued from previous page)

```

("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"),
("pt-br", "Português brasileiro"),
("ru", "Русский"),
("sk", "Slovenčina"),
("sl", "Slovenščina"),
("sq", "Shqip"),
("sr", "Српски"),
("sv", "Svenska"),
("tr", "Türkçe"),
("uk", "Українська"),
("zh-hans", "简体中文"),
("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

# 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.

```

(continues on next page)

(continued from previous page)

```

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(DATA_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",
    "django.contrib.staticfiles.finders.AppDirectoriesFinder",
    "compressor.finders.CompressorFinder",
)

# Make this unique, and don't share it with anybody.
# You can generate it using weblate/examples/generate-secret-key
SECRET_KEY = ""

_TEMPLATE_LOADERS = [
    "django.template.loaders.filesystem.Loader",
    "django.template.loaders.app_directories.Loader",
]
if not DEBUG:
    _TEMPLATE_LOADERS = [("django.template.loaders.cached.Loader", _TEMPLATE_
↵LOADERS)]
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",
            ],
            "loaders": _TEMPLATE_LOADERS,
        },
    },
]

# GitHub username for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None

# GitLab username for sending merge requests.

```

(continues on next page)

(continued from previous page)

```

# Please see the documentation for more details.
GITLAB_USERNAME = None
GITLAB_TOKEN = None

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth",
    # "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_BITBUCKET_KEY = ""
SOCIAL_AUTH_BITBUCKET_SECRET = ""
SOCIAL_AUTH_BITBUCKET_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",

```

(continues on next page)

(continued from previous page)

```

        "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

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")
    #     }
    # },
]

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [

```

(continues on next page)

(continued from previous page)

```

"weblate.middleware.RedirectMiddleware",
"weblate.middleware.ProxyMiddleware",
"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",
"weblate.accounts.middleware.RequireLoginMiddleware",
"weblate.api.middleware.ThrottlingMiddleware",
"weblate.middleware.SecurityMiddleware",
]

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",
    # 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",
    "compressor",
    "rest_framework",
    "rest_framework.authtoken",
    "django_filters",
]

# Custom exception reporter to include some details
DEFAULT_EXCEPTION_REPORTER_FILTER = "weblate.trans.debug.
↳ WeblateExceptionReporterFilter"

```

(continues on next page)

(continued from previous page)

```

# Default logging of Weblate messages
# - to syslog in production (if available)
# - otherwise to console
# - 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

if DEBUG or not HAVE_SYSLOG:
    DEFAULT_LOG = "console"
else:
    DEFAULT_LOG = "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",
        },
    },
}

```

(continues on next page)

(continued from previous page)

```

    "syslog": {
        "level": "DEBUG",
        "class": "logging.handlers.SysLogHandler",
        "formatter": "syslog",
        "address": "/dev/log",
        "facility": SysLogHandler.LOG_LOCAL2,
    },
    # 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.microsoft.MicrosoftCognitiveTranslation",
    # "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    # "weblate.machinery.modernmt.ModernMTTranslation",
    # "weblate.machinery.mymemory.MyMemoryTranslation",

```

(continues on next page)

(continued from previous page)

```

# "weblate.machinery.netease.NeteaseSightTranslation",
# "weblate.machinery.tmserver.AmagamaTranslation",
# "weblate.machinery.tmserver.TMServerTranslation",
# "weblate.machinery.yandex.YandexTranslation",
# "weblate.machinery.saptranslationhub.SAPTranslationHub",
# "weblate.machinery.youdao.YoudaoTranslation",
"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

# 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

```

(continues on next page)

(continued from previous page)

```

MT_SAP_SANDBOX_APIKEY = None
MT_SAP_USERNAME = None
MT_SAP_PASSWORD = None
MT_SAP_USE_MT = True

# Title of site to use
SITE_TITLE = "Weblate"

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

# 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
# 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
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000

# 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

# 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

```

(continues on next page)

(continued from previous page)

```

LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
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_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.CSharpFormatCheck",
#     "weblate.checks.format.JavaFormatCheck",

```

(continues on next page)

(continued from previous page)

```

# "weblate.checks.format.JavaMessageFormatCheck",
# "weblate.checks.format.PercentPlaceholdersCheck",
# "weblate.checks.format.VueFormattingCheck",
# "weblate.checks.format.I18NextInterpolationCheck",
# "weblate.checks.format.ESTemplateLiteralsCheck",
# "weblate.checks.angularjs.AngularJSInterpolationCheck",
# "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",
# )

# 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.consistency.LanguaugeConsistencyAddon",
#     "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.properties.PropertiesSortAddon",

```

(continues on next page)

(continued from previous page)

```

# "weblate.addons.git.GitSquashAddon",
# "weblate.addons.removal.RemoveComments",
# "weblate.addons.removal.RemoveSuggestions",
# "weblate.addons.resx.ResxUpdateAddon",
# "weblate.addons.yaml.YAMLCustomizeAddon",
# "weblate.addons.cdn.CDNJSAddon",
# "weblate.addons.autotranslate.AutoTranslateAddon",
# )

# 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",
            "PASSWORD": None,
            "CONNECTION_POOL_KWARGS": {},
        },
        "KEY_PREFIX": "weblate",
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "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
        "rest_framework.permissions.IsAuthenticated"
        if REQUIRE_LOGIN
        else "rest_framework.permissions.IsAuthenticatedOrReadOnly"
    ],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework.authentication.TokenAuthentication",
        "weblate.api.authentication.BearerAuthentication",
    ),
}

```

(continues on next page)

(continued from previous page)

```

        "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": ("rest_framework.pagination.PageNumberPagination"),
    "PAGE_SIZE": 20,
    "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
# )

# 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
# 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

```

(continues on next page)

(continued from previous page)

```

CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_SCHEDULE_FILENAME = os.path.join(DATA_DIR, "celery", "beat-schedule")
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"},
}

# 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
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 сховища 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 `pip3` 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:

Встановлення за допомогою Docker, Встановлення на Debian і Ubuntu, Встановлення у SUSE та openSUSE, Встановлення у RedHat, Fedora і CentOS, Встановлення 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.

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.

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 сховища 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 сховища Git модуля обробки.

Ви можете або визначити проєкт або складник для оновлення (наприклад, `weblate/application`), або скористатися `--all` для оновлення усіх наявних складників.

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 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`.

2.18.10 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

2.18.11 dumpuserdata

weblate dumpuserdata <file.json>

Створити дамп даних користувачів у файлі для наступного використання у *importuserdata*

Підказка: Ця можливість стане у пригоді, якщо ви вирішите перенести або злити екземпляри Weblate.

2.18.12 import_demo

weblate import_demo

Нове в версії 4.1.

Створює демонстраційний проєкт зі складниками на основі <<https://github.com/WeblateOrg/demo>>.

Ця можливість може бути корисною для розробників Weblate.

2.18.13 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 вказане сховище системи керування версіями з цього складника для усіх 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",
    "repo": "weblate://test/test",
    "file_format": "aresource"
  }
]
```

Дивись також:

import_memory

2.18.14 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

2.18.15 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 проєкту на основі маски файлів.

`<project>` є назвою наявного проєкту, до якого буде імпортовано складники.

`<gitrepo>` визначає адресу сховища 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 сховища. 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 компонентів на основі файлів і додає компоненти, яких не існує. Він не вносить змін до наявних компонентів.

--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. *Налаштування складників*). Значенням параметра має бути коректний формальний вираз.

--main-component

Ви можете вказати, який складник буде вибрано як основний — той, який насправді містить сховище 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`.

2.18.16 importuserdata

weblate importuserdata <file.json>

Імпортує дані користувача з файла, який створено за допомогою `dumpuserdata`

2.18.17 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
```

2.18.18 install_addon

Нове в версії 3.2.

weblate install_addon --addon ADDON <project|project/component>

Встановлює 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 проект або складник для встановлення додатка (наприклад, `weblate/application`), або скористатися `--all` для включення усіх наявних складників.

Встановлення *Налаштувати виведення gettext* для усіх компонентів:


```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --  
↪update --all
```

Дивись також:

Доповнення

2.18.19 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.110n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF>.

2.18.20 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.

2.18.21 list_versions

weblate list_versions

Виводить список усіх залежностей Weblate та їхніх версій.

2.18.22 loadpo

weblate loadpo <project|project/component>

Перезавантажує переклади з диска (наприклад, у випадку, якщо ви виконували певні оновлення у сховищі 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 змінюєте підлегле сховище системи керування версіями Weblate, або в особливих випадках після оновлення Weblate.

2.18.23 lock_translation

weblate lock_translation <project|project/component>

Забороняє подальший переклад складника.

Підказка: Корисна, якщо ви хочете виконати певні дії із супроводу над підлеглим сховищем.

Ви можете або визначити проєкт або складник для оновлення (наприклад, `weblate/application`), або скористатися `--all` для оновлення усіх наявних складників.

Дивись також:

`unlock_translation`

2.18.24 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 конкурентність при одночасному внесенню оновлень до сховища сторонньою особою), і вилучити мови із суфіксом (*generated*).

2.18.25 pushgit

weblate pushgit <project|project/component>

Записує внесені зміни до основного сховища системи керування версіями.

--force-commit

Примусово вносить до сховища усі зміни з черги до надсилання змін до сховища.

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 сховища автоматично, якщо позначено пункт *Відправляти при поданні* у налаштуваннях *складника* (типові налаштування).

2.18.26 unlock_translation

weblate unlock_translation <project|project/component>

Розблоковує вказаний складник, роблячи його доступним для перекладу.

Підказка: Корисна, якщо ви хочете виконати певні дії із супроводу над підлеглим сховищем.

Ви можете або визначити проєкт або складник для оновлення (наприклад, `weblate/application`), або скористатися `--all` для оновлення усіх наявних складників.

Дивись також:

lock_translation

2.18.27 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:

Керування доступом

2.18.28 setuplang

weblate setuplang

Оновлює список визначених мов у Weblate.

--no-update

Вимикає автоматичні оновлення наявних мов (лише додає нові мови).

2.18.29 updatechecks

weblate updatechecks <project|project/component>

Оновлює усі перевірки для усіх 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.

2.18.30 updategit

weblate updategit <project|project/component>

Отримує дані з віддалених сховищ 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 сховищі, 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 інформацію вашим перекладачам за допомогою створення оголошень — на рівні усього сайта, 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 сайта до визначення цілей для перекладу.

Оголошення можна розміщувати на будь-якому рівні меню *Управління* за допомогою *Повідомлення оголошення*:

The screenshot shows the 'Post announcement' interface in the Weblate web application. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. A blue notification banner states: 'Translations will be used only if they reach 60%'. Below the navigation bar, there's a secondary bar with 'Components', 'Languages', 'Info', 'Search', 'Glossaries', 'Insights', 'Files', 'Tools', 'Manage', 'Share', and 'Not watching'. The main form is titled 'Post announcement' and contains the following fields:

- Message:** A large text input area with a placeholder. Below it, a note says: 'You can use Markdown and mention users by @username.'
- Category:** A dropdown menu currently set to 'Info (light blue)'. A note below says: 'Category defines color used for the message.'
- Expiry date:** A date input field with the format 'mm/dd/yyyy' and a calendar icon. A note below says: 'The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.'
- Notify users:** A checked checkbox. A note below says: 'The message is shown for all translations within the project, until its given expiry, or permanently until it is deleted.'

At the bottom left of the form is a dark 'Add' button. The footer of the page shows 'Powered by Weblate 4.4' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and '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.

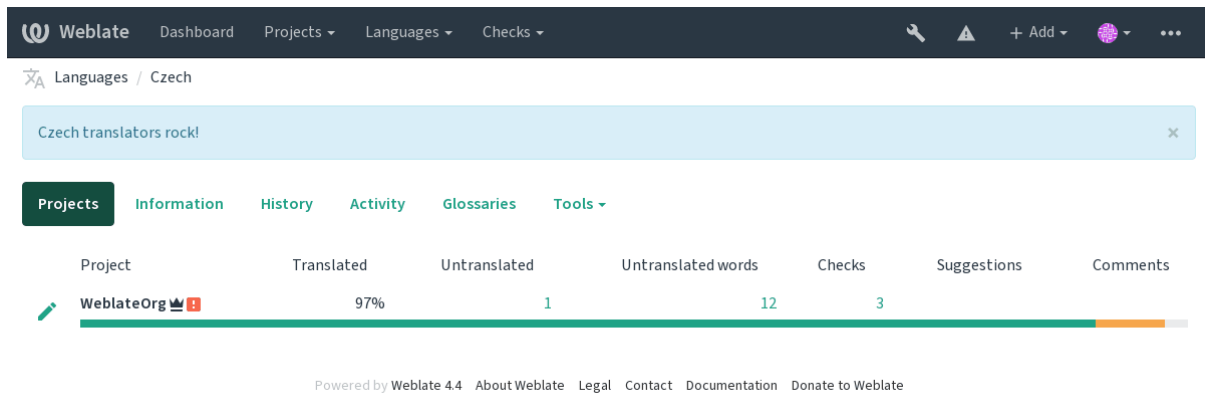
Вказано складник

Показано для вказаного складника і усіх його перекладів.

Вказано мову

Показано для огляду мови і 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 ідентифікатор створенням правил *Автоматичне призначення списку складників*.

- Корисно для супроводу списків складників для великих встановлень, або якщо вам потрібен один список складників із 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

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="✕"/>

[+ Add another Automatic component list assignment](#)

2.21 Додаткові модулі Weblate

Для вашої конфігурації доступними є декілька додаткових модулів.

2.21.1 Засіб експортування Git

Нове в версії 2.10.

Надає вам доступ лише до читання підлеглого сховища Git за допомогою HTTP(S).

Встановлення

1. Додайте `weblate.gitexport` до запису встановлених програм у `settings.py`:

```
INSTALLED_APPS += ("weblate.gitexport",)
```

2. Експортування наявних сховищ перенесенням вашої бази даних після встановлення:

```
weblate migrate
```

Користування

The module automatically hooks into Weblate and sets the exported repository URL in the *Налаштовування складників*. The repositories are accessible under the `/git/` part of the Weblate URL, for example `https://example.org/git/weblate/master/`.

Repositories for publicly available projects can be cloned without authentication:

```
git clone 'https://example.org/git/weblate/master/'
```

Access to the repositories with restricted access (using *Керування доступом на рівні проекту* or when `REQUIRE_LOGIN` is enabled) requires a API token which can be obtained in your *Профіль користувача*:

```
git clone 'https://user:KEY@example.org/git/weblate/master/'
```

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.

Встановлення

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 суперкористувача (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 проєкту користувач може вибрати, до якого рахунка слід включати проєкт, якщо користувач має доступ до декількох з них.

2.21.3 Юридичні 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

Примітка: Legal documents for the Hosted Weblate service are available in this Git repository <<https://github.com/WeblateOrg/wllegal/tree/master/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.

Встановлення

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
```

3. 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

Дивись також:

Кешування аватару, AVATAR_URL_PREFIX, ENABLE_AVATARS

2.21.5 Захист від спаму

Ви можете захистити систему від спаму пропозиціями від користувачів, які не зареєструвалися у системі, за допомогою служби akismet.com.

