



# **The Weblate Manual**

***Реліз 4.2***

**Michal Čihař**

серп. 18, 2020

<b>1</b>	<b>Документація для користувачів</b>	<b>1</b>
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	22
1.6	Пошук	37
1.7	Application developer guide	41
1.8	Процеси перекладу	61
1.9	Поширені питання та відповіді на них	65
1.10	Підтримувані формати файлів	72
1.11	Налаштування інтеграції із керуванням версіями	91
1.12	Weblate's REST API	98
1.13	Клієнт Weblate	138
1.14	Програмний інтерфейс Python Weblate	142
<b>2</b>	<b>Документація для адміністраторів</b>	<b>144</b>
2.1	Настанови з налаштування	144
2.2	Розгортання Weblate	199
2.3	Оновлення Weblate	199
2.4	Резервне копіювання і пересування Weblate	204
2.5	Розпізнавання	209
2.6	Керування доступом	218
2.7	Проекти перекладу	226
2.8	Визначення мов	242
2.9	Безперервна локалізація	244
2.10	Ліцензування перекладів	252
2.11	Процес перекладу	254
2.12	Перевірки і виправлення	260
2.13	Машинний переклад	268
2.14	Доповнення	275
2.15	Пам'ять перекладів	284
2.16	Налаштування	286
2.17	Зразок налаштувань	310
2.18	Команди керування	325
2.19	Оголошення	336
2.20	Списки складників	338
2.21	Додаткові модулі Weblate	339
2.22	Налаштовування Weblate	344
2.23	Інтерфейс керування	346
2.24	Отримання підтримки щодо Weblate	353

2.25	Правові документи . . . . .	354
<b>3</b>	<b>Документація для учасників . . . . .</b>	<b>356</b>
3.1	Contributing to Weblate . . . . .	356
3.2	Starting contributing code to Weblate . . . . .	357
3.3	Початковий код Weblate . . . . .	361
3.4	Діагностика у Weblate . . . . .	362
3.5	Нутроші Weblate . . . . .	363
3.6	Оболонка Weblate . . . . .	364
3.7	Надсилання звітів щодо вад у Weblate . . . . .	365
3.8	Weblate testsuite and continuous integration . . . . .	365
3.9	Data schemas . . . . .	367
3.10	Створення випусків Weblate . . . . .	371
3.11	Про Weblate . . . . .	371
3.12	Ліцензія . . . . .	372
<b>4</b>	<b>Журнал змін . . . . .</b>	<b>373</b>
4.1	Weblate 4.2 . . . . .	373
4.2	Weblate 4.1.1 . . . . .	374
4.3	Weblate 4.1 . . . . .	374
4.4	Weblate 4.0.4 . . . . .	375
4.5	Weblate 4.0.3 . . . . .	376
4.6	Weblate 4.0.2 . . . . .	376
4.7	Weblate 4.0.1 . . . . .	376
4.8	Weblate 4.0 . . . . .	377
4.9	Weblate 3.x series . . . . .	377
4.10	Weblate 2.x series . . . . .	389
4.11	Weblate 1.x series . . . . .	400
4.12	Weblate 0.x series . . . . .	404
	<b>Python Module Index . . . . .</b>	<b>408</b>
	<b>HTTP Routing Table . . . . .</b>	<b>409</b>
	<b>Індекс . . . . .</b>	<b>411</b>

## 1.1 Основи Weblate

### 1.1.1 Структура проєкту

In Weblate translations are organized into projects and components. Each project can contain number of components and those contain translations into individual languages. The component corresponds to one translatable file (for example *GNU Gettext* or *Рядкові ресурси Android*). The projects are there to help you organize component into logical sets (for example to group all translations used within one application).

Internally, each project has translations to common strings propagated across other components within it by default. This lightens the burden of repetitive and multi version translation. Disable it as per *Налаштування складників*, still producing errors for seemingly inconsistent resulting translations.

## 1.2 Реєстрація і профіль користувача

### 1.2.1 Реєстрація

Everybody can browse projects, view translations or suggest translations by default. Only registered users are allowed to actually save changes, and are credited for every translation made.

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.

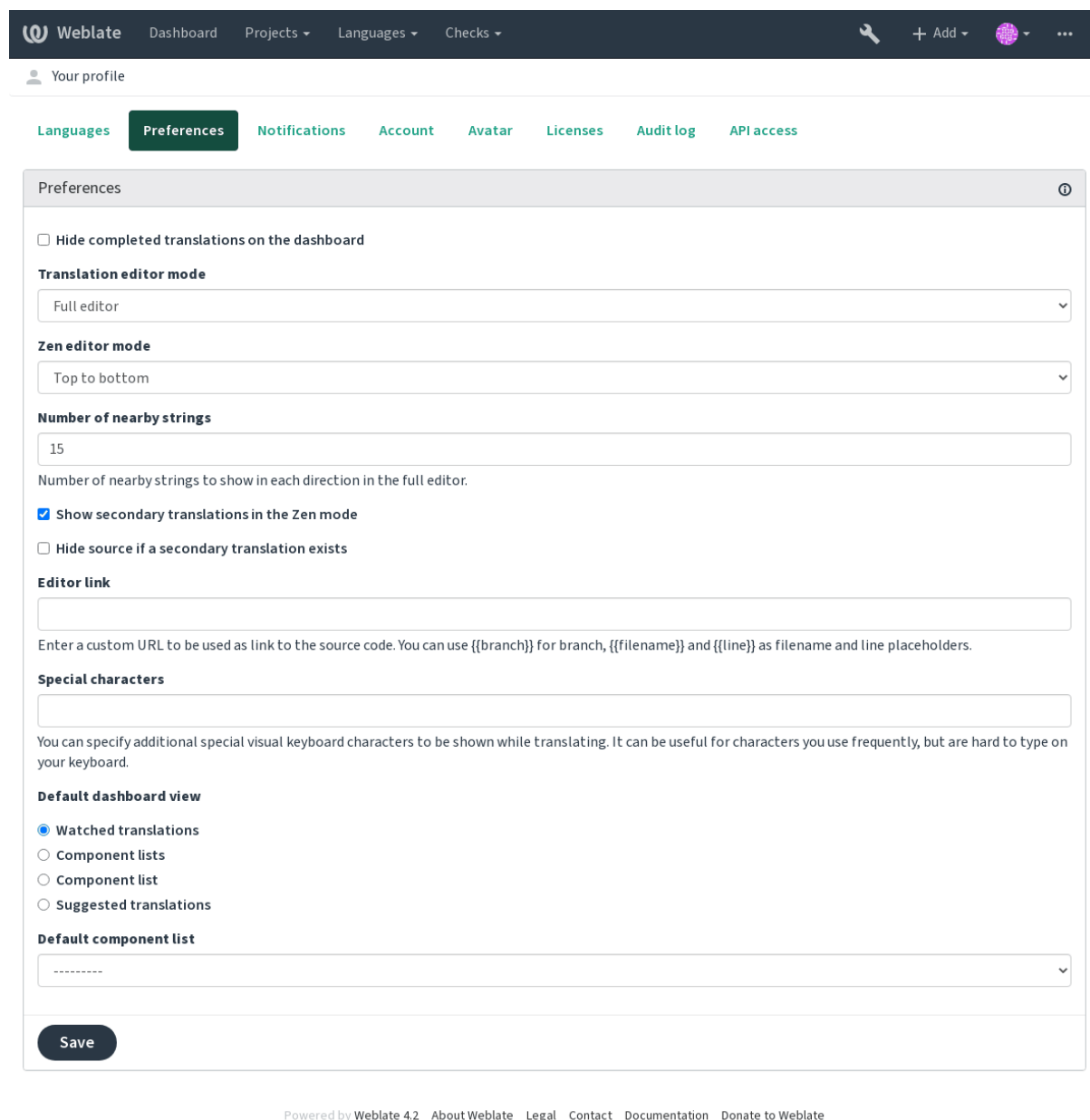
### 1.2.2 Домівка

When you sign in, you will see an overview of projects and components, as well as their respective translation progression.

Нове в версії 2.5.

Components of projects you are watching are shown by default, and cross-referenced with your preferred languages.

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



The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo, 'Dashboard', 'Projects', 'Languages', and 'Checks' menus. On the right are icons for settings, adding new items, a user profile, and a menu. Below the navigation bar is a light-colored header with a 'Your profile' link and a row of tabs: 'Languages', 'Preferences' (which is active and highlighted in green), 'Notifications', 'Account', 'Avatar', 'Licenses', 'Audit log', and 'API access'. The main content area is titled 'Preferences' and contains several settings sections: 'Hide completed translations on the dashboard' (unchecked), 'Translation editor mode' (set to 'Full editor'), 'Zen editor mode' (set to 'Top to bottom'), 'Number of nearby strings' (set to 15), 'Show secondary translations in the Zen mode' (checked), 'Hide source if a secondary translation exists' (unchecked), 'Editor link' (empty text field), 'Special characters' (empty text field), 'Default dashboard view' (radio buttons for 'Watched translations', 'Component lists', 'Component list', and 'Suggested translations', with 'Watched translations' selected), and 'Default component list' (dropdown menu). A 'Save' button is at the bottom left of the preferences panel. At the very bottom of the page, a small footer reads 'Powered by Weblate 4.2' followed by links for 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

У меню ви побачите такі пункти:

- *Projects > Browse all projects* in the main menu showing translation status for each project on the Weblate instance.
- Selecting a language in the main menu *Languages* will show translation status of all projects, filtered by one of your primary languages.

- *Watched translations* in the Dashboard will show translation status of only those projects you are watching, filtered by your primary languages.

In addition, the drop-down can also show any number of *component lists*, sets of project components preconfigured by the Weblate administrator, see [Списки складників](#).

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

---

**Примітка:** When Weblate is configured for a single project using `SINGLE_PROJECT` in the `settings.py` file (see [Налаштування](#)), the dashboard will not be shown, as the user will be redirected to a single project or component instead.

---

### 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:** All language selections only offer currently translated languages.

---

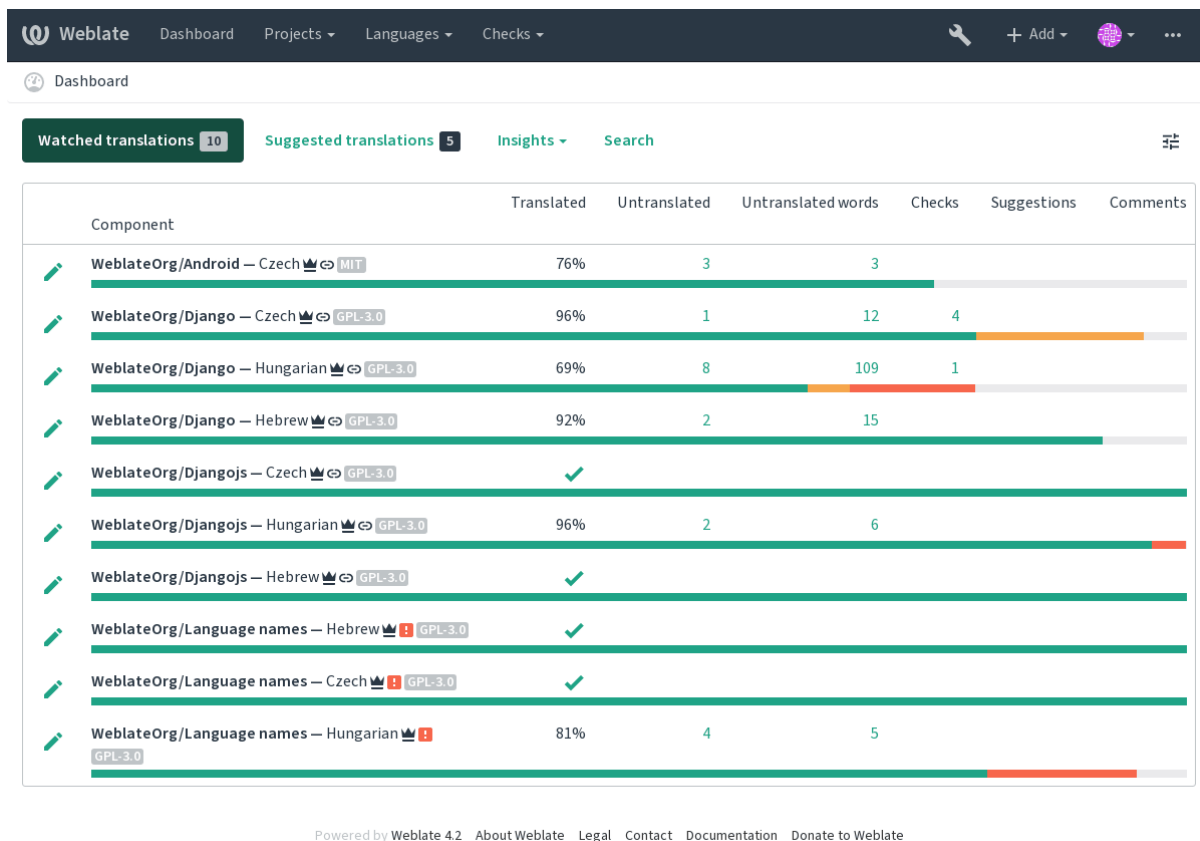
---

**Підказка:** Request or add other languages you want to translate by clicking the button to make them available too.

---

#### Перекладено мовами

Choose which languages you prefer to translate, and they will be offered on the main page of watched projects, so that you have easier access to these all translations in each of those languages.



## Другорядні мови

You can define which secondary languages are shown to you as a guide while translating. An example can be seen in the following image, where the Hebrew language is shown as secondarily:

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, a breadcrumb shows 'WeblateOrg / Django / Czech / Translate'. A status bar indicates 'translated 96%'. The main interface has a top bar with navigation icons and a 'Position and priority' dropdown. The central part is the 'Translation' editor for the string 'Files'. It shows the English source 'Files', the Hebrew translation 'קבצים', and the Czech translation 'Soubory'. There are buttons for 'Save', 'Suggest', and 'Skip'. Below the editor, there are tabs for 'Nearby strings' (16), 'Comments', 'Machinery', 'Other languages' (selected), and 'History'. A table below shows the string in English, Hebrew, and Hungarian with their respective statuses and edit buttons. The right sidebar contains a 'Glossary' section, 'Source information' with a 'Screenshot context' section, and 'Source string location'.

## Усталений вигляд домівки

On the *Preferences* tab, you can pick which of the available dashboard views to present by default. If you pick the *Component list*, you have to select which component list will be displayed from the *Default component list* drop-down.

Дивись також:

*Списки складників*

## Аватар

Avatar can be shown for each user (depending on `ENABLE_AVATARS`). These images are obtained using <https://gravatar.com/>.

## Посилання на редактор

A source code link is shown in the web-browser configured in the *Налаштування складників* by default.

**Підказка:** By setting the *Editor link*, you use your local editor to open the VCS source code file of translated strings. You can use *Розмітка шаблонів*.

Usually something like `editor://open/?file={{filename}}&line={{line}}` is a good option.

Дивись також:



You can find more info on registering custom URL protocols for the editor in the [Nette documentation](#).

### 1.2.4 Сповіщення

Subscribe to various notifications from the *Notifications* tab. Notifications for selected events on watched or administered projects will be sent to you per e-mail.

Some of the notifications are sent only for events in your languages (for example about new strings to translate), while some trigger at component level (for example merge errors). These two groups of notifications are visually separated in the settings.

You can toggle notifications for watched projects and administered projects and it can be further tweaked (or muted) per project and component. Visit the component page and select appropriate choice from the *Watching* menu.

---

**Примітка:** Ви не отримуватиме сповіщень щодо ваших власних дій.

---

Powered by [Weblate 4.2](#) [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)


### 1.2.5 Обліковий запис

The *Account* tab lets you set up basic account details, connect various services you can use to sign in into Weblate, completely remove your account, or download your user data (see [Weblate user data export](#)).


---


**Примітка:** The list of services depends on your Weblate configuration, but can be made to include popular sites such as GitLab, GitHub, Google, Facebook, or Bitbucket or other OAuth 2.0 providers.

---


 Weblate

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

[+ Add](#)  [...](#)

 Your profile

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

Account 

**Username**

testuser

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

**Full name**

Weblate Test


**E-mail**






weblate@example.org

You can add another e-mail address below.


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

Save

Current user identities 

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

Add new association

 E-mail

Removal

Account removal deletes all your private data.

Remove my account

User data

You can download all your private data.

Download user data

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

## 1.2.6 Часопис перевірок

Audit log keeps track of the actions performed with your account. It logs IP address and browser for every important action with your account. The critical actions also trigger a notification to a primary e-mail address.

**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:

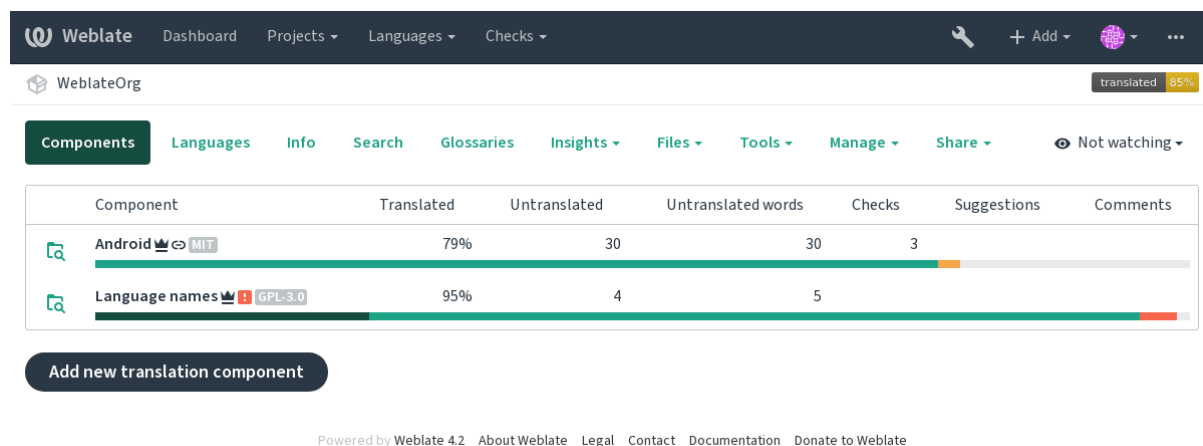
- Видимий для усіх, участь може взяти будь-хто
- Видимий лише певній групі перекладачів

**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

Having navigated to a component, a set of links lead to actual translation. The translation is further divided into individual checks, like *Untranslated* or *Needing review*. If the whole project is translated, without error, *All translations* is still available. Alternatively you can use the search field to find a specific string or term.

The screenshot shows the Weblate web interface for the Django project in Czech. The top navigation bar includes links for Dashboard, Projects, Languages, and Checks. The sidebar on the left has tabs for Overview, Info, Search, Glossary, Insights, Files, Tools, Manage, and Share. The main content area displays the translation status for the Django project in Czech, showing 26 strings (96% translated) and 183 words (93% translated). Below this, the Strings status section lists various categories of strings, such as 'All strings — 183 words', 'Translated strings — 171 words', and 'Strings needing action — 12 words'. The Other components section shows a table of components with their translation status, including Android, Language names, and Djangojs.

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

### 1.3.3 Пропозиції

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

Anonymous users can only (if permitted) forward suggestions. Doing so is still available to signed in users, in cases where uncertainty about the translation arises, which will prompt another translator to review it.

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

### 1.3.4 Коментарі

The comments can be posted in two scopes - source string or translation. Choose the one which matches the topic you want to discuss. The source string comments are good for providing feedback on the original string, for example that it should be rephrased or it is confusing.

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

### 1.3.5 Варіанти

Variants are used to group variants of the string in different lengths. The frontend can use different strings depending on the screen or window size.

**Дивись також:**

*Варіанти рядків*

### 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:**

*Мітки рядків*

### 1.3.7 Переклад

On the translation page, the source string and an edit area for translating are shown. Should the translation be plural, multiple source strings and edit areas are shown, each described and labeled in plural form.

All special whitespace characters are underlined in red and indicated with grey symbols. More than one subsequent space is also underlined in red to alert the translator to a potential formatting issue.

Various bits of extra information can be shown on this page, most of which coming from the project source code (like context, comments or where the message is being used). When you choose secondary languages in your preferences, translation to these languages will be shown (see *Другорядні мови*) above the source string.

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

### Множина

Words that change form to account of their numeric designation are called plurals. Each language has its own definition of plurals. English, for example, supports one plural. In the singular definition of for example «car», implicitly one car is referenced, in the plural definition, «cars» two or more cars are referenced, or the concept of cars as a noun. Languages like for example Czech or Arabic have more plurals and also their rules for plurals are different.

Weblate has full support for each of these forms, in each respective language by translating every plural separately. The number of fields and how it is used in the translated application depends on the configured plural formula. Weblate shows the basic information, but you can find a more detailed description in the [Language Plural Rules](#) by the Unicode Consortium.

**Дивись також:**

*Форма множини*

The screenshot displays the Weblate web interface for a translation project. The top navigation bar includes links for Dashboard, Projects, Languages, and Checks. The main area shows a translation for the string "%(count)s word" in English, with corresponding translations in Czech for singular, plural, and other forms. The interface includes a search bar, a sidebar with a glossary, source information, and a comments section. The bottom of the page features a footer with links to the Weblate documentation and a donation page.

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

## 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:

**Alt+Home** Перейти до першого перекладу у поточних результатах пошуку.

**Alt+End** Перейти до останнього перекладу у поточних результатах пошуку.

**Alt+PageUp** Перейти до попереднього перекладу у поточних результатах пошуку.



**Alt+PageDown** Перейти до наступного перекладу у поточних результатах пошуку.

**Alt+Enter, 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.

**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 комп'ютерного перекладу.

**Ctrl+<ЧИСЛО> або Cmd+<ЧИСЛО>** Копіює заміник із вказаним номером з початкового рядка.

**Ctrl+M <ЧИСЛО> або Cmd+M <ЧИСЛО>** Копіює комп'ютерний переклад із вказаним номером до поточного перекладу.

**Ctrl+I <ЧИСЛО> або Cmd+I <ЧИСЛО>** Ігнорувати один запис у списку непройдених перевірок.

**Ctrl+J або Cmd+J** Показує вкладку *Сусідні рядки*.

**Ctrl+S або Cmd+S** Показує вкладку пошуку.

**Ctrl+O або Cmd+O** Копіює початковий рядок.

**Ctrl+Y або Cmd+Y** Перемикає прапорець *Потребує редагування*.

### Візуальна клавіатура

Одразу 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. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, the breadcrumb 'WeblateOrg / Django / Hebrew / Translate' is visible. The main area is divided into a 'Translation' section and a sidebar. The 'Translation' section shows the source string 'Files' in English and the target string 'קבצים' in Hebrew. It includes buttons for 'Clone source', 'Save', 'Suggest', and 'Skip'. The sidebar contains a 'Glossary' section with 'English' and 'Hebrew' tabs, and a 'Source information' section with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'Source string age', and 'Translation file'. Below the translation area, there's a table of 'Other languages' with columns for 'Language', 'Status', 'Translation', and 'Edit'.

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

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

## Контекст перекладу

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

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

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

**Сусідні рядки** Displays neighbouring messages from the translation file. These are usually also used in a similar context and prove useful in keeping the translation consistent.

**Інші входження** In case a message appears in multiple places (e.g. multiple components), this tab shows all of them if they are found to be inconsistent (see [Неузгодженість](#)). You can choose which one to use.

**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

Every change is by default (unless turned off in component settings) saved in the database, and can be reverted. Optionally one can still also revert anything in the underlying version control system.

### Довжина перекладеного рядка

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

- The default limitation for translation is ten times longer than source string. This can be turned of by `LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH`. In case you are hitting this, it might be also caused by monolingual translation being configured as bilingual, making Weblate see translation key as source string instead of the actual source string. See *Bilingual and monolingual formats* for more info.
- 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

Each project can have an assigned glossary for any language as a shorthand for storing terminology. Consistency is more easily maintained this way. Terms from the currently translated string can be displayed in the bottom tabs.

#### Керування глосаріями

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

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

[WeblateOrg](#) / [Glossaries](#)

[Components](#)
[Languages](#)
[Info](#)
[Search](#)
[Glossaries](#)
[Insights](#)
[Files](#)
[Tools](#)
[Manage](#)
[Share](#)

WeblateOrg

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

Glossary name

Color

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

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

Additional projects

Search...

Available:

Chosen:

Choose additional projects where this glossary can be used.