1. Встановіть модуль Python *akismet*
2. Налаштуйте ключ до програмного інтерфейсу Akismet.

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

2.21.6 Підписування внесків Git за допомогою GnuPG


Нове в версії 3.1.

Усі внески можна підписувати за допомогою ключа GnuPG екземпляра Weblate.

1. 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»:


[Dashboard](#)
[Projects](#)
[Languages](#)
[Checks](#)
[Register](#)
[Sign in](#)

[About Weblate](#) / [Weblate keys](#)

[About Weblate](#)
[Statistics](#)
[Keys](#)

SSH key

SSH key not available.

Commit signing

All commits made with Weblate are signed with the GPG key C0F6E5B4379EF46B1DF13D27CBFA89B076096E48, for which the corresponding public key is found below.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQGNBF/Yd8sBDAC6NWpo5LpLY5rB2VNIIdgGoM2Ohv9F4xC+IDRI4u6i7uuhAhvI0
d+6RiCaErdJMIFb2Z+4VxDxcX2AJyGC7n1asDdgPKlu7LQ3D4lxaRw/L9iRkDn5
BsZ4gm3j1g7q4b9+FUiejfAFrFKDAxjnQ77NiT93nze7FIAaL5fmUPtjCiePIOG
oYn8XZAeOJpBwAfF1uj3dBjYuuCpnB1oN1GOf4pNylmiciKpusqAquZi7t7krmYE
p6oR07uAz3t/3egbyJaRzr8ECBaSfGRqx6OT1XxvjZEXhb6EdP7NGJzaWWJ8TIQq
BnGHBRSV/Hrn7bsqGhsP0/jlOBUD8E7xiuyfSWw5mE4DYU5OqCKoTg48t/uaOMG0
7gAe+mBnGlo+zhLFWVhYNEA0dlkRkNBvbKkU2koOYKjD91ePiPjHf3sO7PaRkWi
uAWJ0FFSeaeYn6ASP9AjiMcnJRG6QMtoEol4JcwmMQY3MrHWDfJVnenwRcbk44v
hwK7PViZH2prB8sAEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J
Ac4EEwEKADgWlQTA9uW0N570ax3xPSfL+omwdgluSAUCX9h3ywlbaAwULCQgHAgYV
CgkICwIEFgIDAQIeAQIXgAAKCRDL+omwdgluSBBDDAC16BEvUHHpBLS7GTQkbeD
lmzz9hXbW3VZUWf+NgQv7TtWh+/iECb2Pt4vH9V2s7K3NvfNn3lsyFx4jnsI0CYA
GETRRQygUt0Zxv3l0aTQbLpfEdOUiere2vmTMfgDcBV/+uhRFRsxL5rAtBv85bT
G8FJwGUMTvOhVhC7inlkO5wS9cu3vnlVv9ofD+BB6inn9BfSw5QJYeN08C1eivm2+
```

Powered by Weblate 4.4
 [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.

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	5	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

Якщо користувач не зможе увійти до системи `AUTH_LOCK_ATTEMPTS` разів, на обліковому записі буде виключено розпізнавання за паролем, аж доки не буде виконано процедуру скидання пароля облікового запису.

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

2.22 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 нетипового коду Python (який називається модулем), потрібне сховище для зберігання його даних — або у загальносистемному каталозі (зазвичай, щось подібне до `/usr/lib/python3.7/site-packages/`), або у каталозі Weblate, який також додається до шляху пошуку інтерпретатора мови.

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 модуля 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:

Managing 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* і 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",)

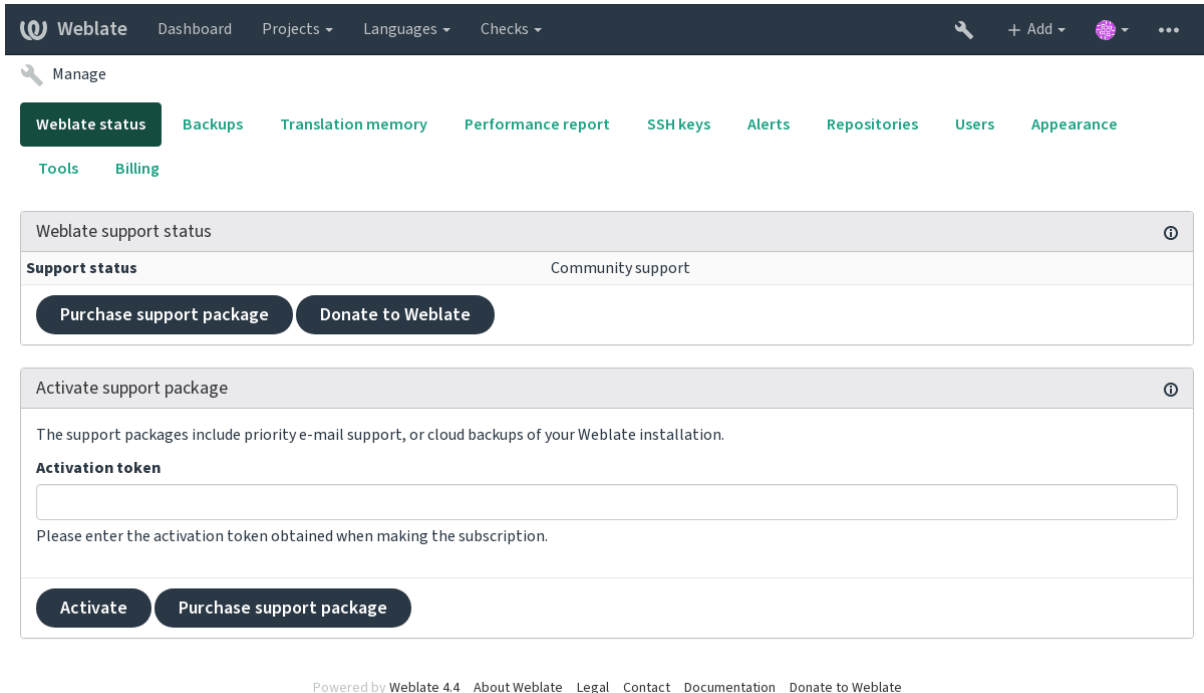
# Addons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

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 `/management/`. 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:



2.23.1 Адміністративний інтерфейс Django

Попередження: Буде вилучено згодом, оскільки його використання призводить до проблем — більша частина можливостей доступна до безпосереднього керування у Weblate.

Тут ви можете керувати об'єктами, які зберігаються у базі даних, зокрема записами користувачів, перекладів та іншими параметрами:

Weblate administration
WELCOME **WEBLATE TEST** / RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT

Site administration

REPORTS

Weblate support status
Status of repositories
SSH keys
Performance report
Translation memory

ACCOUNTS

Audit logs
Profiles
Verified emails

AUTH TOKEN

Tokens

AUTHENTICATION

Groups
Roles
Users

BILLING

Billings
Invoices
Plans

FONTS

Font groups
Fonts

GLOSSARIES

Glossaries

LEGAL

Agreements

PYTHON SOCIAL AUTH

Associations
Nonces
User social auths

SCREENSHOTS

Screenshots

TRANSLATION MEMORY

Memorys

WEBLATE CONFIGURATION

Settings

WEBLATE LANGUAGES

Languages

WEBLATE TRANSLATIONS

Announcements
Component lists
Components
Contributor agreements
Projects

Recent actions

My actions
None available

У розділі *Звіти* ви можете перевірити стан вашого сайта, скоригувати його для *Промислові налаштування* або керувати ключами SSH, які використовуються для доступу до *Доступ до сховищ*.

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 усіх 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:
Display name

URL slug:
Name used in URLs and filenames.

Project website:
Main website of translated project.

Mailing list:
Mailing list for translators.

Translation instructions:
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.

Controlo de acceso:
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:
Comma-separated list of language code mappings, for example: en_GB:en,en_US:en

Дивись також:

Налаштування проєкту

Двомовні складники

Після додавання проєкту до нього можна додати складники перекладу. (Див. [Налаштування складників](#), щоб дізнатися більше про окремі параметри):

Webate administration

HELLO! WEBLATE TESTRETURN TO WEBLATEDOCUMENTATIONCHANGE ALIASESLOG OUT

HomeWebate TranslationsComponentsAdd Component

Report error or improvement

Add Component

Required fields are marked in bold.

Component name:

Language names

Display name

URL slug:

language-names

Name used in URLs and for names

Project:

weblate.org

Name used in URLs and for names

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 (language) component to share it with other components

Repository push URL:

URL of a push repository, pushing is turned off if empty

Repository browser:

https://github.com/weblateorg/demo/blob/branch/\$(filename).it

Link to repository browser, use (branch) for branch, (filename) and (branch) as filename and file placeholders

Exported repository URL:

URL of repository where users can fetch changes from Weblate

Source string bug reporting address:

Email address for reports on errors in source strings. Leave empty for no emails

Repository branch:

Repository branch to translate

Push branch:

Branch for pushing changes, leave empty to use repository branch

Filename:

weblate/language/foruk/*LC_MESSAGES/it

Path of file to translate relative to repository root, use * instead of language code, for example: path/to/foruk/*LC_MESSAGES/it

Monolingual base language file:

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, it is most cases this is a translation file provided by developers and is used when creating actual source strings

Template for new translations:

weblate/language/foruk/gettext

Filename of file used for creating new translations, for gettext choose .pot file

File format:

gettext PO file

☐ Locked

Locked components will not get any translation updates

☒ Allow translation propagation

Whether translation updates in other components will create automatic translation in this one

☒ Turn on suggestions

Whether to allow translator suggestions at all

☐ Suggestion voting

Whether users can vote for suggestions

Autosuggest suggestions:

0

Automatically suggest suggestions with this number of votes, and it is turn it off

Translation flags:

Additional comma separated flags to influence quality checks. Possible values can be found in the documentation

Enforced checks:

List of checks which cannot be ignored

Translation license:

GNU General Public License v3.0 or later

Contributor agreement:

User agreement which needs to be approved before a user can translate this component

Adding new translation:

Create new language file

How to handle requests for creating new translations

Language code style:

Default based on the file format

Customize language code used to generate the filename for translations created by Weblate

Merge style:

Release

Define whether Weblate should merge the upstream repository or release changes into it

Current message when translating:

Translated using Weblate (% language, name %)
Currently translated at (% state translated, percent %%) (% state translated %) of (% state all %) strings
Translation (% project, name %) (% component, name %)
Translate URL: (% url %)

You can use template language for various info, please consult the documentation for more details

Current message when adding translation:

Added translation using Weblate (% language, name %)

You can use template language for various info, please consult the documentation for more details

Current message when removing translation:

Deleted translation using Weblate (% language, name %)

You can use template language for various info, please consult the documentation for more details

Current message when merging translation:

Merge branch (% component, name, branch %) into Weblate

You can use template language for various info, please consult the documentation for more details

Current message when adding a change:

Update translation files
Updated by (% editor, name %) Push in Weblate
Translation (% project, name %) (% component, name %)
Translate URL: (% url %)

You can use template language for various info, please consult the documentation for more details

Contributor name:

Weblate

Contributor e-mail:

noreply@weblate.org

☒ Push on commit

Whether the repository should be pushed upstream on every commit

Age of changes to commit:

24

Time in hours after which any pending changes will be committed to the VCS

☒ Lock on error

Whether the component should be locked on repository errors

Source language:

English

Language used for source strings in all components

Language filter:

tr@heh@it

Regular expression used to filter translation files when searching for filename

Variants regular expression:

Regular expressions used to determine variants of a string

Priority:

Medium

Components with higher priority are offered first to translators

☐ Restricted component

Restrict access to the component to only those explicitly given permission

Save and add anotherSave and edit existing stringsSave

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):

Webplate administration

HELLO! WEBLATE TESTRETURN TO WEBLATEDOCUMENTATIONCHANGE PASSWORDSIGN OUT

HomeWebplate translationsComponentsAdd Component

Add Component

Report error / documentation

Required fields are marked in bold.

Component name:

Android

Display name

URL slug:

android

Name used in URLs and filenames

Project:

weblate.org

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:

weblate:weblateorg/language-names

URL of a repository used by Weblate (language component to share it with other components)

Repository push URL:

URL of a push-repository pushing is turned off if empty

Repository browser:

Link to repository browser used (branch) for branch, (filename) and (diff) as filename and file placeholders

Exported repository URL:

URL of repository where users can fetch changes from Weblate

Source string bug reporting address:

Email address for reports on errors in source strings. Leave empty for no emails

Repository branch:

Repository branch to translate

Push branch:

Branch for pushing changes, leave empty to use repository branch

Filename:

app/src/main/res/values-*/strings.xml

Path of file to translate relative to repository root, use "*" instead of language code, for example: src/main/res/values-*/strings.xml

Monolingual base language file:

app/src/main/res/values/strings.xml

Filename of translation base file, containing all strings and their sources. 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. It is most cases this is a translation file provided by developers and is used when creating actual source strings

Template for new translations:

Filename of file used for creating new translations. For gettext choose .pot file

File format:

Android String Resource

☐ Locked

Locked components will not get any translation updates

☒ Allow translation propagation

Whether translation updates in other components will create automatic translation in this one

☒ Turn on suggestions

Whether to allow translator suggestions at all

☐ Suggestion voting

Whether users can vote for suggestions

Autosuggest suggestions:

0

Automatically suggest suggestions with this number of votes, use 0 to turn it off

Translation flags:

Additional comma separated flags to influence quality checks. Possible values can be found in the documentation

Enforced checks:

List of checks which cannot be ignored

Translation license:

MIT License

Contributor agreement:

User agreement which needs to be approved before a user can translate this component

Adding new translation:

Create new language file

How to handle requests for creating new translations

Language code style:

Default based on the file format

Customize language code used to generate the filename for translations created by Weblate

Merge style:

Release

Define whether Weblate should merge the upstream repository or release changes into it

Current message when translating:

Translated using Weblate (language, name)
Currently translated at (state translated, percent) % of (state all) strings
Translated (project, name) (component, name)
Translated URL (url)

You can use template language for various info, please consult the documentation for more details

Current message when adding translation:

Added translation using Weblate (language, name)

You can use template language for various info, please consult the documentation for more details

Current message when removing translation:

Deleted translation using Weblate (language, name)

You can use template language for various info, please consult the documentation for more details

Current message when merging translation:

Merge branch (component, remote, branch) into Weblate

You can use template language for various info, please consult the documentation for more details

Current message when adding a change:

Update translation files
Updated (language, name) push in Weblate
Translated (project, name) (component, name)
Translated URL (url)

You can use template language for various info, please consult the documentation for more details

Contributor name:

Weblate

Contributor e-mail:

noreply@weblate.org

☒ Push on commit

Whether the repository should be pushed upstream on every commit

Age of changes to commit:

24

Time in hours after which any pending changes will be committed to the VCS

☒ Lock on error

Whether the component should be locked on repository errors

Source language:

English

Language used for source strings in all components

Language filter:

--*

Regular expression used to filter translation files when scoring for filename

Variables regular expression:

Regular expressions used to determine variants of a string

Priority:

Medium

Components with higher priority are offered first to translators

☐ Restricted component

Restrict access to the component to only those explicitly given permission

Save and add pointerSave and commit addressSave

Дивись також:

Налаштовування складників, Двомовні і одномовні формати

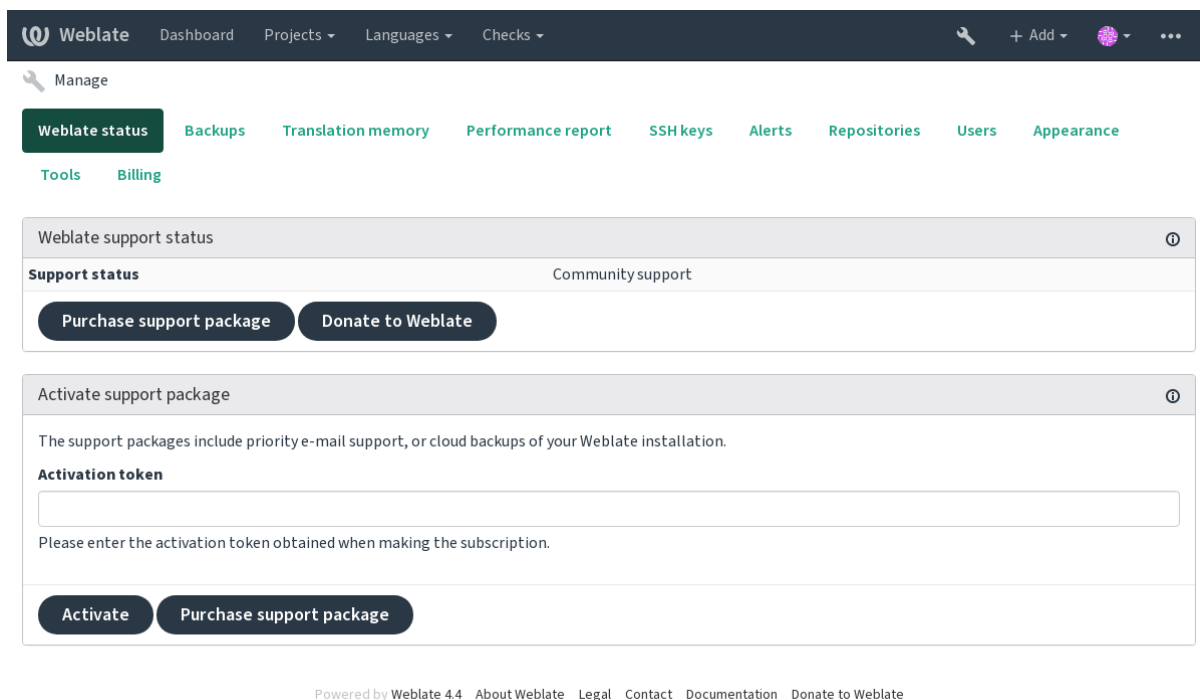
2.24 Отримання підтримки щодо Weblate

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.



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

Інші дані не подаються.

2.24.3 Служби інтеграції

- Переконайтеся, що ваш пакунок підтримки є чинним
- *Передбачене у Weblate сховище резервних копій*

Підказка: Придбані пакунки підтримки є активованими з моменту придбання — ними можна користуватися без інтеграції.

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 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.

2.25.2 Умови керування шифруванням США

У Weblate немає специфічного коду для шифрування, але Weblate може підлягати експортному контролю, оскільки використовує сторонні компоненти, 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 компоненти, які використовуються Weblate (у списку наведено лише компоненти, 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 компонентів, 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 розвитку Weblate. Будемо раді будь-якій допомозі — програмуванню, графічному дизайну, документації та безпосередній фінансовій підтримці:

- *Надсилання звітів щодо вад у Weblate*
- *Перші кроки у програмуванні коду Weblate*
- *Переклад Weblate*
- *Фінансування розробки Weblate*

3.1.1 Переклад Weblate

Weblate [перекладається](#) за допомогою самого Weblate. Не вагайтеся і долучайтесь до зусиль з забезпечення доступності Weblate якомога більшою кількістю мов.

3.1.2 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](#)

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 нашого 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](#).

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 джерельний код Weblate:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Створіть virtualenv:

```
virtualenv .venv
.venv/bin/activate
```

3. Встановіть 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
./manage.py test
```

Дивись також:

Встановлення 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:

```
./rundev.sh test --failfast weblate.trans
```

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
```

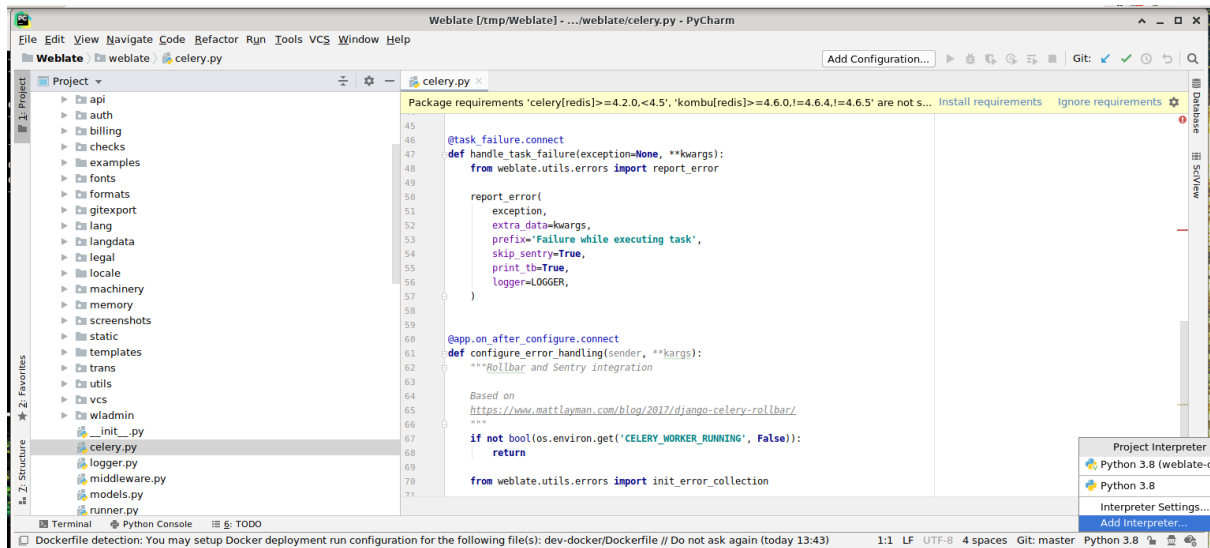
Запуск скрипту без аргументів призведе до повторного створення контейнера Docker і його перезапуску.

Примітка: Ця конфігурація є непридатною для промислових варіантів, до неї включено декілька модифікацій, які не є безпечними, але пришвидшують розробку.

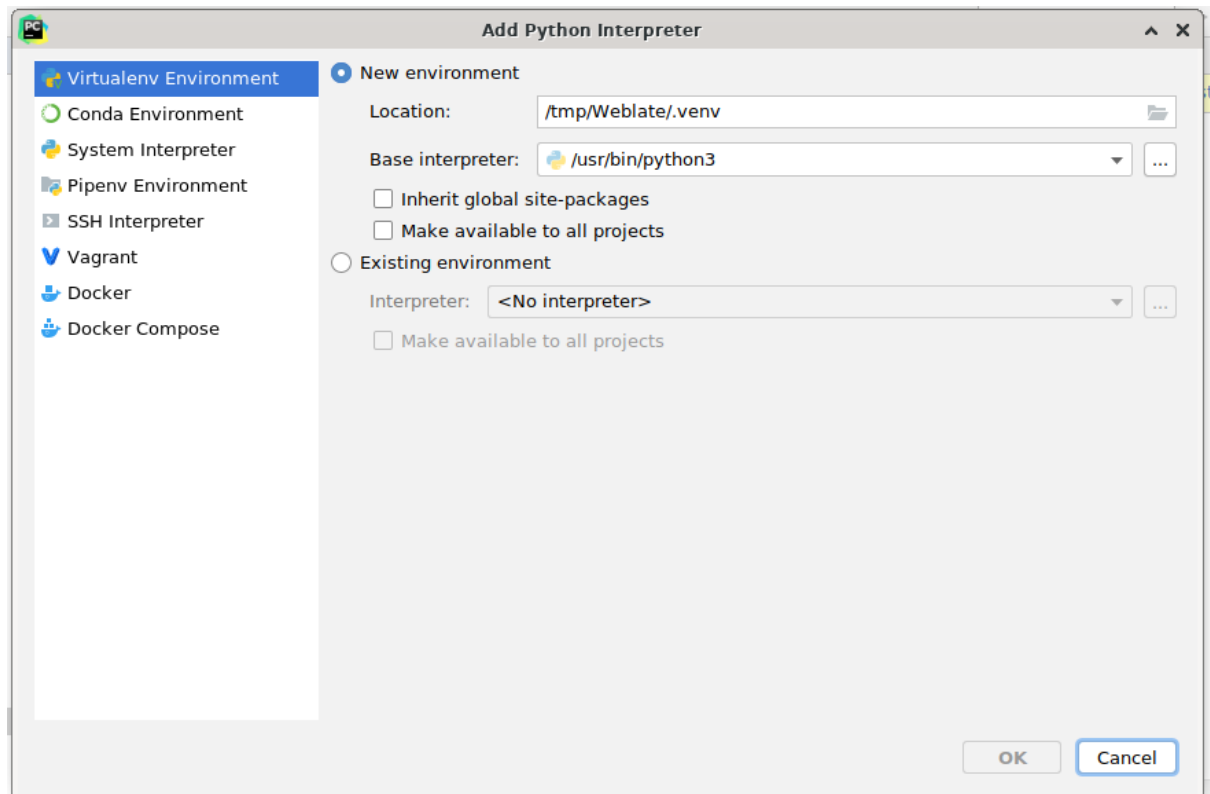
3.2.4 Програмування Weblate з використанням PyCharm

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 сховища Github і відкрили теку, до якої було клоновано код, у PyCharm. Після відкриття вікна середовища розробки першим кроком є визначення потрібного вам інтерпретатора:



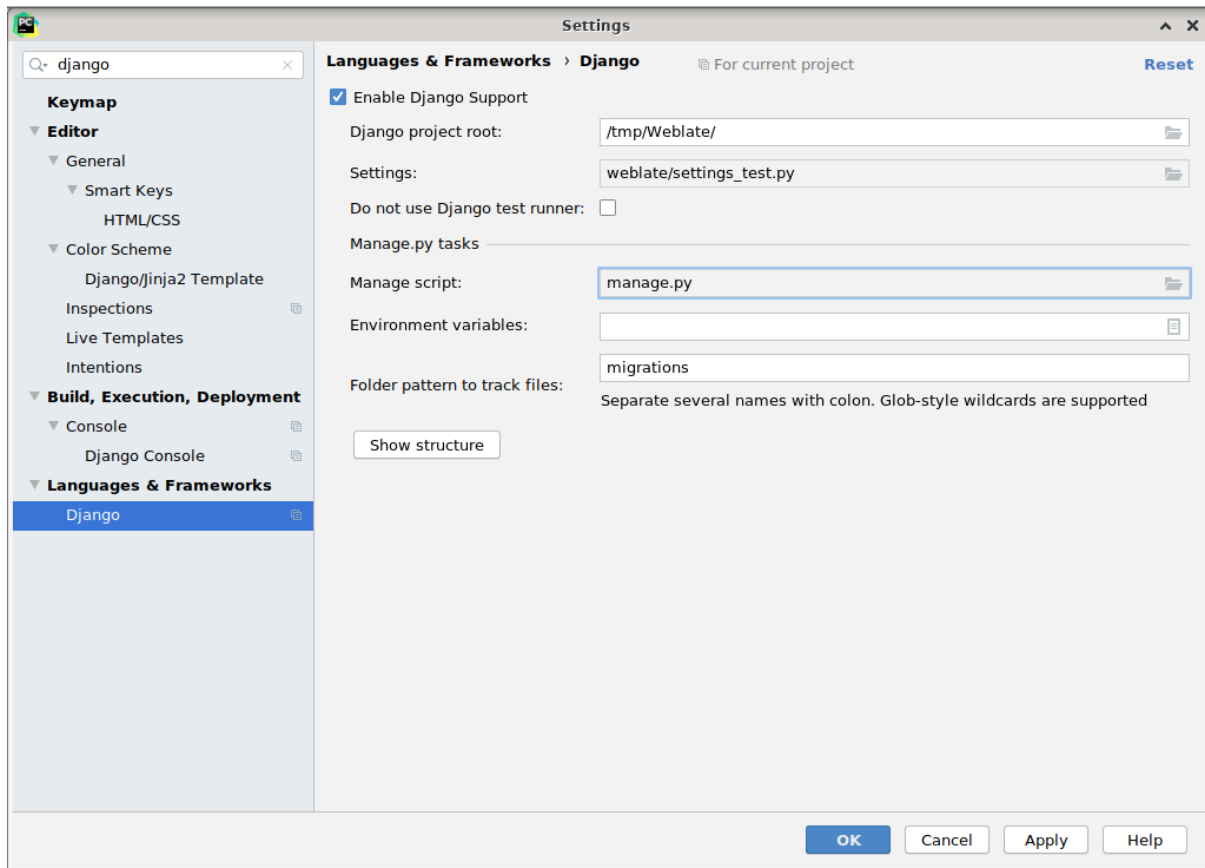
Ви можете або дозволити PyCharm створити для вас віртуальне середовище, або вибрати наявне середовище:



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.

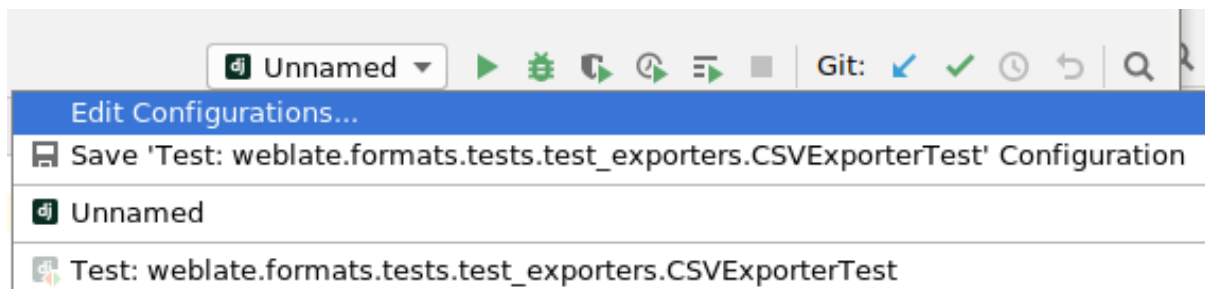
Другим кроком є належне налаштування для використання Django у PyCharm: основна проблема полягає в уможливленні негайного запуску тестів модулів у комплексному середовищі для розробки. Для цього вам

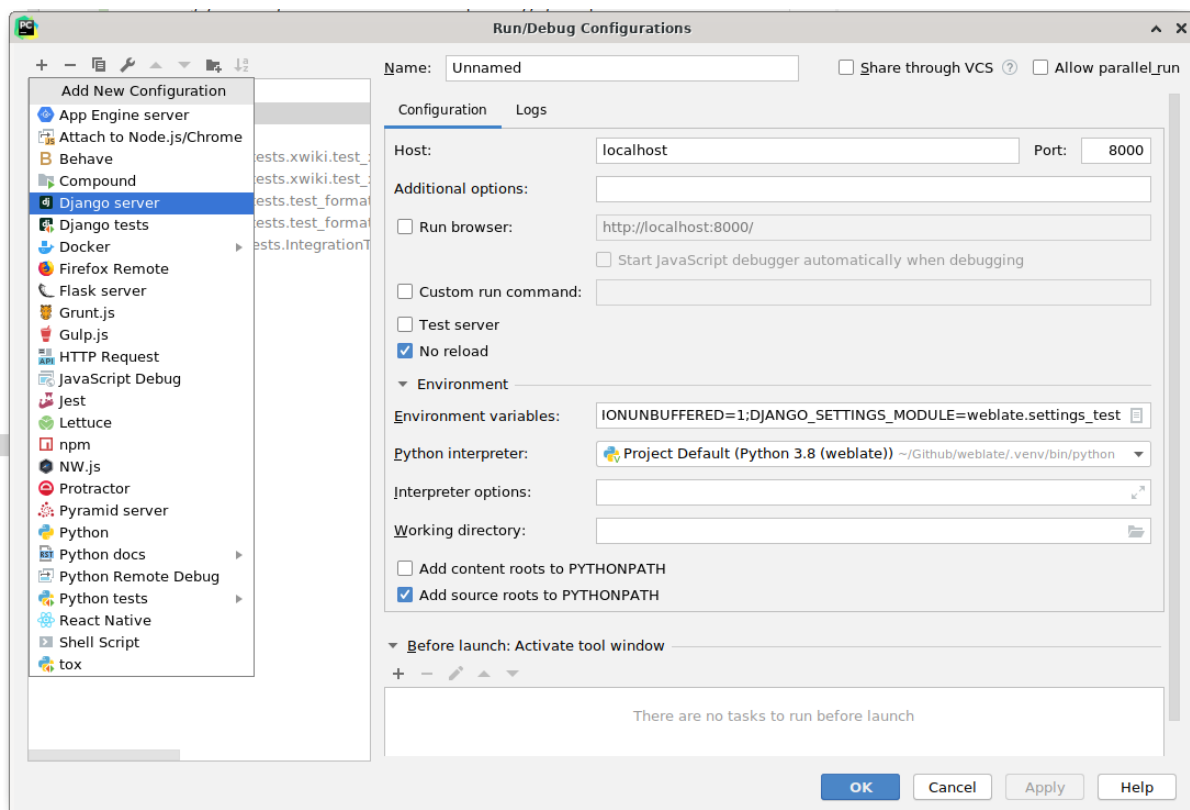
слід вказати кореневий шлях до Django та шлях до його параметрів:



Будьте обережні, *корінь проєкту Django* є кореневою текою сховища, а не підкаталогом weblate. Щодо параметрів, особисто автор використовує *settings_test* зі сховища коду, але ви можете створити власні параметри і вказати їх у відповідному місці.

Останнім кроком є уможливлення запуску сервера і розставляння точок зупинки у коді, щоб уможливити його діагностування. Це можна зробити створенням нових налаштувань *Django Server*:





Слід бути обережним із належним позначенням пункту «Не перезавантажувати»: якщо ви внесете зміни до якихось файлів, інтерактивне перезавантаження сервера не відбуватиметься, але засіб діагностики буде зупинено на встановленій вами точці зупинки.

3.2.5 Самопобудова вашого екземпляра для розробки

Ймовірно, вам варто скористатися `import_demo` для створення демонстраційних перекладів і `createadmin`, щоб створити запис користувача-адміністратора.

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 [Няпроці Weblate](#), щоб дізнатися про те, як Weblate виглядає з середини.

3.3.1 Принципи безпеки за компонуванням

Будь-який код для Weblate має бути написано із використанням [Security by Design Principles](#) (принципів безпеки за компонуванням).

3.3.2 Стандарт кодування

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 сховища `pre-commit`. Weblate, у якому містяться налаштування для перевірки внесених файлів на відповідність умовам. Після встановлення сховища (його вже включено у `requirements-lint.txt`) увімкніть його за допомогою команди `pre-commit install` у клоні сховища 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

Хиби можуть проявлятися як аварійні завершення роботи програми або помилкова поведінка програми. Ми будемо раді, якщо ви зберете будь-які дані щодо вади та надасте їх у [нашій системі стеження за вадами](#).

3.4.1 Режим зневадження

Вмикання діагностичного режиму призведе до показу виключень у браузері. Це корисно для діагностування проблем у вебінтерфейсі, але є неприйнятним для промислового середовища, оскільки призводить до погіршення швидкодії та може призвести до витоку конфіденційних даних.

Дивись також:

[Вимкнути режим діагностики](#)

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 є звичною практикою у царині Docker), отже ви можете ознайомитися із журналом за допомогою команди `docker-compose logs`.

Дивись також:

[Зразок налаштувань](#) містить налаштування `LOGGING`.

3.4.3 Аналіз аварійних завершень програми

Якщо програма завершує роботу в аварійному режимі, корисно зібрати якомога більше відомостей щодо аварії. Найпростішим способом досягти цього є використання сторонніх служб, які можуть зібрати потрібні дані в автоматичному режимі. Знайти дані щодо того, як налаштувати систему, можна у розділі *Збирання звітів щодо помилок*.

3.4.4 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.

3.4.5 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, що запобігти неналежній поведінці.

Якщо певні запити виконуються надто довго без додаткової індикації, вам варто встановити *dogslow* <<https://pypi.org/project/dogslow/>> і *Збирання звітів щодо помилок* і отримати точні та докладні трасування за допомогою засобу збирання даних помилок.

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 сховища Weblate:

docs Початковий код цієї документації побудовано з використанням [Sphinx](#).

dev-docker Код Docker для запуску сервера розробки, див. *Запуск Weblate локально у Docker*.

weblate Початковий код Weblate як програми [Django](#), 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. *Доповнення*.

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 *Юридичні відомості*.

machinery

Інтеграція служб комп'ютерного перекладу.

memory

Вбудована пам'ять перекладів, див. *Пам'ять перекладів*.

screenshots

Керування знімками вікон та модуль оптичного розпізнавання тексту.

trans

Головний модуль для обробки перекладів.

utils

Різноманітні допоміжні інструменти.

vcs

Абстракція системи керування версіями.

wladmin

Налаштовування адміністративного інтерфейсу Django.

3.6 Розробка додатків

За допомогою можливості *Доповнення* ви можете налаштувати процедуру локалізації у Weblate.

```
class weblate.addons.base.BaseAddon (storage=None)
```

```
    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 сховища.

post_push (*component*)

Скрипт, який вмикається після запису до сховища основної гілки розробки.

post_update (*component, previous_head: str, skip_push: bool*)

Скрипт обробки, який вмикається після оновлення сховища у основній гілці розробки.

Параметри

- **previous_head** (*str*) – HEAD сховища до оновлення, може бути порожнім для початкового клону
- **skip_push** (*bool*) – Визначає, чи слід пропускати при обробці додатком запис до основного сховища 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 сховища.

pre_push (*component*)

Скрипт, який вмикається перед записом із сховища до основної гілки розробки.

pre_update (*component*)

Скрипт, який вмикається перед оновленням сховища з основної гілки розробки.

save_state ()

Збереження відомостей щодо стану додатка.

stay_on_create = **False**

Базовий клас для додатків Weblate.

store_post_load (*translation, store*)

Скрипт обробки, який вмикається після обробки файлу і побудови класу сховища.

unit_pre_create (*unit*)

Скрипт обробки, який вмикається до створення нового модуля.

Ось приклад додатка:

```
#
# Copyright © 2012 - 2020 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
```

(continues on next page)

(continued from previous page)

```
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#

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 addon should receive
    events = (EVENT_PRE_COMMIT,)
    # Addon unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example addon")
    # Detailed addon description
    description = _("This addon 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 Керування залежностями

Для оновлення сторонніх бібліотек використовується програма для керування пакунками yarn. Налаштування цієї програми зберігаються `scripts/yarn`. Існує скрипт-обгортка `scripts/yarn-update` для оновлення бібліотек, збирання їх і копіювання до належних каталогів у `weblate/static/vendor`, де зберігається увесь код сторонніх бібліотек.

3.7.2 Стиль програмного коду

Weblate використовує для форматування коду у файлах JavaScript і CSS [Prettier](#).

Ми також використовуємо [ESLint](#) для перевірки коду JavaScript.

3.7.3 Локалізація

Якщо вам потрібен будь-який видимий користувачеві текст у коді бібліотеки, цей текст має бути локалізовано. У більшості випадків вам достатньо огорнути ваш текст у функцію `gettext`, але доступними є і 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

3.7.4 Піктограми

Поточна версія Weblate використовує піктограми material design. Якщо вам потрібна якась нова піктограма, скористайтеся <<https://materialdesignicons.com/>>.

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

Нашу систему стеження за вадами розташовано на GitHub:

Будемо раді звітам про будь-які вади або пропозиціям щодо поліпшень у Weblate. Якщо вами було виявлено проблему із захистом у Weblate, будь ласка, ознайомтеся із розділом «Негаразди з безпекою» нижче.

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](#).

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@cihar.com. 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](#), а дані щодо покриття розміщено на [Codecov](#).

Передбачено декілька завдань для перевірки різних аспектів:

- Перевірка модулів
- Збирання документації та зовнішні посилання
- Тестування перенесення для усіх 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`:

```
# Simple way to configure test database from environment

# 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
./ci/run-migrate
./ci/run-test
./ci/run-docs
./ci/run-setup
```

3.9.2 Локальне тестування

Щоб запустити комплекс тестів локально, скористайтесь 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:

```
# Simple way to configure test database from environment

# 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 Схема 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	
• 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
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	^[^]+\$
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	^[^]+\$
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 <i>admin</i>
		типове значення
	• full_name	<i>Повне ім'я</i>
		тип <i>рядок</i>
		приклади <i>Адміністратор Weblate</i>
		типове значення
	• email	<i>Електронна пошта</i>
		тип <i>рядок</i>
		приклади <i>noreply@example.com</i>
		типове значення
	• date_joined	<i>Дата приєднання</i>
		тип <i>рядок</i>
		приклади <i>2019-11-18T18:53:54.862Z</i>
		типове значення
• profile	<i>Профіль</i>	
	тип	<i>object</i>
	властивості	
	• language	<i>Мова</i>
		тип <i>рядок</i>
		приклади <i>cs</i>
		взірець <i>^.*\$</i>
		типове значення
	• suggested	<i>Кількість рядків-пропозицій</i>
		тип <i>integer</i>
		приклади <i>1</i>
		типове значення <i>0</i>
	• translated	<i>Кількість перекладених рядків</i>
		тип <i>integer</i>
		приклади <i>24</i>
		типове значення <i>0</i>
	• uploaded	<i>Кількість вивантажених знімків вікон</i>
		тип <i>integer</i>
		приклади <i>1</i>
		типове значення <i>0</i>
	• hide_completed_translations	<i>Приховати завершені переклади у домівці</i>
		тип <i>boolean</i>
		приклади <i>False</i>
		типове значення <i>True</i>
	• show_secondary_translations	<i>Показувати переклади іншими мовами в режимі Дзен</i>
		тип <i>boolean</i>
		приклади <i>True</i>
		типове значення <i>True</i>
	• hide_source_secondary_translations	<i>Приховати джерельний текст, коли є переклад будь-якою іншою мовою</i>
		тип <i>boolean</i>
		приклади <i>False</i>

continues on next page

Table 2 – continued from previous page

	• editor_link	типове значення		True	
		Посилання на редактор			
		тип		рядок	
		приклади			
		взірець		^.*\$	
	• translate_mode	Режим редактора перекладу			
		тип		integer	
		приклади		0	
		типове значення		0	
	• zen_mode	Режим Дзен-редактора			
		тип		integer	
		приклади		0	
		типове значення		0	
	• special_chars	Спеціальні символи			
		тип		рядок	
		приклади			
		взірець		^.*\$	
	• dashboard_view	Типовий вигляд домівки			
		тип		integer	
		приклади		1	
		типове значення		0	
	• dashboard_component_list	Типовий список складників			
		component_list		Немає	
		anyOf		тип	null
				тип	integer
	• languages	Перекладено мовами			
		тип		array	
		типове значення			
		записи		Код мови	
				тип	рядок
				приклади	cs
				взірець	^.*\$
				типове значення	
	• secondary_languages	Інші мови			
		languages		array	
		типове значення			
		записи		Код мови	
				тип	рядок
				приклади	sk
				взірець	^.*\$
				типове значення	
	• watched	Відстежувані проекти			
		тип		array	
типове значення					
записи		Ідентифікатор проекту			
		тип	рядок		
		приклади	weblate		
		взірець	^.*\$		
		типове значення			
• auditlog	Часопис перевірок				
	тип		array		
	типове значення				
	записи		Елементи		
			тип	object	

continues on next page

Table 2 – continued from previous page

		влас­тис­тос­ті		
		• address	IP-адреса	
			тип	рядок
			приклади	127.0.0.1
			взірець	^.*\$
			типове значення	
		• user_agent	User agent	
			тип	рядок
			приклади	PC / Linux / Firefox 70.0
			взірець	^.*\$
			типове значення	
		• timestamp	Часова позначка	
			тип	рядок
			приклади	2019-11-18T18:58:30.845Z
			взірець	^.*\$
			типове значення	
		• activity	Діяльність	
			тип	рядок
			приклади	вхід
			взірець	^.*\$
			типове значення	
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 оновити програми 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 -C docs update-screenshots`.

Створити випуск:

4. Створити випуск за допомогою команди `./scripts/create-release --tag` (нижче наведено вимоги).

Після випуску виконати вручну:

5. Оновити образ Docker.
6. Закрити етап на GitHub.
7. Щойно образ Docker буде перевірено, додати мітку і записати його до сховища.
8. Оновити до нової версії діаграму Helm.
9. Включити нову версію до `.github/workflows/migrations.yml`, щоб її було оброблено засобами тестування перенесення.
10. Збільшити номер версії у сховищі за допомогою команди `./scripts/set-version`.

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 сховищ git Weblate (скрипт записує мітки)
- Налаштована програма **hub** для доступу і створення випусків у сховищі Weblate
- SSH-доступ до сервера отримання пакунків Weblate (туди буде скопійовано пакунки Website)

3.12 Про Weblate

3.12.1 Цілі проєкту

Заснований на інтернет-технологіях інструмент локалізації із тісною прив'язкою до *систем керування версіями*, підтримкою широкого діапазону *форматів*. Спрощує доступ до перекладів для перекладачів.

3.12.2 Назва проєкту

«Weblate» є словом-гібридом від слів «web» («мережа») та «translate» («перекладати»).

3.12.3 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/>>.

3.12.4 Логотипи проекту

Логотипи проекту та інші графічні роботи можна знайти у сховищі <<https://github.com/WeblateOrg/graphics/>>.

3.12.5 Лідерство

Супровід цього проекту виконується Міхалом Чихаржем (Michal Čihař) <michal@cihar.com>.

3.12.6 Автори

Розробку Weblate було розпочато Міхалом Чихаржем (Michal Čihař) <michal@cihar.com>. З часу появи програми у 2012 внесок до неї зробили тисячі людей.

3.13 Ліцензія

© Michal Čihař <michal@cihar.com>, 2012–2020

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 ваше 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. Детальніше 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/>>.

4.1 Weblate 4.4

Released on December 15th 2020.

- Вдосконалено 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 видалення 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 заміників 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 дієпису.
- Удосконалено сторінки подробиць щодо рахунків та процедуру розміщення вільних проєктів.
- API розширеної статистики перекладів.
- Поліпшено вкладку інших перекладів під час перекладу.
- Додано API завдань.
- Вдосконалено механізми завантаження файлів.
- Improved display of user defined special characters.
- Improved auto translate performance.
- Several minor user interface improvements.

- Improved naming of ZIP downloads.
- Added option for getting notifications on unwatched projects.

4.2 Weblate 4.3.2

Випущено 4 листопада 2020 року.

- Виправлено збої роботи під час використання певних масок файлів складників.
- Вдосконалено точність перевірки на послідовне дублювання слів.
- Додано підтримку запитів щодо об'єднання Pagure.
- Удосконалено повідомлення про помилку при помилках реєстрації.
- Скасовано показ коментарів розробника розміткою.
- Спрощено налаштування сховищ Git з типовою гілкою, іншою ніж master.
- Гілка main тепер є типовою для новостворених внутрішніх сховищ.
- 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 сховищ із довгими шляхами.

4.3 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.
- Вдосконалено сумісність сценаріїв спостереження із сервером Bitbucket.
- Вдосконалено швидкодію оновлення пам'яті перекладів.
- Зменшено споживання пам'яті.
- Поліпшено швидкодію матричних подань.
- Додано підтвердження перед вилученням користувача з проєкту.

4.4 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.
- Вдосконалено відстеження початкових рядків.
- Вдосконалено сумісність з форматами JSON, YAML і CSV.
- Додано підтримку вилучення рядків.
- Поліпшено швидкодію під час отримання файлів.
- Вдосконалено панель керування сховищами.
- Реалізовано автоматичне вмикання формату java для Android.
- Додано підтримку локалізованих знімків вікон.
- Додаано підтримку Python 3.9.
- Виправлено переклад файлів HTML за певних умов.

4.5 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.

4.6 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.

4.7 Weblate 4.2

Випущено 18 серпня 2020 року.

- Удосконалено сторінки користувачів і додано список користувачів.
- Відкинута 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 зі сховищем.
- 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.
- Відкинута 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.

4.8 Weblate 4.1.1

Випущено 19 червня 2020 року.

- Виправлено зміну налаштувань автовиправлення або додатків у Docker.
- Виправлено можливе аварійне завершення роботи на сторінці інформації щодо 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](#).

4.9 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 сховища.
- 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](#).
- Реалізовано безумовний показ перевірки на обробку тексту, якщо її увімкнено.
- У програмному інтерфейсі передбачено фільтрування змін.
- Додано підтримку спільного використання глосаріїв між проєктами.

4.10 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.

4.11 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 суперкористувачами.
- Виправлено зміну адреси сховища при створенні складника.
- Виправлено аварійне завершення роботи, якщо зникає основне сховище коду.

4.12 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 сховища.
- 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.

4.13 Weblate 4.0.1

Випущено 16 квітня 2020 року.

- Виправлено встановлення пакунка з PyPI.

4.14 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 сховищем.
- 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 `i18next`.
- 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.

4.15 Випуски Weblate 3.x

4.15.1 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.

4.15.2 Weblate 3.11.2

Випущено 22 лютого 2020 року.

- Виправлено показ пропозицій.
- Виправлено помилкове сповіщення щодо відсутності слів у деяких рядках.

4.15.3 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 сховища коду.
- Удосконалено встановлення відповідності записам глосарія.

4.15.4 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 переспрямування 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.

4.15.5 Weblate 3.10.3

Випущено 18 січня 2020 року.

- Реалізовано підтримку translate-toolkit 2.5.0.

4.15.6 Weblate 3.10.2

Випущено 18 січня 2020 року.

- Додано позначку блокування для проєктів.
- виправлено ваду CSS, яка спричиняла блимання у деяких браузерях.
- виправлено пошук у системах із локалями, відмінними від англійської.
- Удосконалено встановлення відповідності у сховищах 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.

4.15.7 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.
- Удосконалено швидкодію при очищенні 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.

4.15.8 Weblate 3.10

Випущено 20 грудня 2019 року.

- Удосконалено інтерфейс користувача програми.
- 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 сховищ Git.
- Додано підтримку пошуку за віком рядка.
- Реалізовано показ початкової мови для усіх перекладів.
- Реалізовано показ контексту для сусідніх рядків.
- Додано підтримку сповіщень щодо дій у сховищі.
- 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.

4.15.9 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 сховищ 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.

4.15.10 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 вебпрограм.
- Додано попередження щодо складників при непідтримуваних 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 налаштовувати складники.
- Додано підтримку запитів щодо об'єднання GitLab.
- Удосконалено показ стану сховища.
- Реалізовано виконання автоматичного перекладу у фоновому режимі.

4.15.11 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 сховища.
- 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 сховищ 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 форматованих 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.

4.15.12 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.

4.15.13 Weblate 3.7

Випущено 21 червня 2019 року.

- Додано окрему чергу Celery для сповіщень.
- Використано однорідний вигляд із програмою 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 сховищем 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.

4.15.14 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 сховища системи керування версіями.

4.15.15 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 видобування 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 налаштовувати сповіщення для адміністративних проєктів.
- Удосконалено обробку трілітерних кодів мов.

4.15.16 Weblate 3.5.1

Випущено 10 березня 2019 року.

- Виправлено приклад модуля Celery для systemd.
- Виправлено сповіщення від сховищ 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.

4.15.17 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 сховища коду.
- Виправлено можливу втрату метаданих при пересуванні складника між проєктами.
- 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 проєкти із несплачених рахунків після 45 днів.

4.15.18 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 сховища.
- 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.

4.15.19 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.

4.15.20 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.

4.15.21 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 показника.
- Удосконалено перевірки при створенні складників.
- Додано підтримку вилучення застарілих пропозицій.

4.15.22 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 із 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 сховищ.
- 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.

4.15.23 Weblate 3.1.1

Випущено 27 липня 2018 року.

- Виправлено помилки комплексу тестування у деяких конфігураціях.

4.15.24 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.

4.15.25 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.

4.15.26 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.

4.16 Випуски Weblate 2.x

4.16.1 Weblate 2.20

Випущено 4 квітня 2018 року.

- Підвищення швидкості клонування сховищ `subversion`.
- Змінено блокування сховища — 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.

4.16.2 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.

4.16.3 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 сховищ Subversion, у яких не використовується stdlayout.
- Додано додатки для налаштування процедур перекладу.

4.16.4 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.

4.16.5 Weblate 2.17.1

Випущено 13 жовтня 2017 року.

- Виправлено роботу комплексу тестування у певних ситуаціях.
- Оновлення локалей.

4.16.6 Weblate 2.17

Випущено 13 жовтня 2017 року.

- Реалізовано типове поверхнєве клонування Weblate сховищ 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.

4.16.7 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.

4.16.8 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.

4.16.9 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.

4.16.10 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.

4.16.11 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.

4.16.12 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.
- ALC на рівні проєкту у новій версії реалізовано за допомогою групових ACL.
- Додано засоби точнішого керування правами доступу.
- Різноманітні незначні удосконалення інтерфейсу користувача.

4.16.13 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.

4.16.14 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.

4.16.15 Weblate 2.10.1

Випущено 20 січня 2017 року.

- Усунено витік даних щодо існування облікового запису у формі скидання пароля (CVE-2017-5537).

4.16.16 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 сховищ 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 усіх перекладів
- Додано базовий програмний інтерфейс для змін і рядків.
- Додано підтримку сервера комп'ютерного перекладу Apertium APu.

4.16.17 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 проєкту можуть налаштовувати складник перекладу і подробиці щодо проєкту.
- Використано термін «Watched» замість «Subscribed» для проєктів.
- Реалізовано спостереження за проєктом безпосередньо зі сторінки проєкту.
- 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 сховищ Git.
- Додано можливість пошуку із заміною.
- Додано підтримку вивантаження візуального контексту (знімків вікон) для перекладів.

4.16.18 Weblate 2.8

Випущено 31 серпня 2016 року.

- Удосконалення у документації.
- Переклади.
- Оновлено паковані 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.

4.16.19 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 іврити.
- 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 сховища, статистика, ...).
- Для комплексу тестування більше не потрібен доступ до мережі для тестування сховищ.

4.16.20 Weblate 2.6

Випущено 28 квітня 2016 року.

- Виправлено перевірку чинності компонентів 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.

4.16.21 Weblate 2.5

Випущено 10 березня 2016 року.

- Виправлено автоматичний переклад для власників проєктів.
- Удосконалено швидкодію створення внесків і запису внесків до сховища.
- 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 RHP.
- 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 RHP
- Додано експортування XLIFF для словника.
- Додано експортування XLIFF і PO gettext для усіх перекладів.
- Удосконалення у документації.
- Додано підтримку для придатних до налаштування автоматичних призначень груп.
- Удосконалено додавання нових перекладів.

4.16.22 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 показника.
- 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.

4.16.23 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 сховища.
- 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.

4.16.24 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.

4.16.25 Weblate 2.1

Випущено 5 грудня 2014 року.

- Додано підтримку сховищ 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 сховищ.
- Реалізовано придатні до налаштування перевірки якості для окремих початкових рядків.
- Уможливлено приховування виконаних перекладів з панелі приладів.

4.16.26 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 сховищ 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.

4.17 Випуски Weblate 1.x

4.17.1 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.