Save

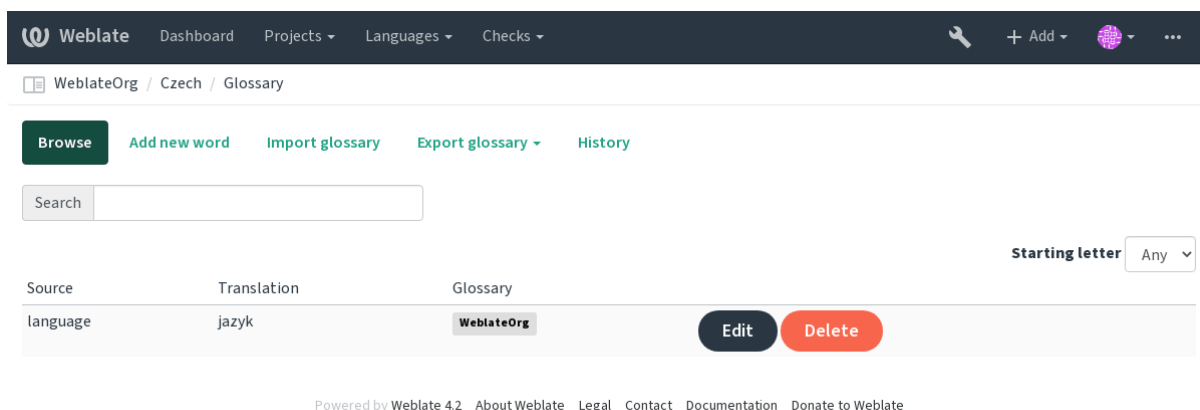
Powered by Weblate 4.2
 [About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

An empty glossary for a given project is automatically created when project is created. Glossaries are shared among all components of the same project and you can also choose to share them with another projects. You can do this only for projects you can administer.

On this list, you can choose which glossary to manage (all languages used in the current project are shown). Following one of the language links will lead you to a page which can be used to edit, import or export the selected glossary, or view the edit history:

1.3. Переклад за допомогою Weblate

17



### 1.3.9 Машинний переклад

Based on configuration and your translated language, Weblate provides you suggestions from several machine translation tools. All machine translations are available in a single tab of each translation page.

#### Дивись також:

Список підтримуваних інструментів наведено у розділі [Машинний переклад](#).

### 1.3.10 Автоматичний переклад

You can use automatic translation to bootstrap translation based on external sources. This tool is called *Automatic translation* accessible in the *Tools* menu, once you have selected a component and a language:

Можна користуватися двома способами дій:

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

Ви також можете вибрати, які рядки буде перекладено автоматично.

**Попередження:** Be mindful that this will overwrite existing translations if employed with wide filters such as *All strings*.

Useful in several situations like consolidating translation between different components (for example website and application) or when bootstrapping translation for a new component using existing translations (translation memory).

**Дивись також:**

*Підтримання синхронізації перекладів у різних складниках*

### 1.3.11 Обмеження частоти

To avoid abuse of the interface, there is rate limiting applied to several operations like searching, sending contact form or translating. In case you are hit by this, you are blocked for a certain period until you can perform the operation again.

The default limits are described in the administrative manual in *Обмеження частоти*, but can be tweaked by configuration.

### 1.3.12 Масове редагування

Bulk edit allows you to perform operation on number of strings. You define search strings and operation to perform and all matching strings are updated. Following operations are supported:

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

---

**Підказка:** This tool is called *Bulk edit* accessible in the *Tools* menu for each project, component or translation.

---

**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

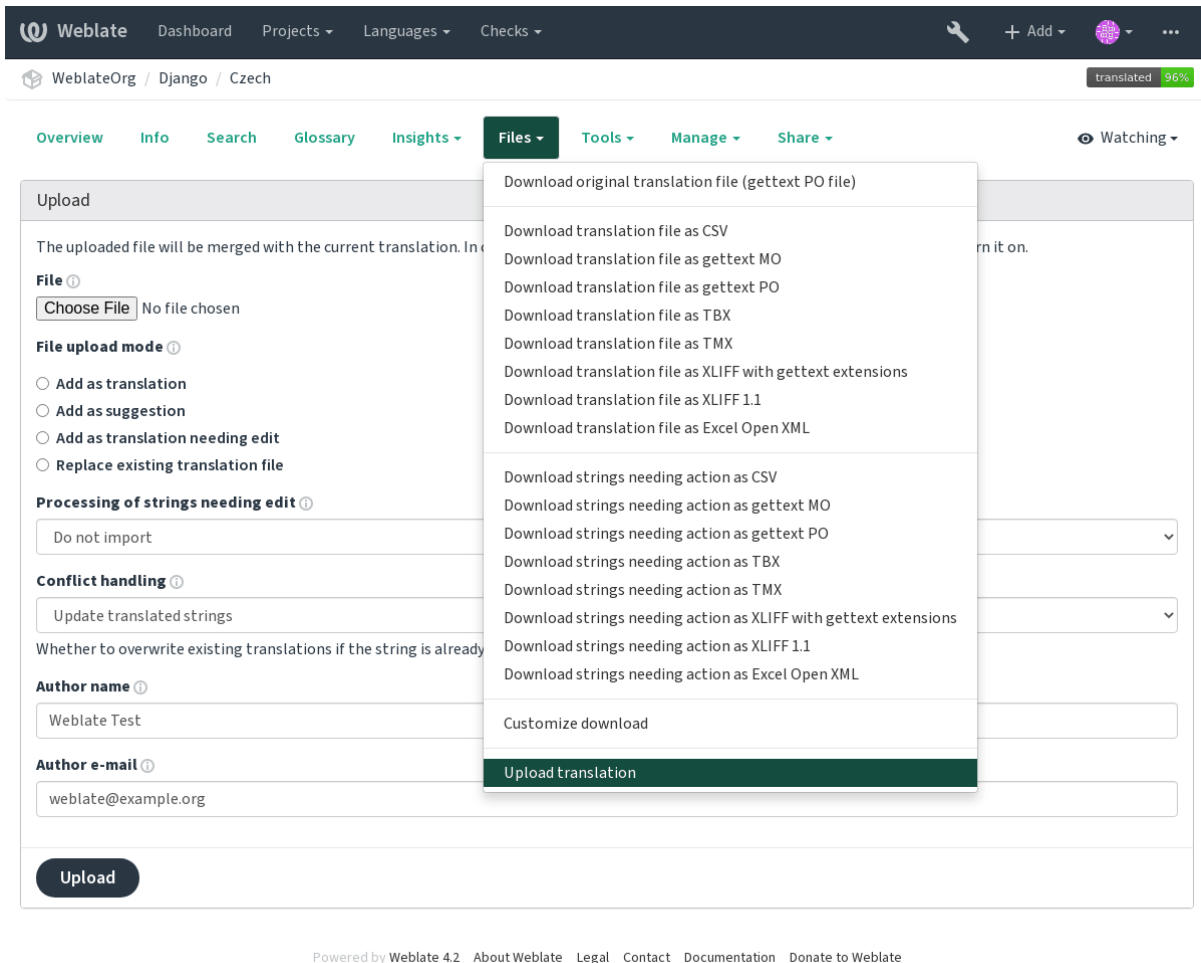
З панелі прикладів проекту або складника, придатні до перекладу файли можна отримати за допомогою пункту *Звантажити початковий файл перекладу* у меню *Файли*. У результаті буде створено копію початкового файла із тим самим вмістом, що і в основній системі керування версіями.

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`)** Updates source strings in bilingual translation file. This is similar to what *Оновити файли po, порівнявши із pot (`msgmerge`)* does.

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

There is also an option for how to handle strings needing edit in the imported file. Such strings can be handle in one of the three following ways: «Do not import», «Import as string needing edit», or «Import as translated».

### Перевизначення авторства

With admin permissions, you can also specify authorship of uploaded file. This can be useful in case you've received the file in another way and want to merge it into existing translations while properly crediting the actual author.

## 1.5 Перевірки і виправлення

The quality checks help catch common translator errors, ensuring the translation is in good shape. The checks can be ignored in case of false positives.

Once submitting a translation with a failing check, this is immediately shown to the user:

W

Weblate

Dashboard

Projects

Languages

Checks

+

Add

WeblateOrg

Django

Czech

Translate

translated

96%

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

<

<

1 / 1

>

>

Custom Search

'%(count)s word'

Position

⌵

Zen

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

Clone source

NBS

Czech, Few

Clone source

NBS

několik slov

Czech, Other

Clone source

NBS

%(count)s slov

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

Needs editing

Save

Suggest

Skip

Nearby strings

20

Comments

Machinery

Other languages

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Things to check

Python format

1

Following format strings are missing:  
%(count)s

Dismiss

Dismiss for all languages

Missing plurals

2

Some plural forms are not translated

Dismiss

Dismiss for all languages

Glossary

English

Czech

No related strings found in the glossary.

Add term to glossary

Source information

Screenshot context

No screenshot currently associated.

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

Source string age

10 seconds ago

Translation file

weblate/locale/cs/LC\_MESSAGES/django.po, string 5

Powered by Weblate 4.2

About Weblate

Legal

Contact

Documentation

Donate to Weblate

1.5. Перевірки і виправлення

23

### 1.5.1 Автоматичні виправлення

In addition to *Перевірки якості*, Weblate can also fix some common errors in translated strings automatically. Use it with caution to not have it add errors.

**Дивись також:**

*AUTOFIX\_LIST*

### 1.5.2 Перевірки якості

Weblate employs a wide range of quality checks on strings. The following section describes them all in further detail. There are also language specific checks. Please file a bug if anything is reported in error.

**Дивись також:**

*CHECK\_LIST*, *Налаштування поведінки*

### 1.5.3 Перевірки перекладів

Executed upon every translation change, helping translators maintain good quality translations.

#### Розмітка BBcode

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

BBCode represents simple markup, like for example highlighting important parts of a message in bold font, or italics.

This check ensures they are also found in translation.

---

**Примітка:** The method for detecting BBcode is currently quite simple so this check might produce false positives.

---

#### Послідовне повторення слів

*Текст містить те саме слово двічі поспіль:*

Нове в версії 4.1.

Checks that no consecutive duplicate words occur in a translation. This usually indicates a mistake in the translation.

---

**Підказка:** This check includes language specific rules to avoid false positives. In case it triggers falsely in your case, let us know. See *Надсилання звітів щодо вад у Weblate*.

---

#### Подвійна прогалина

*Переклад містить подвійну прогалину*

Checks that double space is present in translation to avoid false positives on other space-related checks.

Check is false when double space is found in source meaning double space is intentional.

## Форматовані рядки

Checks that formatting in strings are replicated between both source and translation. Omitting format strings in translation usually causes severe problems, so the formatting in strings should usually match the source.

Weblate supports checking format strings in several languages. The check is not enabled automatically, only if a string is flagged appropriately (e.g. *c-format* for C format). Gettext adds this automatically, but you will probably have to add it manually for other file formats or if your PO files are not generated by **xgettext**.

This can be done per unit (see [Додаткові відомості щодо початкових рядків](#)) or in [Налаштовування складників](#). Having it defined per component is simpler, but can lead to false positives in case the string is not interpreted as a formatting string, but format string syntax happens to be used.

**Підказка:** In case specific format check is not available in Weblate, you can use generic [Заповнювачі](#).

Besides checking, this will also highlight the formatting strings to easily insert them into translated strings:

The screenshot shows the Weblate web interface for a translation unit. The top navigation bar includes 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main header shows 'WeblateOrg / Django / Czech / Translate' and a 'translated 96%' status. The main content area is titled 'Translation' and shows the source string '%(count)s word' in English. Below this, the plural forms are listed: 'Singular' and 'Plural'. The target string in Czech is '%(count)s slovo'. The interface also shows a sidebar with a 'Glossary' section, 'Source information' (including 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'Source string age', and 'Translation file'), and a 'History' tab. The bottom of the interface shows 'Nearby strings' (20), 'Comments', 'Machinery', 'Other languages', and 'History'.

## Рядок інтерполяції AngularJS

*Рядки інтерполяції AngularJS не відповідають джерелу*

Іменованій рядок форматування	Your balance is {{amount}} {{ currency }}
Прапорець для вмикання	<i>angularjs-format</i>

**Дивись також:**

[AngularJS: API: \\$interpolate](#)

## Формат C

*Рядок формату C# не відповідає джерелу*

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

**Дивись також:**

[C format strings](#), [C printf format](#)

## Формат C#

*Рядок формату C# не відповідає джерелу*

Position format string	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
Position format string	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

Position format string	There are {0} apples
Прапорець для вмикання	<i>java-messageformat</i> вмикає перевірку безумовно
	<i>auto-java-messageformat</i> enables check only if there is a format string in the source

**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
Position format string	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
Position format string	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
Іменованій 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
Іменований 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*

Position format string	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
Position format string	Your balance is %1\$f %2\$s
Іменований 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



### Вже перекладено

*Цей рядок перекладено раніше*

Means a string has been translated already. This can happen when the translations have been reverted in VCS or lost otherwise.

### Неузгодженість

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

Weblate checks translations of the same string across all translation within a project to help you keep consistent translations.

The check fails on differing translations of one string within a project. This can also lead to inconsistencies in displayed checks. You can find other translations of this string on the *Other occurrences* tab.

---

**Примітка:** This check also fires in case the string is translated in one component and not in another. It can be used as a quick way to manually handle strings which are not translated in some components just by clicking on the *Use this translation* button displayed on each line in the *Other occurrences* tab.

You can use [Автоматичний переклад](#) addon to automate translating of newly added strings which are already translated in another component.

---

### 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:**

[Markdown span elements](#)

## Максимальна довжина перекладу

*Переклад не перевищує заданої довжини*

Checks that translations are of acceptable length to fit available space. This only checks for the length of translation characters.

Unlike the other checks, the flag should be set as a `key:value` pair like `max-length:100`.

---

**Підказка:** This checks looks at number of chars, what might not be the best metric when using proportional fonts to render the text. The [Найбільший розмір перекладу](#) check does check actual rendering of the text.

The `replacements:` flag might be also useful to expand placeables before checking the string.

---

## Найбільший розмір перекладу

*Перекладений текст не повинен перевищувати заданий розмір*

Нове в версії 3.7.

Translation rendered text should not exceed given size. It renders the text with line wrapping and checks if it fits into given boundaries.

This check needs one or two parameters - maximal width and maximal number of lines. In case the number of lines is not provided, one line text is considered.

You can also configure used font by `font-*` directives (see [Налаштування поведінки](#)), for example following translation flags say that the text rendered with ubuntu font size 22 should fit into two lines and 500 pixels:

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

---

**Підказка:** You might want to set `font-*` directives in [Налаштування складників](#) to have the same font configured for all strings within a component. You can override those values per string in case you need to customize it per string.

The `replacements:` flag might be also useful to expand placeables before checking the string.

---

### Дивись також:

*Керування шрифтами, Налаштовування поведінки, Максимальна довжина перекладу*

### Не збіглися n

*Число n у перекладі не відповідає джерелу*

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

### Невідповідність двокрапки

*Джерело та переклад не закінчуються двокрапкою*

Checks that colons are replicated between both source and translation. The presence of colons is also checked for various languages where they do not belong (Chinese or Japanese).

### Дивись також:

*Двокрапка у Вікіпедії*

### Невідповідність багатокрапок

*Джерело і переклад не завершуються трикрапкою*

Checks that trailing ellipses are replicated between both source and translation. This only checks for real ellipsis (...) not for three dots (. . .).

An ellipsis is usually rendered nicer than three dots in print, and sounds better with text-to-speech.

### Дивись також:

*Багатокрапка у Вікіпедії*

### Невідповідність знаків оклику

*Джерело та переклад не закінчуються знаком оклику*

Checks that exclamations are replicated between both source and translation. The presence of exclamation marks is also checked for various languages where they do not belong (Chinese, Japanese, Korean, Armenian, Limbu, Myanmar or Nko).

### Дивись також:

*Знак оклику у Вікіпедії*

### Невідповідність кінцевої крапки

*Джерело і переклад не завершуються кінцевою крапкою*

Checks that full stops are replicated between both source and translation. The presence of full stops is checked for various languages where they do not belong (Chinese, Japanese, Devanagari or Urdu).

### Дивись також:

*Крапка у Вікіпедії*

## Невідповідність знаків питання

*Джерело та переклад не закінчуються знаком питання*

Checks that question marks are replicated between both source and translation. The presence of question marks is also checked for various languages where they do not belong (Armenian, Arabic, Chinese, Korean, Japanese, Ethiopic, Vai or Coptic).

**Дивись також:**

[Знак питання у Вікіпедії](#)

## Невідповідність крапки з комою

*Джерело і переклад не завершуються крапкою з комою*

Checks that semicolons at the end of sentences are replicated between both source and translation. This can be useful to keep formatting of entries such as desktop files.

**Дивись також:**

[Крапка з комою у Вікіпедії](#)

## Розбіжність розривів рядків

*Кількість знаків нових рядків у перекладі не відповідає джерелу*

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

## Бракує множини

*Деякі форми множини не перекладені*

Checks that all plural forms of a source string have been translated. Specifics on how each plural form is used can be found in the string definition.

Failing to fill in plural forms will in some cases lead to displaying nothing when the plural form is in use.

## Заповнювачі

*У перекладі відсутні деякі замітники:*

Нове в версії 3.9.

Translation is missing some placeholders. These are either extracted from the translation file or defined manually using `placeholders` flag, more can be separated with colon, strings with space can be quoted:

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

**Дивись також:**

[Налаштування поведінки](#)

### Щільність знаків

*Пропущений нерозривний пробіл перед подвійним знаком пунктуації*

Нове в версії 3.9.

Checks that there is non breakable space before double punctuation sign (exclamation mark, question mark, semicolon and colon). This rule is used only in a few selected languages like French or Breton, where space before double punctuation sign is a typographic rule.

**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*

Newlines usually appear in source strings for good reason, omissions or additions can lead to formatting problems when the translated text is put to use.

**Дивись також:**

[Кінцевий знак нового рядка](#)

### Початкові пропуски

*Джерело і переклад не починаються з однакового числа пропусків*

A space in the beginning of a string is usually used for indentation in the interface and thus important to keep.

### Кінцевий знак нового рядка

*Джерело і переклад не завершуються переходом у новий рядок*

Newlines usually appear in source strings for good reason, omissions or additions can lead to formatting problems when the translated text is put to use.

**Дивись також:**

[Перехід у новий рядок](#)

## Кінцеві пропуски

*Джерело і переклад не завершуються пропуском*

Checks that trailing spaces are replicated between both source and translation.

Trailing space is usually utilized to space out neighbouring elements, so removing it might break layout.

## Переклад не змінено

*Джерело і переклад збігаються*

Happens if the source and corresponding translation strings is identical, down to at least one of the plural forms. Some strings commonly found across all languages are ignored, and various markup is stripped. This reduces the number of false positives.

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

The default behavior of this check is to exclude words from the built-in blacklist from the checking. These are words which are frequently not being translated. This is useful to avoid false positives on short strings, which consist only of single word which is same in several languages. This blacklist can be disabled by adding `strict-same` flag to string or component.

**Дивись також:**

*Налаштування складників, Налаштування поведінки*

## Небезпечний HTML-код

*Переклад використовує небезпечну HTML-розмітку*

Нове в версії 3.9.

The translation uses unsafe HTML markup. This check has to be enabled using `safe-html` flag (see *Налаштування поведінки*). There is also accompanied autofixer which can automatically sanitize the markup.

**Дивись також:**

The HTML check is performed by the [Bleach](#) library developed by Mozilla.

## URL

*Цей переклад не містить URL*

Нове в версії 3.5.

The translation does not contain an URL. This is triggered only in case the unit is marked as containing URL. In that case the translation has to be a valid URL.

## Розмітка XML

*Теги XML у перекладі не збігаються з тегами XML у джерелі*

This usually means the resulting output will look different. In most cases this is not a desired result from changing the translation, but occasionally it is.

Checks that XML tags are replicated between both source and translation.

## 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*

Zero-width space (<U+200B>) characters are used to break messages within words (word wrapping).

As they are usually inserted by mistake, this check is triggered once they are present in translation. Some programs might have problems when this character is used.

**Дивись також:**

[Пробіл нульової ширини у Вікіпедії](#)

## 1.5.4 Перевірки джерела

Source checks can help developers improve the quality of source strings.

### Трикrapка

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

This fails when the string uses three dots ( . . . ) when it should use an ellipsis character (...).

Using the Unicode character is in most cases the better approach and looks better rendered, and may sound better with text-to-speech.

**Дивись також:**

[Багатокрапка у Вікіпедії](#)

### Тривалий час не перекладено

*Цей рядок вже тривалий час не перекладено*

Нове в версії 4.1.

When the string has not been translated for a long time, it is can indicate problem in a source string making it hard to translate.

### Кратні невдалі перевірки

*Переклади на декількох мовах мають непройдені перевірки*

Numerous translations of this string have failing quality checks. This is usually an indication that something could be done to improve the source string.

This check failing can quite often be caused by a missing full stop at the end of a sentence, or similar minor issues which translators tend to fix in translation, while it would be better to fix it in the source string.

### 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.*

Consider using named variables instead to allow translators to reorder them.

### Немножинний

*Рядок використовується як множина, але не використовує форми множини*

The string is used as a plural, but does not use plural forms. In case your translation system supports this, you should use the plural aware variant of it.

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.

Advanced queries using boolean operations, parentheses, or field specific lookup can be used to find the strings you want.

When no field is defined, the lookup happens on *Source*, *Target* and *Context* fields.



Search

Custom Search  Sort By

Advanced query builder

Source strings  Search for... ☐ Exact

String has suggestion

String changed after

Query examples

Review strings changed by other users	<code>changed:&gt;=2020-07-18 AND NOT changed_by:testuser</code>	<input type="button" value="Add"/>
Translated strings	<code>state:&gt;=translated</code>	<input type="button" value="Add"/>
Strings with comments	<code>has:comment</code>	<input type="button" value="Add"/>
Strings with any failing checks	<code>has:check</code>	<input type="button" value="Add"/>
Strings with suggestions from others	<code>has:suggestion AND NOT suggestion_author:testuser</code>	<input type="button" value="Add"/>
Approved strings with suggestions	<code>state:approved AND has:suggestion</code>	<input type="button" value="Add"/>
All untranslated strings added the past month	<code>added:&gt;=2020-07-18 AND state:&lt;=needs-editing</code>	<input type="button" value="Add"/>

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

### 1.6.1 Простий пошук

Any phrase typed into the search box is split into words. Strings containing any of them are shown. To look for an exact phrase, put «the searchphrase» into quotes (both single (,) and double («) quotes will work): "this is a quoted string" or 'another quoted string'.

### 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** Project slug, see *Ідентифікатор проекту*.

**changed\_by:ТЕКСТ** Рядок було змінено автором із вказаним ім'ям користувача.

**changed:ДАТА\_І\_ЧАС** Рядок було змінено у вказаний момент часу, передбачено підтримку *Оператори полів*.

**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

You can do an exact match query on different string fields using = operator. For example, to search for all source strings exactly matching hello world, use: source:="hello world". For searching single word expressions, you can skip quotes. For example, to search for all source strings matching hello, you can use: source:=hello.

## 1.6.6 Формальні вирази

Anywhere text is accepted you can also specify a regular expression as `r"regexp"`. For instance, to search for all source strings which contain any digit between 2 and 5, use: `source:r"[2-5]"`

## 1.6.7 Попередньо визначені запити

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

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. A 'Custom Search' dropdown menu is open, showing a list of predefined search filters. The filters are as follows:

- Not translated strings • `state:empty`
- Strings needing action • `state:<translated`
- Translated strings • `state:>=translated`
- Strings marked for edit • `state:needs-editing`
- Strings with suggestions • `has:suggestion`
- Strings with variants • `has:variant`
- Strings with labels • `has:label`
- Strings with context • `has:context`
- Strings needing action without suggestions • `state:<translated AND NOT has:suggestion`
- Strings with comments • `has:comment`
- Strings with any failing checks • `has:check`
- Approved strings • `state:approved`
- Strings waiting for review • `state:translated`

The main interface shows the 'English' source string '%(count)s word' and its Czech translation '%(count)s slovo'. The 'Plural formula' is set to `(n==1)?0:(n>=2&& n<=4)?1:2`. The 'Needs editing' checkbox is unchecked. The 'Save', 'Suggest', and 'Skip' buttons are visible at the bottom of the string editor.

On the right side, there is a sidebar with sections like 'Glossary', 'Source information', 'Labels', 'Flags', 'Source string location', 'Source string age', and 'Translation file'.

At the bottom of the page, there is a 'New comment' section with a text area and a 'Save' button.

## 1.6.8 Упорядкування результатів

Передбачено багато варіантів упорядкування, відповідно до ваших потреб:

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows a translation editor for a string. The source string is 'The string uses three dots (...) instead of an ellipsis character (...)' in English. The target string is empty in Czech. A dropdown menu is open over the 'Position and priority' button, showing options: Position and priority, Position, Priority, Labels, Age of string, Number of words, Number of failing checks, Number of comments, and Key. The sidebar on the right contains a 'Glossary' section with a table for English and Czech terms, and a 'Source information' section with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'Source string age', and 'Translation file'. The bottom section has a 'New comment' form with a text area and a 'Save' button. The footer shows 'Powered by Weblate 4.2' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

## 1.7 Application developer guide

Using Weblate is a process that brings your users closer to you, by bringing you closer to your translators. It up to you to decide how many of its features you want to make use of.

### 1.7.1 Starting with internationalization

Have a project and want to translate it into several languages? This guide will help you do so. Several typical situations are showcased, but most of the examples are generic and can be applied to other scenarios as well.