4.17.2 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.

4.17.3 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.

4.17.4 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.

4.17.5 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.

4.17.6 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.
- Реалізовано базове покриття тестів рядків.

4.17.7 Weblate 1.3

Випущено 16 листопада 2012 року.

- Реалізовано сумісність із модулем бази даних PostgreSQL.
- Реалізовано вилучення мов, які вилучено у сховищі 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.

4.17.8 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 сховищами 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 сховища 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 порахованих рядків, які не пройшли перевірки.
- Реалізовано перевірку вимог під час налаштування.
- Удосконалення у документації.

4.17.9 Weblate 1.1

Випущено 4 липня 2012 року.

- Удосконалено декілька перекладів.
- Удосконалено перевірку при створенні складника.
- Додано підтримку спільних сховищ `git` для компонентів.
- Усунено обов'язкові внески при кожній спробі отримання даних з віддаленого сховища.
- Додано підтримку розвантажувального індексування.

4.17.10 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.

4.18 Випуски Weblate 0.x

4.18.1 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 ігнорування повідомлень, які складаються лише з рядків форматування.

4.18.2 Weblate 0.8

Випущено 3 квітня 2012 року.

- Власний повнотекстовий пошук замінено на Whoosh.
- Різноманітні виправлення і удосконалення у перевітках.
- Нова команда updatechecks.
- Багато оновлень перекладів.
- Додано словник для зберігання більшості часто використовуваних термінів.
- Додано /admin/report/ для огляду стану сховищ.
- 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 перевітками.
- 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 сховища за допомогою вебінтерфейсу.
- Додано огляд перекладів, які виконано іншими користувачами.

4.18.3 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.

4.18.4 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.

4.18.5 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 сховищі.
- 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).

4.18.6 Weblate 0.4

Випущено 8 лютого 2012 року.

- До документації додано підручник з користування.
- Виправлено скрипти стеження програмного інтерфейсу так, щоб використовувався захист CSRF.

4.18.7 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.

4.18.8 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.

4.18.9 Weblate 0.1

Випущено 6 лютого 2012 року.

- Початковий випуск.

W

`wlc`, [130](#)

`wlc.config`, [131](#)

`wlc.main`, [131](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component)/, 106
ANY /, 80	GET /api/components/(string:project)/(string:component)/, 104
/api	POST /api/components/(string:project)/(string:component)/, 102
GET /api/, 83	POST /api/components/(string:project)/(string:component)/, 103
/api/changes	POST /api/components/(string:project)/(string:component)/, 104
GET /api/changes/, 114	PUT /api/components/(string:project)/(string:component)/, 101
GET /api/changes/(int:id)/, 114	DELETE /api/components/(string:project)/(string:component)/, 101
/api/component-lists	PATCH /api/components/(string:project)/(string:component)/, 100
GET /api/component-lists/, 117	/api/glossary
GET /api/component-lists/(str:slug)/, 117	GET /api/glossary/, 118
POST /api/component-lists/(str:slug)/components/, 117	GET /api/glossary/(int:id)/, 118
PUT /api/component-lists/(str:slug)/, 117	GET /api/glossary/(int:id)/projects/, 120
DELETE /api/component-lists/(str:slug)/, 117	GET /api/glossary/(int:id)/terms/, 120
DELETE /api/component-lists/(str:slug)/components/(string:component_slug)/, 118	GET /api/glossary/(int:id)/terms/(int:term_id)/, 120
PATCH /api/component-lists/(str:slug)/, 117	POST /api/glossary/(int:id)/projects/, 120
/api/components	POST /api/glossary/(int:id)/terms/, 120
GET /api/components/, 97	PUT /api/glossary/(int:id)/, 119
GET /api/components/(string:project)/(string:component)/, 97	PUT /api/glossary/(int:id)/terms/(int:term_id)/, 120
GET /api/components/(string:project)/(string:component)/, 101	DELETE /api/glossary/(int:id)/, 120
GET /api/components/(string:project)/(string:component)/changes/, 101	DELETE /api/glossary/(int:id)/projects/, 120
GET /api/components/(string:project)/(string:component)/lock/, 102	DELETE /api/glossary/(int:id)/terms/(int:term_id)/, 121
GET /api/components/(string:project)/(string:component)/monolingual_base/, 104	PATCH /api/glossary/(int:id)/terms/(int:term_id)/, 121
GET /api/components/(string:project)/(string:component)/new_template/, 104	/api/groups
GET /api/components/(string:project)/(string:component)/repository/, 103	GET /api/groups/, 86
GET /api/components/(string:project)/(string:component)/screenshots/, 102	GET /api/groups/(int:id)/, 86
	POST /api/groups/, 86

POST /api/groups/(int:id)/componentlist	BATCH /api/projects/(string:project)/,
89	93
POST /api/groups/(int:id)/components/,	/api/roles
88	
POST /api/groups/(int:id)/languages/,	GET /api/roles/, 89
88	GET /api/roles/(int:id)/, 89
POST /api/groups/(int:id)/projects/,	POST /api/roles/, 89
88	PUT /api/roles/(int:id)/, 89
POST /api/groups/(int:id)/roles/, 88	DELETE /api/roles/(int:id)/, 90
PUT /api/groups/(int:id)/, 87	PATCH /api/roles/(int:id)/, 90
DELETE /api/groups/(int:id)/, 88	/api/screenshots
DELETE /api/groups/(int:id)/componentlist	POST /api/screenshots/(int:project_id),
89	GET /api/screenshots/, 114
DELETE /api/groups/(int:id)/components/(int:component_id),	GET /api/screenshots/(int:id)/, 114
88	GET /api/screenshots/(int:id)/file/,
DELETE /api/groups/(int:id)/languages/(string:language_code),	115
88	POST /api/screenshots/, 115
DELETE /api/groups/(int:id)/projects/(int:project_id),	POST /api/screenshots/(int:id)/file/,
88	115
PATCH /api/groups/(int:id)/, 87	POST /api/screenshots/(int:id)/units/,
	115
/api/languages	PUT /api/screenshots/(int:id)/, 116
GET /api/languages/, 90	DELETE /api/screenshots/(int:id)/, 116
GET /api/languages/(string:language)/,	DELETE /api/screenshots/(int:id)/units/(int:unit_
90	115
GET /api/languages/(string:language)/statistics/,	PATCH /api/screenshots/(int:id)/, 116
91	
POST /api/languages/, 90	/api/tasks
PUT /api/languages/(string:language)/,	GET /api/tasks/, 121
91	GET /api/tasks/(str:uuid)/, 121
DELETE /api/languages/(string:language)/,	/api/translations
91	
PATCH /api/languages/(string:language)/,	GET /api/translations/, 106
91	GET /api/translations/(string:project)/(string:com
	106
/api/projects	GET /api/translations/(string:project)/(string:com
GET /api/projects/, 92	108
GET /api/projects/(string:project)/,	GET /api/translations/(string:project)/(string:com
92	109
GET /api/projects/(string:project)/changes/,	GET /api/translations/(string:project)/(string:com
93	110
GET /api/projects/(string:project)/components/,	GET /api/translations/(string:project)/(string:com
94	111
GET /api/projects/(string:project)/languages/,	GET /api/translations/(string:project)/(string:com
97	109
GET /api/projects/(string:project)/repository/,	POST /api/translations/(string:project)/(string:co
93	109
GET /api/projects/(string:project)/statistics/,	POST /api/translations/(string:project)/(string:co
97	110
POST /api/projects/, 92	POST /api/translations/(string:project)/(string:co
POST /api/projects/(string:project)/components/,	111
95	POST /api/translations/(string:project)/(string:co
POST /api/projects/(string:project)/repository/,	109
94	DELETE /api/translations/(string:project)/(string
PUT /api/projects/(string:project)/,	108
93	
DELETE /api/projects/(string:project)/,	/api/units
93	
	GET /api/units/, 112

GET /api/units/(int:id)/, 112
 PUT /api/units/(int:id)/, 113
 DELETE /api/units/(int:id)/, 113
 PATCH /api/units/(int:id)/, 113

/api/users

GET /api/users/, 83
 GET /api/users/(str:username)/, 83
 GET /api/users/(str:username)/notifications/,
 85
 GET /api/users/(str:username)/notifications/(int:subscription_id)/,
 85
 GET /api/users/(str:username)/statistics/,
 85
 POST /api/users/, 83
 POST /api/users/(str:username)/groups/,
 85
 POST /api/users/(str:username)/notifications/,
 85
 PUT /api/users/(str:username)/, 84
 PUT /api/users/(str:username)/notifications/(int:subscription_id)/,
 85
 DELETE /api/users/(str:username)/, 85
 DELETE /api/users/(str:username)/notifications/(int:subscription_id)/,
 86
 PATCH /api/users/(str:username)/, 84
 PATCH /api/users/(str:username)/notifications/(int:subscription_id)/,
 86

/exports

GET /exports/rss/, 125
 GET /exports/rss/(string:project)/, 125
 GET /exports/rss/(string:project)/(string:component)/,
 125
 GET /exports/rss/(string:project)/(string:component)/(string:language)/,
 125
 GET /exports/rss/language/(string:language)/,
 125
 GET /exports/stats/(string:project)/(string:component)/,
 124

/hooks

GET /hooks/update/(string:project)/,
 122
 GET /hooks/update/(string:project)/(string:component)/,
 122
 POST /hooks/azure/, 123
 POST /hooks/bitbucket/, 122
 POST /hooks/gitea/, 123
 POST /hooks/gitee/, 123
 POST /hooks/github/, 122
 POST /hooks/gitlab/, 122
 POST /hooks/pagure/, 123

Symbols

- .XML resource file
 - file format, 67
- add
 - auto_translate command line option, 328
- addon ADDON
 - install_addon command line option, 333
- age HOURS
 - commit_pending command line option, 328
- author USER@EXAMPLE.COM
 - add_suggestions command line option, 327
- base-file-template TEMPLATE
 - import_project command line option, 331
- check
 - importusers command line option, 333
- config PATH
 - wlc command line option, 127
- config-section SECTION
 - wlc command line option, 127
- configuration CONFIG
 - install_addon command line option, 333
- convert
 - wlc command line option, 128
- email USER@EXAMPLE.COM
 - createadmin command line option, 329
- file-format FORMAT
 - import_project command line option, 331
- force
 - loadpo command line option, 334
- force-commit
 - pushgit command line option, 335
- format {csv,json,text,html}
 - wlc command line option, 127
- ignore
 - import_json command line option, 330
- inconsistent
 - auto_translate command line option, 328
- input
 - wlc command line option, 128
- key KEY
 - wlc command line option, 127
- lang LANGUAGE
 - loadpo command line option, 334
- language-code
 - list_translators command line option, 334
- language-map LANGMAP
 - import_memory command line option, 331
- language-regex REGEX
 - import_project command line option, 332
- license NAME
 - import_project command line option, 332
- license-url URL
 - import_project command line option, 332
- main-component
 - import_project command line option, 332
- main-component COMPONENT
 - import_json command line option, 330
- mt MT
 - auto_translate command line option, 328
- name
 - createadmin command line option, 329
- name-template TEMPLATE
 - import_project command line option, 331
- new-base-template TEMPLATE
 - import_project command line option, 331
- no-password
 - createadmin command line option, 329
- no-privs-update
 - setupgroups command line option, 336
- no-projects-update

setupgroups command line option, 336
 --no-update
 setuplang command line option, 336
 --output
 wlc command line option, 128
 --overwrite
 auto_translate command line option, 327
 wlc command line option, 128
 --password PASSWORD
 createadmin command line option, 329
 --project PROJECT
 import_json command line option, 330
 --source PROJECT/COMPONENT
 auto_translate command line option, 327
 --threshold THRESHOLD
 auto_translate command line option, 328
 --update
 createadmin command line option, 329
 import_json command line option, 330
 install_addon command line option, 333
 --url URL
 wlc command line option, 127
 --user USERNAME
 auto_translate command line option, 327
 --username USERNAME
 createadmin command line option, 329
 --vcs NAME
 import_project command line option, 332

A

add_suggestions
 weblate admin command, 327
 add_suggestions command line option
 --author USER@EXAMPLE.COM, 327
 ADMINS
 setting, 173
 AKISMET_API_KEY
 setting, 283
 ALLOWED_HOSTS
 setting, 173
 Android
 file format, 62
 ANONYMOUS_USER_NAME
 setting, 283
 API, 80, 125, 130
 Apple strings
 file format, 63
 ARB
 file format, 66
 AUDITLOG_EXPIRY
 setting, 284
 AUTH_LOCK_ATTEMPTS

 setting, 284
 AUTH_TOKEN_VALID
 setting, 285
 auto_translate
 weblate admin command, 327
 auto_translate command line option
 --add, 328
 --inconsistent, 328
 --mt MT, 328
 --overwrite, 327
 --source PROJECT/COMPONENT, 327
 --threshold THRESHOLD, 328
 --user USERNAME, 327
 AUTO_UPDATE
 setting, 284
 AUTOFIX_LIST
 setting, 285
 AVATAR_URL_PREFIX
 setting, 285

B

BASE_DIR
 setting, 286
 BaseAddon (*клас в weblate.addons.base*), 365
 BASIC_LANGUAGES
 setting, 286
 bilingual
 translation, 55

C

can_install() (*weblate.addons.base.BaseAddon class method*), 365
 celery_queues
 weblate admin command, 328
 changes
 wlc command line option, 128
 CHECK_LIST
 setting, 286
 checkgit
 weblate admin command, 328
 cleanup
 wlc command line option, 128
 cleanuptrans
 weblate admin command, 329
 Comma separated values
 file format, 68
 Command (*клас в wlc.main*), 132
 COMMENT_CLEANUP_DAYS
 setting, 287
 commit
 wlc command line option, 127
 commit_pending
 weblate admin command, 328
 commit_pending command line option
 --age HOURS, 328
 COMMIT_PENDING_HOURS
 setting, 287
 commitgit

weblate admin command, 328
 configure() (*weblate.addons.base.BaseAddon* *method*), 365
 createadmin
 weblate admin command, 329
 createadmin command line option
 --email USER@EXAMPLE.COM, 329
 --name, 329
 --no-password, 329
 --password PASSWORD, 329
 --update, 329
 --username USERNAME, 329
 CSP_CONNECT_SRC
 setting, 286
 CSP_FONT_SRC
 setting, 286
 CSP_IMG_SRC
 setting, 286
 CSP_SCRIPT_SRC
 setting, 286
 CSP_STYLE_SRC
 setting, 286
 CSV
 file format, 68