Before translating any software, you should realize that languages around the world are really different and you should not make any assumption based on your experience. For most of languages it will look weird if you try to concatenate a sentence out of translated segments. You also should properly handle plural forms because many languages have complex rules for that and the internationalization framework you end up using should support this.

Last but not least, sometimes it might be necessary to add some context to the translated string. Imagine a translator would get string `Sun` to translate. Without context most people would translate that as our closest star, but it might

be actually used as an abbreviation for Sunday.

## Вибір оболонки інтернаціоналізації

Choose whatever is standard on your platform, try to avoid reinventing the wheel by creating your own framework to handle localizations. Weblate supports most of the widely used frameworks, see *Підтримувані формати файлів* for more information (especially *Можливості тинів перекладу*).

Our personal recommendation for some platforms is in the following table. This is based on our experience, but that can not cover all use cases, so always consider your environment when doing the choice.

Платформа	Рекомендований формат
Android	<i>Рядкові ресурси Android</i>
iOS	<i>Рядки Apple iOS</i>
Qt	<i>Qt Linguist .ts</i>
Python	<i>GNU Gettext</i>
PHP	<i>GNU Gettext<sup>1</sup></i>
C/C++	<i>GNU Gettext</i>
C#	<i>Файли ресурсів .XML</i>
Perl	<i>GNU Gettext</i>
Ruby	<i>Файли YAML Ruby</i>
Веброзширення	<i>JSON WebExtension</i>
Java	<i>XLIFF<sup>2</sup></i>
JavaScript	<i>Файли JSON i18next<sup>3</sup></i>

Докладнішу процедуру для деяких форматів описано у наступних розділах:

- *Translating software using GNU Gettext*
- *Переклад документації за допомогою Sphinx*
- *Переклад HTML і JavaScript за допомогою CDN Weblate*

## Інтеграція із Weblate

### Отримання оновлень перекладів з Weblate

To fetch updated strings from Weblate you can simply fetch the underlying repository (either from filesystem or it can be made available through *Засіб експортування Git*). Prior to this, you might want to commit any pending changes (see «*Лінії*» *внески*). This can be achieved in the user interface (in the *Repository maintenance*) or from command line using *Клієнт Weblate*.

This can be automated if you grant Weblate push access to your repository and configure *Push URL* in the *Налаштовування складників*.

### Дивись також:

*Безперервна локалізація*

---

<sup>1</sup> The native Gettext support in PHP is buggy and often missing on Windows builds, it is recommended to use third party library *motranslator* instead.

<sup>2</sup> Ви також можете скористатися *Властивості Java*, якщо форми множини не використовуються.

<sup>3</sup> Ви також можете скористатися простим *файли JSON*, якщо форми множини не використовуються.

## Надсилання змін у рядках на Weblate

To push newly updated strings to Weblate, just let it pull from the upstream repository. This can be achieved in the user interface (in the *Repository maintenance*) or from command line using *Клієнт Weblate*.

This can be automated by installing a webhook on your repository to trigger Weblate whenever there is a new commit, see *Оновлення сховищ* for more details.

### Дивись також:

*Безперервна локалізація*

## 1.7.2 Translating software using GNU Gettext

GNU Gettext is one of the most widely used tool for internationalization of free software. It provides a simple yet flexible way to localize the software. It has great support for plurals, it can add further context to the translated string and there are quite a lot of tools built around it. Of course it has great support in Weblate (see *GNU Gettext* file format description).

---

**Примітка:** If you are about to use it in proprietary software, please consult licensing first, it might not be suitable for you.

---

GNU Gettext can be used from a variety of languages (C, Python, PHP, Ruby, JavaScript and many more) and usually the UI frameworks already come with some support for it. The standard usage is through the *gettext()* function call, which is often aliased to *\_()* to make the code simpler and easier to read.

Additionally it provides *pgettext()* call to provide additional context to translators and *ngettext()* which can handle plural types as defined for target language.

As a widely spread tool, it has many wrappers which make its usage really simple, instead of manual invoking of Gettext described below, you might want to try one of them, for example *intltool*.

### Зразок програми

Проста програма мовою C із використанням Gettext може виглядати так:

```
#include <libintl.h>
#include <locale.h>
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int count = 1;
    setlocale(LC_ALL, "");
    bindtextdomain("hello", "/usr/share/locale");
    textdomain("hello");
    printf(
        ngettext(
            "Orangutan has %d banana.\n",
            "Orangutan has %d bananas.\n",
            count
        ),
        count
    );
    printf("%s\n", gettext("Thank you for using Weblate."));
    exit(0);
}
```

### Видобування рядків для перекладу

Щойно у вас буде код, де використовуються виклики `gettext`, ви можете скористатися **xgettext** для видобування повідомлень з коду і зберігання їх до файла `.pot`:

```
$ xgettext main.c -o po/hello.pot
```

---

**Примітка:** Існують альтернативні програми для видобування рядків з коду, наприклад, `pybabel`.

---

Ця програма створює файл шаблону, яким ви можете скористатися для того, щоб розпочати новий переклад (за допомогою програми **msginit**) або оновити наявні переклади після змін у коді (вам варто скористатися для цього програмою **msgmerge**). Отриманий файл є просто структурований текстовий файл:

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2015-10-23 11:02+0200\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"Language: \n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"
"Plural-Forms: nplurals=INTEGER; plural=EXPRESSION;\n"

#: main.c:14
#, c-format
msgid "Orangutan has %d banana.\n"
msgid_plural "Orangutan has %d bananas.\n"
msgstr[0] ""
msgstr[1] ""

#: main.c:20
msgid "Thank you for using Weblate."
msgstr ""
```

Кожен рядок `msgid` визначає рядок для перекладу. Спеціальний порожній рядок на початку файла є заголовком, який містить метадані щодо перекладу.

### Започаткування нового перекладу

Коли у нас буде шаблон, ми можемо розпочати перший переклад:

```
$ msginit -i po/hello.pot -l cs --no-translator -o po/cs.po
Created cs.po.
```

Щойно створений файл `cs.po` вже містить деякі попередньо заповнені дані. Найважливіше те, що у файлі вже визначено належні форми множини для вибраної мови — кількість форм множини буде змінено відповідним чином:

```
# Czech translations for PACKAGE package.
# Copyright (C) 2015 THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# Automatically generated, 2015.
#
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2015-10-23 11:02+0200\n"
"PO-Revision-Date: 2015-10-23 11:02+0200\n"
"Last-Translator: Automatically generated\n"
"Language-Team: none\n"
"Language: cs\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=ASCII\n"
"Content-Transfer-Encoding: 8bit\n"
"Plural-Forms: nplurals=3; plural=(n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2;\n"

#: main.c:14
#, c-format
msgid "Orangutan has %d banana.\n"
msgid_plural "Orangutan has %d bananas.\n"
msgstr[0] ""
msgstr[1] ""
msgstr[2] ""

#: main.c:20
msgid "Thank you for using Weblate."
msgstr ""
```

Цей файл компілюється у оптимізовану двійкову форму — файл `.mo`, який використовується функціями GNU Gettext під час роботи програми.

## Оновлення рядків

Після додавання рядків або зміни рядків у вашій програмі, ви можете запустити `xgettext` знову і повторно створити файл шаблону:

```
$ xgettext main.c -o po/hello.pot
```

Далі, ви можете оновити окремі файли перекладів, щоб вони відповідали новоствореному шаблону (це стоїть і перевпорядкування рядків відповідно до нового шаблону):

```
$ msgmerge --previous --update po/cs.po po/hello.pot
```

## Імпортування до Weblate

Щоб імпортувати такий переклад до Weblate, вам достатньо визначити вказані нижче поля при створенні складника (див. [Налаштування складників](#) для докладного опису полів):

Поле	Значення
Сховище з джерелами	Адреса сховища VCS із вашим проектом
Маска файлів	<code>po/*.po</code>
Заготовка для нових перекладів	<code>po/hello.pot</code>
Формат файлу	Виберіть <i>файл PO Gettext</i>
Нова мова	Виберіть <i>Створити новий мовний файл</i>



Ось і все — можна починати переклад вашого програмного забезпечення!

#### Дивись також:

Ви можете знайти приклад Gettext із багатьма мовами у проєкті Weblate Hello на GitHub: <<https://github.com/WeblateOrg/hello>>.

## 1.7.3 Переклад документації за допомогою Sphinx

**Sphinx** — інструмент для створення чудової документації. У ньому використовується простий синтаксис reStructuredText, він може видавати результат у багатьох форматах. Якщо ви шукаєте приклад, цю документацію також побудовано з його використанням. Дуже корисним помічником у використанні Sphinx є служба [Read the Docs](#), яка надає змогу збирати і оприлюднювати вашу документацію безкоштовно.

I will not focus on writing documentation itself, if you need guidance with that, just follow instructions on the [Sphinx](#) website. Once you have documentation ready, translating it is quite easy as Sphinx comes with support for this and it is quite nicely covered in their [Internationalization](#). It's matter of few configuration directives and invoking of the `sphinx-intl` tool.

If you are using Read the Docs service, you can start building translated documentation on the Read the Docs. Their [Localization of Documentation](#) covers pretty much everything you need - creating another project, set its language and link it from main project as a translation.

Now all you need is translating the documentation content. Sphinx generates PO file for each directory or top level file, what can lead to quite a lot of files to translate (depending on `gettext_compact` settings). You can import the `index.po` into Weblate as an initial component and then configure [Виявлення складників](#) addon to automatically discover all others.

Table 1: Налаштовування складників

Назва складника	Documentation
Маска файлів	docs/locales/*/LC_MESSAGES/index.po
Заготовка для нових перекладів	docs/locales/index.pot
Формат файлу	файл PO gettext
Прапорці перекладу	rst-text

Table 2: Налаштовування виявлення складників

Регулярний вираз для зіставлення файлів перекладу	docs/locales/(?P<language>[^\./]*)/LC_MESSAGES/(?P<component>[^\./]*)\.po
Налаштувати назву складника	Documentation: {{ component title }}
Визначити основний файл для нового перекладу	docs/locales/{{ component }}.pot

**Підказка:** Would you prefer Sphinx to generate just single PO file? There is a hacky way to achieve this (used by Weblate documentation) by overriding Sphinx way to get a Gettext domain of a document. Place following snippet to your Sphinx configuration in `conf.py`:

```
import sphinx.transforms.i18n
import sphinx.util.i18n

# Hacky way to have all localized content in single domain
sphinx.transforms.i18n.docname_to_domain = (
    sphinx.util.i18n.docname_to_domain
) = lambda docname, compact: "docs"
```

This might be directly supported by Sphinx in future releases, see <<https://github.com/sphinx-doc/sphinx/issues/784>>.

**Дивись також:**

The [Odorik](#) python module documentation is built using Sphinx, Read the Docs and translated using Weblate.

## 1.7.4 Переклад HTML і JavaScript за допомогою CDN Weblate

Starting with Weblate 4.2 it is possible to export localization to a CDN using *CDN локалізації JavaScript* addon.

---

**Примітка:** Цю можливість налаштовано на Hosted Weblate. Вона потребує додаткового налаштування у вашому екземплярі, див. [LOCALIZE\\_CDN\\_URL](#) і [LOCALIZE\\_CDN\\_PATH](#).

---

Upon installation into your component it will push committed translations (see «[Лінійні](#)» *внески*) to the CDN and these can be used in your web pages to localize them.

### Створення складника

First, you need to create a monolingual component which will hold your strings, see [Додавання проєктів і складників перекладу](#) for generic instructions on that.

In case you have existing repository to start with (for example the one containing HTML files), create an empty JSON file in the repository for the source language (see [Джерельна мова](#)), for example `locales/en.json`. The content should be `{ }` to indicate an empty object. Once you have that, the repository can be imported into Weblate and you can start with an addon configuration.

---

**Підказка:** In case you have existing translations, you can place them into the language JSON files and those will be used in Weblate.

---

For those who do not want to use existing repository (or do not have one), choose *Start from scratch* when creating component and choose *JSON file* as a file format (it is okay to choose any monolingual format at this point).

### Налаштовування додатка CDN Weblate

The *CDN локалізації JavaScript* addon provides few configuration options.

**Порогове значення перекладу** Translations translated above this threshold will be included in the CDN.

**Вибір CSS** Configures which strings from the HTML documents are translatable, see [Видобування рядків для CDN Weblate](#) and [Локалізація HTML за допомогою CDN Weblate](#).

**Назва куки мови** Name of cookie which contains user selected language. Used in the JavaScript snippet for [Локалізація HTML за допомогою CDN Weblate](#).

**Видобути рядки з файлів HTML** List of files in the repository or URLs where Weblate will look for translatable strings and offer them for a translation, see [Видобування рядків для CDN Weblate](#).

### Видобування рядків для CDN Weblate

Рядки перекладу мають бути на Weblate. Ви можете досягти цього або вручну за допомогою створення записів з програмного інтерфейсу, або вказавши адреси за допомогою поля *Видобути рядки з файлів HTML* (Weblate виконає видобування рядків автоматично). Файли мають зберігатися у сховищі коду або містити віддалені адреси, дані з яких може бути отримано і оброблено Weblate на регулярній основі.

За типових налаштувань *Селектор CSS* видобуває елементи із класом `CSS 110n`. Наприклад, буде видобуто два рядки з таких фрагментів коду:

```
<section class="content">
  <div class="row">
    <div class="wrap">
      <h1 class="section-title min-m l10n">Maintenance in progress</h1>
      <div class="page-desc">
        <p class="l10n">We're sorry, but this site is currently down for
↩maintenance.</p>
      </div>
    </div>
  </div>
</section>
```

Якщо ви не хочете модифікувати наявний код, ви можете також скористатися \* як селектором для обробки таких елементів.

---

**Примітка:** У поточній версії видобуватиметься лише текст елементів. У цьому додатку не передбачено підтримки локалізації атрибутів елементів або елементів із дочірніми записами.

---

## Локалізація HTML за допомогою CDN Weblate

Щоб локалізувати документ HTML, вам слід завантажити скрипт `weblate.js`:

```
<script src="https://weblate-cdn.com/a5ba5dc29f39498aa734528a54b50d0a/weblate.js"
↩async></script>
```

Після завантаження він автоматично знайде усі відповідні придатні до перекладу елементи (на основі налаштувань *селектора CSS*) і замінить їх текст на переклад.

Мова користувача визначається за налаштованою кукою і резервним варіантом — бажаною мовою користувача, яку налаштовано у браузері.

The *Language cookie name* can be useful for integration with other applications (for example choose `django_language` when using Django).

## Локалізація JavaScript

Доступ до окремих перекладів у CDN надається у форматі двомовних файлів JSON. Отримати файл можна за допомогою такого коду:

```
fetch("https://weblate-cdn.com/a5ba5dc29f39498aa734528a54b50d0a/cs.json")
  .then(response => response.json())
  .then(data => console.log(data));
```

У цьому випадку слід реалізувати справжню логіку локалізації.

## 1.7.5 Translation component alerts

Shows errors in the Weblate configuration or the translation project for any given translation component. Guidance on how to address found issues is also offered.

Currently the following is covered:

- Duplicated source strings in translation files
- Duplicated languages within translations
- Merge or update failures in the source repository
- Unused new base in component settings

- Parse errors in the translation files
- Duplicate filemask used for linked components
- Broken URLs

Alerts are listed on each respective component page as *Alerts*. If it is missing, the component clears all current checks. Alerts can not be ignored, but will disappear once the underlying problem has been fixed.

A component with both duplicated strings and languages looks like this:

The screenshot shows the Weblate interface with the 'Alerts' tab selected. The top navigation bar includes 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main header shows 'WeblateOrg / Duplicates' and a 'translated 37%' badge. The 'Alerts' tab is active, showing three alert cards:

- Duplicated translation.** The component contains several translation files mapped to a single language in Weblate. Please fix this by removing one of the translation files. Please consider the following:
  - Avoid having translation files for both the plain language code and its equivalent territory designation (for example de and de\_DE).
 The following occurrences were found:
 

Language	Language codes
Czech	cs_CZ, cs

 Appeared a second ago, last seen a second ago
- Duplicated string found in the file.** The component contains several duplicated translation strings. The following occurrences were found:
 

Language	Source
Italian	Thank you for using Weblate.

 Please fix this by removing duplicated strings with same identifier from the translation files.
 Appeared a second ago, last seen a second ago
- License info missing.** Any publicly available project should have defined license to indicate what terms apply to contributions.
 Appeared a second ago, last seen a second ago

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

## Дивись також:

Використання нетипових уповноважень за сертифікатом

## 1.7.6 Building translators community

### Список перевірок локалізації спільнотою

Нове в версії 3.9.

The *Community localization checklist* which can be found in the *Menu* menu of each component can give you guidance to make your localization process easy for community translators.

Community localization checklist

Here you can find guidance to make your localization project attractive to the community.

**Version control integration**

- Configure repository hooks for automated flow of updates to Weblate. [Configure](#)
- Configure push URL for automated flow of translations from Weblate. [Configure](#)

**Building community**

- Define translation instructions to give translators a guideline. [Configure](#)
- Make your translations available under a libre license. [Configure](#)
- Fix this component to clear its alerts. [Configure](#)

**Provide context to the translators**

- Add screenshots to show where strings are being used. [Configure](#)
- Use flags to indicate special strings in your translation. [Configure](#)

**Workflow customization**

- Enable addon: Update LINGUAS file  
Updates the LINGUAS file when a new translation is added. [Configure](#)
- Enable addon: Update ALL\_LINGUAS variable in the "configure" file  
Updates the ALL\_LINGUAS variable in "configure", "configure.in" or "configure.ac" files, when a new translation is added. [Configure](#)

[Return to the component](#)

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

## 1.7.7 Керування перекладами

### Додавання нових перекладів

New strings can be made available for translation when they appear in the base file, called *Template for new translations* (see [Налаштування складників](#)). If your file format doesn't require such a file, as is the case with most monolingual translation flows, you can start with blank files).

New languages can be added right away when requested by a user in Weblate, or a notification will be sent to project admins for approval and manual addition. This can be done using *Start new translation* in [Налаштування складників](#).

---

**Примітка:** Адміністратори проекту завжди можуть розпочинати переклад на Weblate безпосередньо.

---

Language files added manually to the VCS are added to the component when Weblate updates the repository. About repository update settings, see [Оновлення сховищ](#)).

## Варіанти рядків

Variants are useful to group several strings together so that translators can see all variants of the string at one place. You can define regular expression to group the strings in the *Налаштування складників*:

**Webplate** Dashboard Projects Languages Checks

WebplateOrg / Android / Settings

Basic **Translation** Version control Commit messages Files

**Suggestions**

☒ **Turn on suggestions**  
Whether to allow translation suggestions at all.

☐ **Suggestion voting**  
Whether users can vote for suggestions.

**Autoaccept suggestions** ⓘ

0  
Automatically accept suggestions with this number of votes, use 0 to turn it off.

**Translation settings**

☒ **Allow translation propagation**  
Whether translation updates in other components will cause automatic translation in this one

**Translation flags** ⓘ

Additional comma-separated flags to influence quality checks. Possible values can be found in the documentation.

**Variants regular expression** ⓘ

\_(short|min)\$

Regular expression used to determine variants of a string.

**Enforced checks** ⓘ

Search...

Available:	Chosen:
AngularJS interpolation string	
BBcode markup	
C format	
C# format	
Consecutive duplicated words	

List of checks which can not be ignored.

**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**

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to Weblate

The expression is matched against *Key* to generate root key of the variant. All matching strings are then part of single variants group, including the translation exactly matching the root key, even if that is not matched by the regular expression.

Нижче наведено декілька прикладів використання:

Варіант використання	Варіант формального виразу	Відповідні ключі перекладу
Ідентифікація за суфіксом	(Short   Min) \$	monthShort, monthMin, month
Вбудована ідентифікація	# [SML]	dial#S.key, dial#M.key, dial.key

The variant is later grouped when translating:

**Weblate** Dashboard Projects Languages Checks

WeblateOrg / Android / English / Translate translated 100%

< > 1 / 3 Custom Search Monday Position and priority Zen

### Source string

**Key**

dow\_monday

**English** Clone source NBS ... " ' - \_

Monday

☐ Needs editing 6/100

Save Suggest Skip

Nearby strings 13 Nearby keys 13 Variants 3 Comments Other languages History

Key	English	State
dow_monday	Monday	✓
dow_monday_min	M	✓
dow_monday_short	Mon	✓

### Things to check

**Variants** There are 3 variants for this string.

View

### Glossary

**English** **English**

No related strings found in the glossary.

+ Add term to glossary

### Source information

**Screenshot context** No screenshot currently associated.

**Explanation** No explanation currently provided.

**Key** dow\_monday

**Labels** No labels currently set.

**Flags** No flags currently set.

**Source string age** 6 seconds ago

**Translation file** app/src/main/res/values/strings.xml, string 11

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to Weblate

## Мітки рядків

Split component translation strings into categories by text and colour in the project configuration.

Label name

Color

Current sprint

Next sprint

Add label

Label name

Color

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

Save

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to Weblate

**Підказка:** Labels can be assigned to units in *Додаткові відомості щодо початкових рядків* by bulk editing, or using the *Масове редагування* addon.

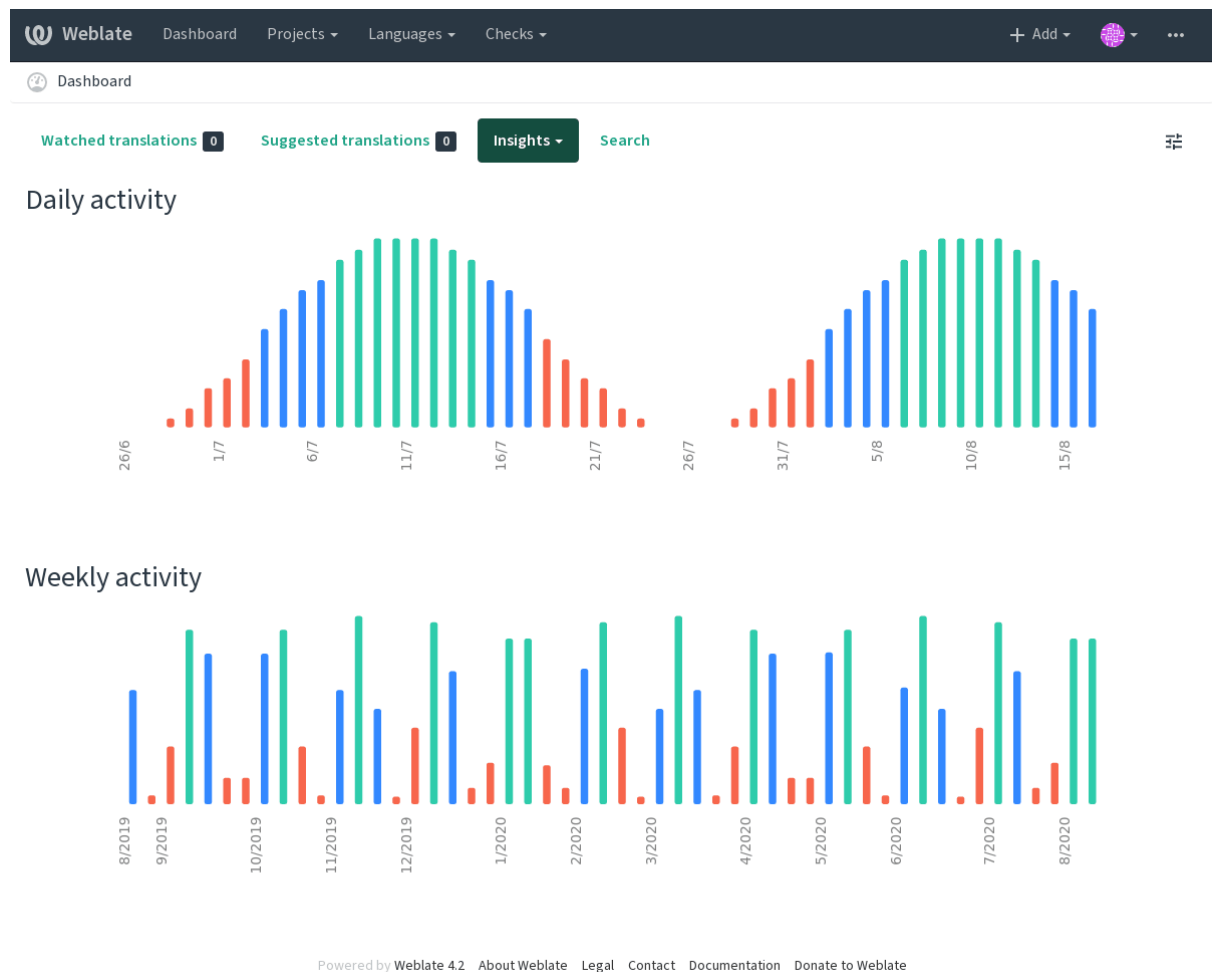
## 1.7.8 Reviewing strings

### Activity reports

Activity reports check changes of translations, for projects, components or individual users.

The activity reports for a project or component is accessible from its dashboard, on the *Insights* tab, selecting *Activity*.





More reports are accessible on the *Insights* tab, selecting *Translation reports*.

The activity of the currently signed in user can be seen by clicking on *Profile* from the user menu on the top right.

## Source strings checks

There are many *Перевірки якості*, some of them focus on improving the quality of source strings. Many failing checks suggest a hint to make source strings easier to translate. All types of failing source checks are displayed on the *Source* tab of every component.

## Translation string checks

Erroneous failing translation string checks indicate the problem is with the source string. Translators sometimes fix mistakes in the translation instead of reporting it - a typical example is a missing full stop at the end of a sentence.

Reviewing all failing checks can provide valuable feedback to improve its source strings. To make source strings review easier, Weblate automatically creates a translation for the source language and shows you source level checks there:

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Android / English
 translated 100%

Overview Info Search Glossary Insights Files Tools Manage Share
 Not watching

### Source strings

13 Strings 100%

46 Words 100%

This translation is being used as source strings within this component.

Translate

### Strings status

13 All strings — 46 words

13 Translated strings — 46 words

13 Strings without a label — 46 words

### Other components

Component	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Language names	✓					
Django	✓			1		

[Browse all components](#)

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to Weblate

One of the most interesting checks here is the *Кратні невдалі перевірки* - it is triggered whenever there is failure on multiple translations of a given string. Usually this is something to look for, as this is a string which translators have problems translating properly.

The detailed listing is a per language overview:

W Weblate

Dashboard

Projects

Languages

Checks

🔧

+

Add

🌐

🌐

...

🌐 WeblateOrg

/ Android

/ English

/ Translate

translated 100%

⏪

⏴

1 / 3

⏵

⏩

Custom Search

Monday

Position and priority

≡

⚡ Zen

⚙

Source string

Key

dow\_monday

English

Monday

Needs editing

Save

Suggest

Skip

Nearby strings 13

Nearby keys 13

Variants 3

Comments

Other languages

History

Key	English	State
auth_activity_title	Authenticate	✓
auth_hint_password	Password	✓
auth_hint_pin	PIN	✓
auth_msg_authenticate	Please authenticate to start andOTP!	✓
auth_msg_confirm_encryption	Please confirm your authentication to generate the new encryption key!	✓
auth_button_unlock	Unlock	✓
auth_toast_password_missing	Please set a password in the <b>Settings</b>!	✓
auth_toast_pin_missing	Please set a PIN in the <b>Settings</b>!	✓
auth_toast_password_again	Wrong password, please try again!	✓
auth_toast_pin_again	Wrong PIN, please try again!	✓
dow_monday	Monday	✓
dow_monday_short	Mon	✓
dow_monday_min	M	✓

Things to check

Variants

There are 3 variants for this string.

View

Glossary

English English

No related strings found in the glossary.

Add term to glossary

Source information

Screenshot context

No screenshot currently associated.

Explanation

No explanation currently provided.

Key

dow\_monday

Labels

No labels currently set.

Flags

No flags currently set.

Source string age

6 seconds ago

Translation file

app/src/main/res/values/strings.xml, string 11

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to Weblate

String comments

Translators can comment on both translation and source strings. Each *Налаштування складників* can be configured to receive such comments to an e-mail address, and using the developers mailing list is usually the best approach. This way you can keep an eye on when problems arise in translation, take care of them, and fix them quickly.

56

Розділ 1. Документація для користувачів

## 1.7.9 Promoting the translation

Weblate provides you widgets to share on your website or other sources to promote the translation project. It also has a nice welcome page for new contributors to give them basic information about the translation. Additionally you can share information about translation using Facebook or Twitter. All these possibilities can be found on the *Share* tab:

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Widgets

### Promoting translation projects

You can point newcomers to the introduction page at <http://localhost:53981/engage/weblateorg/>.

### Promoting specific translations

Besides promoting the whole translation project, you can also choose a specific language or component to promote: All languages

All components

### Image widgets

You can use the following widgets to promote translation of your project. They can increase the visibility of your translation projects and bring in new contributors.

translated 85%

Status badge

Vertical language bar chart

Horizontal language bar chart

35 STRINGS 13 LANGUAGES 85% TRANSLATED

Big status badge

85% TRANSLATED

Small status badge

Panel

Color variants:

translated 85%

HTML code