D

daily() (*weblate.addons.base.BaseAddon* *method*), 365
 DATA_DIR
 setting, 287
 DATABASE_BACKUP
 setting, 288
 DATABASES
 setting, 174
 DEBUG
 setting, 174
 DEFAULT_ACCESS_CONTROL
 setting, 288
 DEFAULT_ADD_MESSAGE
 setting, 289
 DEFAULT_ADDON_MESSAGE
 setting, 289
 DEFAULT_ADDONS
 setting, 289
 DEFAULT_COMMIT_MESSAGE
 setting, 289
 DEFAULT_COMMITTER_EMAIL
 setting, 289
 DEFAULT_COMMITTER_NAME
 setting, 290
 DEFAULT_DELETE_MESSAGE
 setting, 289
 DEFAULT_FROM_EMAIL
 setting, 174
 DEFAULT_LANGUAGE
 setting, 290
 DEFAULT_MERGE_MESSAGE
 setting, 289

DEFAULT_MERGE_STYLE
 setting, 290
 DEFAULT_PULL_MESSAGE
 setting, 290
 DEFAULT_RESTRICTED_COMPONENT
 setting, 289
 DEFAULT_TRANSLATION_PROPAGATION
 setting, 290
 download
 wlc command line option, 128
 DTD
 file format, 69
 dump_memory
 weblate admin command, 329
 dumpuserdata
 weblate admin command, 330

E

ENABLE_AVATARS
 setting, 291
 ENABLE_HOOKS
 setting, 291
 ENABLE_HTTPS
 setting, 291
 ENABLE_SHARING
 setting, 291

F

file format
 .XML resource file, 67
 Android, 62
 Apple strings, 63
 ARB, 66
 Comma separated values, 68
 CSV, 68
 DTD, 69
 gettext, 57
 go-i18n, 66
 GWT properties, 61
 i18next, 65
 INI translations, 61
 Java properties, 60
 Joomla translations, 62
 JSON, 64
 PHP strings, 63
 PO, 57
 Qt, 62
 RC, 70
 RESX, 67
 Ruby YAML, 69
 Ruby YAML Ain't Markup Language, 69
 string resources, 62
 TS, 62
 XLIFF, 58
 XML, 70
 YAML, 68
 YAML Ain't Markup Language, 68

G

`get()` (*wlc.Weblate* *method*), 131
`get_add_form()` (*weblate.addons.base.BaseAddon*
class method), 366
`get_settings_form()`
(weblate.addons.base.BaseAddon method),
366
`gettext`
file format, 57
GITHUB_CREDENTIALS
setting, 292
GITHUB_TOKEN
setting, 293
GITHUB_USERNAME
setting, 292
GITLAB_CREDENTIALS
setting, 291
GITLAB_TOKEN
setting, 292
GITLAB_USERNAME
setting, 292
`go-i18n`
file format, 66
GOOGLE_ANALYTICS_ID
setting, 293
GWT properties
file format, 61

H

HIDE_REPO_CREDENTIALS
setting, 293
HIDE_VERSION
setting, 293

I

`i18next`
file format, 65
`import_demo`
weblate admin command, 330
`import_json`
weblate admin command, 330
`import_json` command line option
--ignore, 330
--main-component COMPONENT, 330
--project PROJECT, 330
--update, 330
`import_memory`
weblate admin command, 331
`import_memory` command line option
--language-map LANGMAP, 331
`import_project`
weblate admin command, 331
`import_project` command line option
--base-file-template TEMPLATE, 331
--file-format FORMAT, 331
--language-regex REGEX, 332
--license NAME, 332
--license-url URL, 332

--main-component, 332
--name-template TEMPLATE, 331
--new-base-template TEMPLATE, 331
--vcs NAME, 332
`importuserdata`
weblate admin command, 333
`importusers`
weblate admin command, 333
`importusers` command line option
--check, 333
INI translations
file format, 61
`install_addon`
weblate admin command, 333
`install_addon` command line option
--addon ADDON, 333
--configuration CONFIG, 333
--update, 333
IP_BEHIND_REVERSE_PROXY
setting, 293
IP_PROXY_HEADER
setting, 294
IP_PROXY_OFFSET
setting, 294
iPad
translation, 63
iPhone
translation, 63

J

Java properties
file format, 60
Joomla translations
file format, 62
JSON
file format, 64

L

LEGAL_URL
setting, 294
LICENSE_EXTRA
setting, 295
LICENSE_FILTER
setting, 295
LICENSE_REQUIRED
setting, 295
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
setting, 296
`list_languages`
weblate admin command, 334
`list_translators`
weblate admin command, 334
`list_translators` command line option
--language-code, 334
`list_versions`
weblate admin command, 334
`list-components`
wlc command line option, 127

list-languages
 wlc command line option, 127

list-projects
 wlc command line option, 127

list-translations
 wlc command line option, 127

load() (*wlc.config.WeblateConfig.memod*), 131

loadpo
 weblate admin command, 334

loadpo command line option
 --force, 334
 --lang LANGUAGE, 334

LOCALIZE_CDN_PATH
 setting, 296

LOCALIZE_CDN_URL
 setting, 296

lock
 wlc command line option, 128

lock_translation
 weblate admin command, 335

lock-status
 wlc command line option, 128

LOGIN_REQUIRED_URLS
 setting, 296

LOGIN_REQUIRED_URLS_EXCEPTIONS
 setting, 296

ls
 wlc command line option, 127

M

MACHINE_TRANSLATION_SERVICES
 setting, 297

main() (*в модьні wlc.main*), 131

MATOMO_SITE_ID
 setting, 297

MATOMO_URL
 setting, 297

monolingual
 translation, 55

move_language
 weblate admin command, 335

MT_APERTIUM_APY
 setting, 298

MT_AWS_ACCESS_KEY_ID
 setting, 298

MT_AWS_REGION
 setting, 298

MT_AWS_SECRET_ACCESS_KEY
 setting, 298

MT_BAIDU_ID
 setting, 298

MT_BAIDU_SECRET
 setting, 299

MT_DEEPL_API_VERSION
 setting, 299

MT_DEEPL_KEY
 setting, 299

MT_GOOGLE_CREDENTIALS

 setting, 299

MT_GOOGLE_KEY
 setting, 299

MT_GOOGLE_LOCATION
 setting, 300

MT_GOOGLE_PROJECT
 setting, 300

MT_MICROSOFT_BASE_URL
 setting, 300

MT_MICROSOFT_COGNITIVE_KEY
 setting, 300

MT_MICROSOFT_ENDPOINT_URL
 setting, 300

MT_MICROSOFT_REGION
 setting, 300

MT_MODERNMT_KEY
 setting, 301

MT_MODERNMT_URL
 setting, 301

MT_MYMEMORY_EMAIL
 setting, 301

MT_MYMEMORY_KEY
 setting, 301

MT_MYMEMORY_USER
 setting, 301

MT_NETEASE_KEY
 setting, 301

MT_NETEASE_SECRET
 setting, 302

MT_SAP_BASE_URL
 setting, 302

MT_SAP_PASSWORD
 setting, 303

MT_SAP_SANDBOX_APIKEY
 setting, 303

MT_SAP_USE_MT
 setting, 303

MT_SAP_USERNAME
 setting, 303

MT_SERVICES
 setting, 297

MT_TMSERVER
 setting, 302

MT_YANDEX_KEY
 setting, 302

MT_YOUDAO_ID
 setting, 302

MT_YOUDAO_SECRET
 setting, 302

N

NEARBY_MESSAGES
 setting, 303

P

PAGURE_CREDENTIALS
 setting, 303

PAGURE_TOKEN

setting, 304
 PAGURE_USERNAME
 setting, 304
 PHP strings
 file format, 63
 PIWIK_SITE_ID
 setting, 297
 PIWIK_URL
 setting, 297
 PO
 file format, 57
 post() (wlc. *Weblate* *method*), 131
 post_add() (weblate.addons.base.*BaseAddon*
 method), 366
 post_commit() (weblate.addons.base.*BaseAddon*
 method), 366
 post_push() (weblate.addons.base.*BaseAddon* *me-*
 mod), 366
 post_update() (weblate.addons.base.*BaseAddon*
 method), 366
 pre_commit() (weblate.addons.base.*BaseAddon* *me-*
 mod), 366
 pre_push() (weblate.addons.base.*BaseAddon*
 method), 366
 pre_update() (weblate.addons.base.*BaseAddon* *me-*
 mod), 366
 pull
 wlc command line option, 128
 push
 wlc command line option, 128
 pushgit
 weblate admin command, 335
 pushgit command line option
 --force-commit, 335
 Python, 130

Q

Qt
 file format, 62

R

RATELIMIT_ATTEMPTS
 setting, 304
 RATELIMIT_LOCKOUT
 setting, 304
 RATELIMIT_WINDOW
 setting, 304
 RC
 file format, 70
 register_command() (в *модулі* *wlc.main*), 131
 REGISTRATION_ALLOW_BACKENDS
 setting, 305
 REGISTRATION_CAPTCHA
 setting, 305
 REGISTRATION_EMAIL_MATCH
 setting, 305
 REGISTRATION_OPEN
 setting, 305

repo
 wlc command line option, 128
 REPOSITORY_ALERT_THRESHOLD
 setting, 306
 REQUIRE_LOGIN
 setting, 306
 reset
 wlc command line option, 128
 REST, 80
 RESX
 file format, 67
 RFC
 RFC 4646, 54
 Ruby YAML
 file format, 69
 Ruby YAML Ain't Markup Language
 file format, 69

S

save_state() (weblate.addons.base.*BaseAddon* *me-*
 mod), 366
 SECRET_KEY
 setting, 174
 SENTRY_DSN
 setting, 306
 SERVER_EMAIL
 setting, 174
 SESSION_COOKIE_AGE_AUTHENTICATED
 setting, 306
 SESSION_ENGINE
 setting, 174
 setting
 ADMINS, 173
 AKISMET_API_KEY, 283
 ALLOWED_HOSTS, 173
 ANONYMOUS_USER_NAME, 283
 AUDITLOG_EXPIRY, 284
 AUTH_LOCK_ATTEMPTS, 284
 AUTH_TOKEN_VALID, 285
 AUTO_UPDATE, 284
 AUTOFIX_LIST, 285
 AVATAR_URL_PREFIX, 285
 BASE_DIR, 286
 BASIC_LANGUAGES, 286
 CHECK_LIST, 286
 COMMENT_CLEANUP_DAYS, 287
 COMMIT_PENDING_HOURS, 287
 CSP_CONNECT_SRC, 286
 CSP_FONT_SRC, 286
 CSP_IMG_SRC, 286
 CSP_SCRIPT_SRC, 286
 CSP_STYLE_SRC, 286
 DATA_DIR, 287
 DATABASE_BACKUP, 288
 DATABASES, 174
 DEBUG, 174
 DEFAULT_ACCESS_CONTROL, 288
 DEFAULT_ADD_MESSAGE, 289

DEFAULT_ADDON_MESSAGE, 289
DEFAULT_ADDONS, 289
DEFAULT_COMMIT_MESSAGE, 289
DEFAULT_COMMITER_EMAIL, 289
DEFAULT_COMMITER_NAME, 290
DEFAULT_DELETE_MESSAGE, 289
DEFAULT_FROM_EMAIL, 174
DEFAULT_LANGUAGE, 290
DEFAULT_MERGE_MESSAGE, 289
DEFAULT_MERGE_STYLE, 290
DEFAULT_PULL_MESSAGE, 290
DEFAULT_RESTRICTED_COMPONENT, 289
DEFAULT_TRANSLATION_PROPAGATION, 290
ENABLE_AVATARS, 291
ENABLE_HOOKS, 291
ENABLE_HTTPS, 291
ENABLE_SHARING, 291
GITHUB_CREDENTIALS, 292
GITHUB_TOKEN, 293
GITHUB_USERNAME, 292
GITLAB_CREDENTIALS, 291
GITLAB_TOKEN, 292
GITLAB_USERNAME, 292
GOOGLE_ANALYTICS_ID, 293
HIDE_REPO_CREDENTIALS, 293
HIDE_VERSION, 293
IP_BEHIND_REVERSE_PROXY, 293
IP_PROXY_HEADER, 294
IP_PROXY_OFFSET, 294
LEGAL_URL, 294
LICENSE_EXTRA, 295
LICENSE_FILTER, 295
LICENSE_REQUIRED, 295
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 296
LOCALIZE_CDN_PATH, 296
LOCALIZE_CDN_URL, 296
LOGIN_REQUIRED_URLS, 296
LOGIN_REQUIRED_URLS_EXCEPTIONS, 296
MACHINE_TRANSLATION_SERVICES, 297
MATOMO_SITE_ID, 297
MATOMO_URL, 297
MT_APERTIUM_APY, 298
MT_AWS_ACCESS_KEY_ID, 298
MT_AWS_REGION, 298
MT_AWS_SECRET_ACCESS_KEY, 298
MT_BAIDU_ID, 298
MT_BAIDU_SECRET, 299
MT_DEEPL_API_VERSION, 299
MT_DEEPL_KEY, 299
MT_GOOGLE_CREDENTIALS, 299
MT_GOOGLE_KEY, 299
MT_GOOGLE_LOCATION, 300
MT_GOOGLE_PROJECT, 300
MT_MICROSOFT_BASE_URL, 300
MT_MICROSOFT_COGNITIVE_KEY, 300
MT_MICROSOFT_ENDPOINT_URL, 300
MT_MICROSOFT_REGION, 300
MT_MODERNMT_KEY, 301
MT_MODERNMT_URL, 301
MT_MYMEMORY_EMAIL, 301
MT_MYMEMORY_KEY, 301
MT_MYMEMORY_USER, 301
MT_NETEASE_KEY, 301
MT_NETEASE_SECRET, 302
MT_SAP_BASE_URL, 302
MT_SAP_PASSWORD, 303
MT_SAP_SANDBOX_APIKEY, 303
MT_SAP_USE_MT, 303
MT_SAP_USERNAME, 303
MT_SERVICES, 297
MT_TMSERVER, 302
MT_YANDEX_KEY, 302
MT_YOUDAO_ID, 302
MT_YOUDAO_SECRET, 302
NEARBY_MESSAGES, 303
PAGURE_CREDENTIALS, 303
PAGURE_TOKEN, 304
PAGURE_USERNAME, 304
PIWIK_SITE_ID, 297
PIWIK_URL, 297
RATELIMIT_ATTEMPTS, 304
RATELIMIT_LOCKOUT, 304
RATELIMIT_WINDOW, 304
REGISTRATION_ALLOW_BACKENDS, 305
REGISTRATION_CAPTCHA, 305
REGISTRATION_EMAIL_MATCH, 305
REGISTRATION_OPEN, 305
REPOSITORY_ALERT_THRESHOLD, 306
REQUIRE_LOGIN, 306
SECRET_KEY, 174
SENTRY_DSN, 306
SERVER_EMAIL, 174
SESSION_COOKIE_AGE_AUTHENTICATED, 306
SESSION_ENGINE, 174
SIMPLIFY_LANGUAGES, 307
SINGLE_PROJECT, 308
SITE_DOMAIN, 307
SITE_TITLE, 307
SPECIAL_CHARS, 308
STATUS_URL, 308
SUGGESTION_CLEANUP_DAYS, 308
UPDATE_LANGUAGES, 308
URL_PREFIX, 308
VCS_BACKENDS, 309
VCS_CLONE_DEPTH, 309
WEBLATE_ADDONS, 310
WEBLATE_EXPORTERS, 310
WEBLATE_FORMATS, 310
WEBLATE_GPG_IDENTITY, 311
setupgroups
 weblate admin command, 336
setupgroups command line option
 --no-privs-update, 336