```
<a href="http://localhost:53981/engage/weblateorg/?utm_source=widget">

```

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to Weblate

All these badges are provided with the link to simple page which explains users how to translate using Weblate:



# Get involved in **WeblateOrg**

**Hello and thank you for your interest** — **WeblateOrg** is being translated using **Weblate**, a web tool designed to ease translating for both developers and translators.

35	13	85.2%
STRINGS	LANGUAGES	TRANSLATED

The translation project for WeblateOrg currently contains **35 string** for translation. It is being translated into **13 languages**. Overall, these translations are **85.2% complete**. If you would like to contribute to translation of WeblateOrg, you need to register on this server. This translation is open only to a limited group of translators, if you want to contribute please get in touch with the project maintainers.

Translate

View project languages

Powered by Weblate 4.2   About Weblate   Legal   Contact   Documentation   Donate to Weblate

## 1.7.10 Translation progress reporting

Reporting features give insight into how a translation progresses over a given period. A summary of contributions to any given component over time is provided. The reporting tool is found in the *Insights* menu of any translation component, project or on the dashboard:

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. The main navigation bar includes 'Translations', 'Info', 'Alerts', 'Search', 'Glossaries', 'Insights', 'Files', 'Tools', 'Manage', and 'Share'. The 'Insights' dropdown menu is open, showing options: 'History', 'Activity', 'Download statistics (CSV)', 'Download statistics (JSON)', and 'Translation reports'. The 'Credits' panel on the left lists translators and includes a 'Generate' button. The 'Stats' panel on the right shows a 'reStructuredText' report format, a 'Report period' of 'As specified', and 'Starting date' and 'Ending date' fields, with a 'Generate' button at the bottom.

Several reporting tools are available on this page and all can produce output in HTML, reStructuredText or JSON. The first two formats are suitable for embedding statistics into existing documentation, while JSON is useful for further processing of the data.

## Перекладачі

Generates a document usable for crediting translators - sorted by language and lists all contributors to a given language:

```
* Czech

  * Michal Čihař <michal@cihar.com> (10)
  * John Doe <john@example.com> (5)

* Dutch

  * Jane Doe <jane@example.com> (42)
```

It will render as:

- чеська
  - Michal Čihař <michal@cihar.com> (10)
  - John Doe <john@example.com> (5)
- нідерландська
  - Jae Doe <jane@example.com> (42)

**Підказка:** The number in parenthesis indicates number of contributions in given period.

## Статистика учасника

Generates the number of translated words and strings by translator name:

=====													
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
=====													
Name		Email											
↪	Count total	Source words total		Source chars total									
↪	Target words total	Target chars total		Count new									
↪	Source words new	Source chars new		Target words new									
↪	Target chars new	Count approved		Source words approved									
↪	Source chars approved	Target words approved		Target chars approved		Count							
↪	edited	Source words edited		Source chars edited		Target							
↪	words edited	Target chars edited											
=====													
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
=====													
Michal Čihar		michal@cihar.com											
↪		1		3		24			↪				
↪		3		21		1			↪				
↪	3		24		3				↪				
↪	21		0		0			0	↪				
↪		0		0		0		0	↪				
↪		0		0		0		0	↪				
↪	0								↪				
Allan Nordhøy		allan@example.com											
↪		2		5		25			↪				
↪		4		28		2			↪				
↪	3		24		3				↪				
↪	21		0		0			0	↪				
↪		0		0		0		0	↪				
↪		0		0		0		0	↪				
↪	0								↪				
=====													
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				
↪	=====	=====	=====	=====	=====	=====	=====	=====	↪				

And it will get rendered as:

Назва	Ел. пошта	Загальна кількість	Загальна кількість	Загальна кількість	Target words total	Target words total	Source words new	Source words new	Кількість нових	Кількість нових	Target words new	Кількість нових	Кількість нових	Source words new	Target words new	Target words new	Source words new	Source words new	Source words new	Target words new	Target words new
		кількість	кількість	кількість					вих сим-во-лів у по-ча-тко-вих ряд-ків	вих сим-во-лів у по-ча-тко-вих ряд-ків		ле-них	твер-джен-них слів по-ча-тко-вих ряд-ків								
Michal Čihař	halichal@cihar.cz	3	24	3	21	1	3	24	3	21	0	0	0	0	0	0	0	0	0	0	0
Allan Nordhøy	allan@examp1e.no	4	25	4	28	2	3	24	3	21	0	0	0	0	0	0	0	0	0	0	0

It can be useful if you pay your translators based on amount of work, it gives you various stats on translators work.

Уся статистика доступна у трьох варіантах:

**Total** Загальна кількість редагованих рядків.

**New** Нових перекладених рядків, які не було перекладено раніше.

**Approved** Кількість затверджень рядків у процесі рецензування (див. [Спеціалізовані рецензенти](#)).

**Edited** Кількість редагованих рядків, які було перекладено раніше.

Наступні дані доступні для кожного з таких пунктів:

**Count** Кількість рядків.

**Edits** Кількість редагувань у рядку, виміряна за відстанню Дамерау-Левенштейна.

**Source words** Number of words in the source string.

**Source characters** Number of characters in the source string.

**Target words** Number of words in the translated string.

**Target characters** Number of characters in the translated string.

## 1.8 Процеси перекладу

Передбачено підтримку декількох робочих процесів перекладу.

The following is not a complete list of ways to configure Weblate. You can base other workflows on the most usual examples listed here.



### 1.8.1 Доступ до перекладу

The *Керування доступом* is not much discussed in the workflows as each access control option can be applied to any workflow. Please consult that documentation for information on how to manage access to translations.

In the following chapters, *any user* means a user who has access to the translation. It can be any authenticated user if the project is public, or a user that has a *Translate* permission for the project.

### 1.8.2 Стани перекладу

Кожен перекладений рядок може перебувати в одному з таких станів:

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

**Потребує редагування** Переклад потребує редагування, зазвичай, у результаті внесення змін до початкового рядка. Переклад зберігається у файлі. Залежно від формату файла, його може бути позначено як такий, що потребує редагування (наприклад, встановлено прапорець неточного перекладу).

**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.

### 1.8.3 Безпосередній переклад

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

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

Параметр	Значення	Примітка
Увімкнути рецензії	вимк.	Налаштовано на рівні проекту.
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	<i>Users</i>	Або <i>Перекласти</i> із <i>Керування доступом</i> .
Група рецензентів	Н/Д	Не використовується.

### 1.8.4 Експертне рецензування

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

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

Параметр	Значення	Примітка
Увімкнути рецензії	вимк.	Налаштовано на рівні 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.

### 1.8.5 Спеціалізовані рецензенти

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

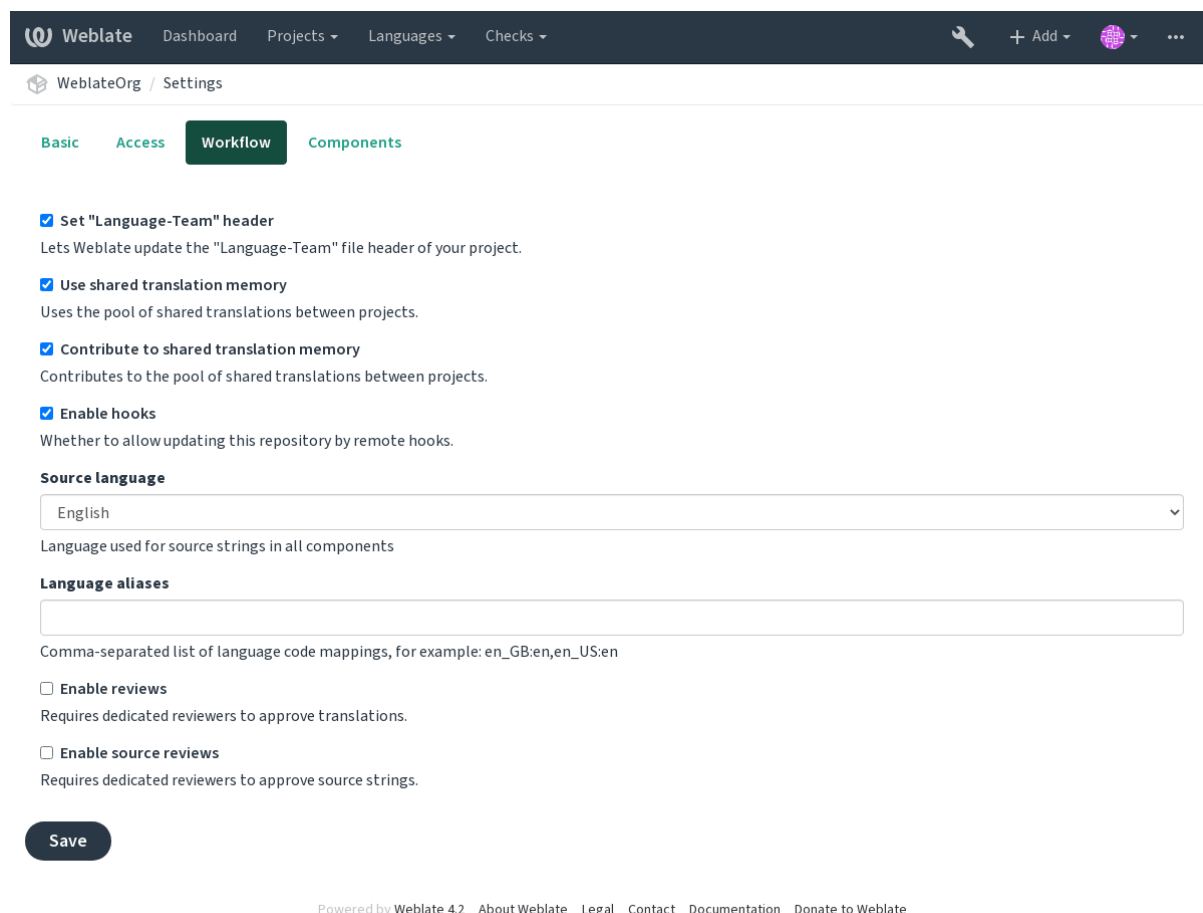
Якщо у проєкті є спеціалізовані рецензенти, ви матимете дві групи користувачів: одна зможе надсилати переклади, а інше рецензувати їх з метою забезпечення коректності та якості.

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

Параметр	Значення	Примітка
Увімкнути рецензії	Увімк.	Налаштовано на рівні 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> .

## 1.8.6 Вмикання рецензування

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



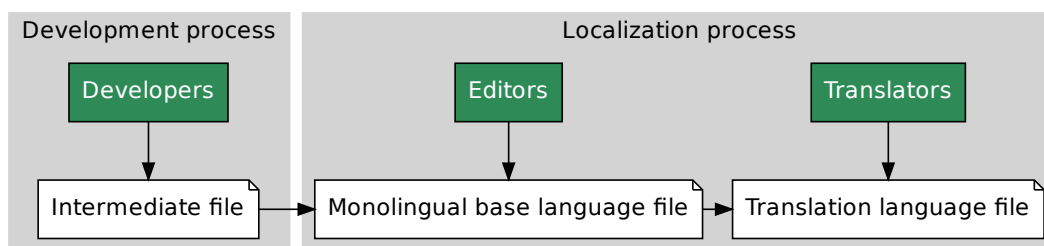
The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo, 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a breadcrumb trail: 'WeblateOrg / Settings'. The main content area has four tabs: 'Basic', 'Access', 'Workflow' (which is active and highlighted in dark green), and 'Components'. Under the 'Workflow' tab, there are four checked checkboxes with descriptions: 'Set "Language-Team" header', 'Use shared translation memory', 'Contribute to shared translation memory', and 'Enable hooks'. Below these is a 'Source language' dropdown menu set to 'English'. Further down is a 'Language aliases' text input field. At the bottom of the settings section are two unchecked checkboxes: 'Enable reviews' and 'Enable source reviews'. A dark green 'Save' button is located at the bottom left of the settings area. At the very bottom of the page, a footer line contains the text 'Powered by Weblate 4.2' followed by links for 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

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

## 1.8.7 Шлюз якості для початкових рядків

In many cases the original source language strings are coming from developers, because they write the code and provide initial strings. However developers are often not a native speakers in the source language and do not provide desired quality of the source strings. The intermediate translation can help you in addressing this - there is additional quality gateway for the strings between developers and translators and users.

By setting *Проміжний мовний файл*, this file will be used as source for the strings, but it will be edited to source language to polish it. Once the string is ready in the source language, it will be also available for translators to translate into additional languages.



**Дивись також:**

*Проміжний мовний файл, Основний файл для одномовного перекладу, Bilingual and monolingual formats*

## 1.8.8 Рецензування початкових рядків

With *Увімкнути рецензування початкових даних* enabled, the review process can be applied on the source strings. Once enabled, users can report issues in the source strings. The actual process depends on whether you use bilingual or monolingual formats.

For monolingual formats, the source string review behaves similarly as with *Спеціалізовані рецензенти* - once issue is reported on the source string, it is marked as *Needs editing*.

The bilingual formats do not allow direct editing of the source strings (these are typically extracted directly from the source code). In this case *Source needs review* label is attached to strings reported by translators. You should review such strings and either edit them in the source or remove the label.

**Дивись також:**

*Bilingual and monolingual formats, Спеціалізовані рецензенти, Мітки рядків*

## 1.9 Поширені питання та відповіді на них

### 1.9.1 Налаштування

#### Як створити автоматизовану процедуру?

Weblate can handle all the translation things semi-automatically for you. If you give it push access to your repository, the translations can happen without interaction, unless some merge conflict occurs.

1. Set up your Git repository to tell Weblate when there is any change, see *Обробники сповіщень* for info on how to do it.
2. Set a push URL at your *Налаштовування складників* in Weblate, this allows Weblate to push changes to your repository.
3. Turn on push-on-commit on your *Налаштування проєкту* in Weblate, this will make Weblate push changes to your repository whenever they happen at 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?

Merge conflicts happen from time to time when the translation file is changed in both Weblate and the upstream repository concurrently. You can usually avoid this by merging Weblate translations prior to making changes in the translation files (e.g. before running msgmerge). Just tell Weblate to commit all pending translations (you can do it in *Repository maintenance* in the *Manage* menu) and merge the repository (if automatic push is not on).

If you've already ran into a merge conflict, the easiest way is to solve all conflicts locally at your workstation - is to simply add Weblate as a remote repository, merge it into upstream and fix any conflicts. Once you push changes back, Weblate will be able to use the merged version without any other special actions.

---

**Примітка:** Depending on your setup, access to the Weblate repository might require authentication. When using the built in *Засіб експортування Git* in Weblate, you authenticate with your username and the API key.

---