- no-projects-update, 336
- setuplang
 - weblate admin command, 336
- setuplang command line option
 - no-update, 336
- show
 - wlc command line option, 127
- SIMPLIFY_LANGUAGES
 - setting, 307
- SINGLE_PROJECT
 - setting, 308
- SITE_DOMAIN
 - setting, 307
- SITE_TITLE
 - setting, 307
- SPECIAL_CHARS
 - setting, 308
- statistics
 - wlc command line option, 128
- STATUS_URL
 - setting, 308
- stay_on_create (*weblate.addons.base.BaseAddon* *ампубум*), 366
- store_post_load()
 - (*weblate.addons.base.BaseAddon* *метод*), 366
- string resources
 - file format, 62
- SUGGESTION_CLEANUP_DAYS
 - setting, 308

T

- translation
 - bilingual, 55
 - iPad, 63
 - iPhone, 63
 - monolingual, 55
- TS
 - file format, 62

U

- unit_pre_create()
 - (*weblate.addons.base.BaseAddon* *метод*), 366
- unlock
 - wlc command line option, 128
- unlock_translation
 - weblate admin command, 336
- UPDATE_LANGUAGES
 - setting, 308
- updatechecks
 - weblate admin command, 336
- updategit
 - weblate admin command, 337
- upload
 - wlc command line option, 128
- URL_PREFIX
 - setting, 308

V

- VCS_BACKENDS
 - setting, 309
- VCS_CLONE_DEPTH
 - setting, 309
- version
 - wlc command line option, 127

W

- Weblate (*клас в wlc*), 131
- weblate admin command
 - add_suggestions, 327
 - auto_translate, 327
 - celery_queues, 328
 - checkgit, 328
 - cleanup_trans, 329
 - commit_pending, 328
 - commitgit, 328
 - createadmin, 329
 - dump_memory, 329
 - dumpuserdata, 330
 - import_demo, 330
 - import_json, 330
 - import_memory, 331
 - import_project, 331
 - importuserdata, 333
 - importusers, 333
 - install_addon, 333
 - list_languages, 334
 - list_translators, 334
 - list_versions, 334
 - loadpo, 334
 - lock_translation, 335
 - move_language, 335
 - pushgit, 335
 - setupgroups, 336
 - setuplang, 336
 - unlock_translation, 336
 - updatechecks, 336
 - updategit, 337
- WEBLATE_ADDONS
 - setting, 310
- WEBLATE_ADMIN_EMAIL, 137, 138, 142
- WEBLATE_ADMIN_NAME, 137, 138
- WEBLATE_ADMIN_PASSWORD, 134, 137, 138
- WEBLATE_ALLOWED_HOSTS, 137, 173, 178, 307
- WEBLATE_EMAIL_PORT, 148
- WEBLATE_EMAIL_USE_SSL, 148
- WEBLATE_EMAIL_USE_TLS, 148
- WEBLATE_EXPORTERS
 - setting, 310
- WEBLATE_FORMATS
 - setting, 310
- WEBLATE_GPG_IDENTITY
 - setting, 311
- WEBLATE_LOCALIZE_CDN_PATH, 149
- WEBLATE_REQUIRE_LOGIN, 306
- WEBLATE_SILENCED_SYSTEM_CHECKS, 201

WEBLATE_SITE_DOMAIN, 176, 194
 WeblateConfig (клас в *wlc.config*), 131
 WeblateException, 130
 wlc, 125
 модуль, 130
 wlc command line option
 --config PATH, 127
 --config-section SECTION, 127
 --convert, 128
 --format {csv,json,text,html}, 127
 --input, 128
 --key KEY, 127
 --output, 128
 --overwrite, 128
 --url URL, 127
 changes, 128
 cleanup, 128
 commit, 127
 download, 128
 list-components, 127
 list-languages, 127
 list-projects, 127
 list-translations, 127
 lock, 128
 lock-status, 128
 ls, 127
 pull, 128
 push, 128
 repo, 128
 reset, 128
 show, 127
 statistics, 128
 unlock, 128
 upload, 128
 version, 127
 wlc.config
 модуль, 131
 wlc.main
 модуль, 131

X

XLIFF
 file format, 58
 XML
 file format, 70

Y

YAML
 file format, 68
 YAML Ain't Markup Language
 file format, 68



змінна оточення
 CELERY_BACKUP_OPTIONS, 150
 CELERY_BEAT_OPTIONS, 150
 CELERY_MAIN_OPTIONS, 150
 CELERY_MEMORY_OPTIONS, 150

CELERY_NOTIFY_OPTIONS, 150
 CELERY_TRANSLATE_OPTIONS, 150
 POSTGRES_ALTER_ROLE, 146
 POSTGRES_DATABASE, 146
 POSTGRES_HOST, 146
 POSTGRES_PASSWORD, 146
 POSTGRES_PORT, 146
 POSTGRES_SSL_MODE, 146
 POSTGRES_USER, 146
 REDIS_DB, 147
 REDIS_HOST, 147
 REDIS_PASSWORD, 147
 REDIS_PORT, 147
 REDIS_TLS, 147
 REDIS_VERIFY_SSL, 147
 ROLLBAR_ENVIRONMENT, 148
 ROLLBAR_KEY, 148
 SENTRY_DSN, 148
 SENTRY_ENVIRONMENT, 149
 SOCIAL_AUTH_SLACK_SECRET, 146
 UWSGI_WORKERS, 150
 WEBLATE_ADD_ADDONS, 149
 WEBLATE_ADD_APPS, 149
 WEBLATE_ADD_AUTOFIX, 149
 WEBLATE_ADD_CHECK, 149
 WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 140
 WEBLATE_ADMIN_EMAIL, 137, 138, 142
 WEBLATE_ADMIN_NAME, 137, 138
 WEBLATE_ADMIN_PASSWORD, 134, 137, 138
 WEBLATE_AKISMET_API_KEY, 141
 WEBLATE_ALLOWED_HOSTS, 137, 138, 173, 178, 307
 WEBLATE_AUTH_LDAP_BIND_DN, 143
 WEBLATE_AUTH_LDAP_BIND_PASSWORD, 143
 WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS, 143
 WEBLATE_AUTH_LDAP_SERVER_URI, 143
 WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 143
 WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 143
 WEBLATE_AUTH_LDAP_USER_SEARCH, 143
 WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 143
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 143
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 143
 WEBLATE_BASIC_LANGUAGES, 141
 WEBLATE_CSP_CONNECT_SRC, 141
 WEBLATE_CSP_FONT_SRC, 141
 WEBLATE_CSP_IMG_SRC, 141
 WEBLATE_CSP_SCRIPT_SRC, 141
 WEBLATE_CSP_STYLE_SRC, 141
 WEBLATE_DATABASE_BACKUP, 147
 WEBLATE_DEBUG, 137

WEBLATE_DEFAULT_ACCESS_CONTROL, 141
 WEBLATE_DEFAULT_COMMITER_EMAIL, 141
 WEBLATE_DEFAULT_COMMITER_NAME, 141
 WEBLATE_DEFAULT_FROM_EMAIL, 138
 WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 141
 WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 141
 WEBLATE_EMAIL_BACKEND, 148
 WEBLATE_EMAIL_HOST, 147
 WEBLATE_EMAIL_HOST_PASSWORD, 148
 WEBLATE_EMAIL_HOST_USER, 148
 WEBLATE_EMAIL_PORT, 148
 WEBLATE_EMAIL_USE_SSL, 148
 WEBLATE_EMAIL_USE_TLS, 148
 WEBLATE_ENABLE_HTTPS, 139
 WEBLATE_GITHUB_TOKEN, 140
 WEBLATE_GITHUB_USERNAME, 140
 WEBLATE_GITLAB_TOKEN, 140
 WEBLATE_GITLAB_USERNAME, 140
 WEBLATE_GOOGLE_ANALYTICS_ID, 140
 WEBLATE_GPG_IDENTITY, 141
 WEBLATE_HIDE_VERSION, 141
 WEBLATE_IP_PROXY_HEADER, 139
 WEBLATE_LICENSE_FILTER, 141
 WEBLATE_LOCALIZE_CDN_PATH, 149
 WEBLATE_LOCALIZE_CDN_URL, 149
 WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 140
 WEBLATE_LOGLEVEL, 137
 WEBLATE_MT_APERTIUM_API, 142
 WEBLATE_MT_AWS_ACCESS_KEY_ID, 142
 WEBLATE_MT_AWS_REGION, 142
 WEBLATE_MT_AWS_SECRET_ACCESS_KEY, 142
 WEBLATE_MT_DEEPL_API_VERSION, 142
 WEBLATE_MT_DEEPL_KEY, 142
 WEBLATE_MT_GLOSBE_ENABLED, 142
 WEBLATE_MT_GOOGLE_KEY, 142
 WEBLATE_MT_MICROSOFT_BASE_URL, 142
 WEBLATE_MT_MICROSOFT_COGNITIVE_KEY, 142
 WEBLATE_MT_MICROSOFT_ENDPOINT_URL, 142
 WEBLATE_MT_MICROSOFT_REGION, 142
 WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED, 142
 WEBLATE_MT_MODERNMT_KEY, 142
 WEBLATE_MT_MMEMORY_ENABLED, 142
 WEBLATE_MT_SAP_BASE_URL, 142
 WEBLATE_MT_SAP_PASSWORD, 143
 WEBLATE_MT_SAP_SANDBOX_APIKEY, 142
 WEBLATE_MT_SAP_USE_MT, 143
 WEBLATE_MT_SAP_USERNAME, 143
 WEBLATE_NO_EMAIL_AUTH, 146
 WEBLATE_PAGURE_TOKEN, 140
 WEBLATE_PAGURE_USERNAME, 140
 WEBLATE_REGISTRATION_ALLOW_BACKENDS, 138
 WEBLATE_REGISTRATION_OPEN, 138
 WEBLATE_REMOVE_ADDONS, 149
 WEBLATE_REMOVE_APPS, 149
 WEBLATE_REMOVE_AUTOFIX, 149
 WEBLATE_REMOVE_CHECK, 149
 WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 140
 WEBLATE_REQUIRE_LOGIN, 140, 306
 WEBLATE_SAML_IDP_ENTITY_ID, 146
 WEBLATE_SAML_IDP_URL, 146
 WEBLATE_SAML_IDP_X509CERT, 146
 WEBLATE_SECURE_PROXY_SSL_HEADER, 139
 WEBLATE_SERVER_EMAIL, 138
 WEBLATE_SILENCED_SYSTEM_CHECKS, 141, 201
 WEBLATE_SIMPLIFY_LANGUAGES, 141
 WEBLATE_SITE_DOMAIN, 137, 176, 194
 WEBLATE_SITE_TITLE, 137
 WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 145
 WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 145
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 145
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET, 145
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID, 145
 WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 144
 WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, 144
 WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 144
 WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 144
 WEBLATE_SOCIAL_AUTH_FEDORA, 145
 WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 144
 WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 144
 WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, 145
 WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 145
 WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, 145
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, 144
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, 144
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, 144
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS, 144
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, 145

WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM,
145
WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL,
145
WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY,
145
WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY,
145
WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET,
145
WEBLATE_SOCIAL_AUTH_OPENSUSE, 145
WEBLATE_SOCIAL_AUTH_SLACK_KEY, 146
WEBLATE_SOCIAL_AUTH_UBUNTU, 145
WEBLATE_TIME_ZONE, 139
WEBLATE_URL_PREFIX, 141
WL_BRANCH, 280
WL_COMPONENT_NAME, 280
WL_COMPONENT_SLUG, 280
WL_COMPONENT_URL, 280
WL_ENGAGE_URL, 280
WL_FILE_FORMAT, 280
WL_FILEMASK, 280
WL_LANGUAGE, 280
WL_NEW_BASE, 280
WL_PATH, 280
WL_PREVIOUS_HEAD, 280
WL_PROJECT_NAME, 280
WL_PROJECT_SLUG, 280
WL_REPO, 280
WL_TEMPLATE, 280
WL_VCS, 280
модуль
wlc, 130
wlc.config, 131
wlc.main, 131