```
# 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
```

If you're using multiple branches in Weblate, you can do the same to all of them:

```
# 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
```

(continues on next page)

(continued from previous page)

```
# Merge master branch:
git checkout master
git merge weblates-second/master
... # Resolve conflicts
git commit

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

In case of gettext PO files, there is a way to merge conflicts in a semi-automatic way:

Fetch and keep a local clone of the Weblate Git repository. Also get a second fresh local clone of the upstream Git repository (i. e. you need two copies of the upstream Git repository: An intact and a working copy):

```
# 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?

Weblate supports pushing translation changes within one *Налаштування проєкту*. For every *Налаштування складників* which has it turned on (the default behavior), the change made is automatically propagated to others. This way translations are kept synchronized even if the branches themselves have already diverged quite a lot, and it is not possible to simply merge translation changes between them.

Once you merge changes from Weblate, you might have to merge these branches (depending on your development workflow) discarding differences:

```
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?

You can use `git submodule` for separating translations from source code while still having them under version control.

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. Link Weblate to this repository, it no longer needs access to the repository containing your source code.
4. You can update the main repository with translations from Weblate by:

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

Please consult the `git submodule` documentation for more details.

## How can I check whether my Weblate is set up properly?

Weblate includes a set of configuration checks which you can see in the admin interface, just follow the *Performance report* link in the admin interface, or open the `/manage/performance/` URL directly.

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

This is the default committer name, configured when you create a translation component. You can change it in the administration at any time.

The author of every commit (if the underlying VCS supports it) is still recorded correctly as the user that made the translation.

**Дивись також:**

*Налаштування складників*

## 1.9.2 Користування

### How do I review the translations of others?

- You can subscribe to any changes made in *Сповіщення* and then check others contributions as they come in by e-mail.
- There is a review tool available at the bottom of the translation view, where you can choose to browse translations made by others since a given date.

### How do I provide feedback on a source string?

On context tabs below translation, you can use the *Source* tab to provide feedback on a source string, or discuss it with other translators.

### How can I use existing translations while translating?

- Use the import functionality to load compendium as translations, suggestions or translations needing review. This is the best approach for a one-time translation using a compendium or a similar translation database.
- You can set up *tmserver* with all databases you have and let Weblate use it. This is good when you want to use it several times during translation.
- Another option is to translate all related projects in a single Weblate instance, which will make it automatically pick up translations from other projects as well.

**Дивись також:**

*Машинний переклад, Машинний переклад*

### Does Weblate update translation files besides translations?

Weblate tries to limit changes in translation files to a minimum. For some file formats it might unfortunately lead to reformatting the file. If you want to keep the file formatted your way, please use a pre-commit hook for that.

For monolingual files (see *Підтримувані формати файлів*) Weblate might add new translation strings not present in the *template*, and not in actual translations. It does not however perform any automatic cleanup of stale strings as that might have unexpected outcomes. If you want to do this, please install a pre-commit hook which will handle the cleanup according to your requirements.

Weblate also will not try to update bilingual files in any way, so if you need `po` files being updated from `pot`, you need to do it yourself.



**Дивись також:**

*Обробка сховища зі скриптами*

### Звідки беруться визначення мов і як можна додати власне визначення?

The basic set of language definitions is included within Weblate and Translate-toolkit. This covers more than 150 languages and includes info about plural forms or text direction.

You are free to define your own languages in the administrative interface, you just need to provide info about it.

### Чи може Weblate підсвічувати зміни у неточно перекладеному рядку?

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

For Gettext PO files, you have to pass the parameter `--previous` to **msgmerge** when updating PO files, for example:

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

For monolingual translations, Weblate can find the previous string by ID, so it shows the differences automatically.

### Why does Weblate still show old translation strings when I've updated the template?

Weblate does not try to manipulate the translation files in any way other than allowing translators to translate. So it also does not update the translatable files when the template or source code have been changed. You simply have to do this manually and push changes to the repository, Weblate will then pick up the changes automatically.

---

**Примітка:** It is usually a good idea to merge changes done in Weblate before updating translation files, as otherwise you will usually end up with some conflicts to merge.

---

For example with gettext PO files, you can update the translation files using the **msgmerge** tool:

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

In case you want to do the update automatically, you can install addon *Оновити файли po, порівнявши із pot (msgmerge)*.

## 1.9.3 Вирішення проблем

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

This happens sometimes when your Git repository grows too much and you have many of them. Compressing the Git repositories will improve this situation.

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](#)

### When accessing the site I get a «Bad Request (400)» error

This is most likely caused by an improperly configured `ALLOWED_HOSTS`. It needs to contain all hostnames you want to access on your Weblate. For example:

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

Дивись також:

[Налаштування дозволених вузлів](#)

### What does mean «There are more files for the single language (en)»?

This typically happens when you have translation file for source language. Weblate keeps track of source strings and reserves source language for this. The additional file for same language is not processed.

- In case the translation to the source language is desired, please change the [Джерельна мова](#) in the project settings.
- 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.

## 1.9.4 Можливості

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

Weblate currently does not have native support for anything other than [Git](#) (with extended support for [GitHub](#), [Gerrit](#) and [Subversion](#)) and `ref:vcs-mercurial`, but it is possible to write backends for other VCSes.

Крім того, ви можете скористатися [Git remote helpers](#) у Git для доступу до інших систем керування версіями.

Weblate also supports VCS less operation, see [Локальні файли](#).

---

**Примітка:** For native support of other VCSes, Weblate requires using distributed VCS, and could probably be adjusted to work with anything other than Git and Mercurial, but somebody has to implement this support.

---

Дивись також:

[Налаштування інтеграції із керуванням версіями](#)

### Як Weblate зберігає авторство перекладів?

Every change made in Weblate is committed into VCS under the translators name. This way every single change has proper authorship, and you can track it down using the standard VCS tools you use for code.

Additionally, when the translation file format supports it, the file headers are updated to include the translator's name.

Дивись також:

[list\\_translators](#), [Translation progress reporting](#)

### Why does Weblate force showing all PO files in a single tree?

Weblate was designed in a way that every PO file is represented as a single component. This is beneficial for translators, so they know what they are actually translating. If you feel your project should be translated as one, consider merging these po files. It will make life easier even for translators not using Weblate.

---

**Примітка:** In case there is great demand for this feature, it might be implemented in future versions.

---

### Why does Weblate use language codes such `sr_Latn` or `zh_Hant`?

These are language codes defined by [RFC 4646](#) to better indicate that they are really different languages instead previously wrongly used modifiers (for `@latin` variants) or country codes (for Chinese).

Weblate still understands legacy language codes and will map them to current one - for example `sr@latin` will be handled as `sr_Latn` or `zh@CN` as `sr_Hans`.

## 1.10 Підтримувані формати файлів

Weblate supports most translation format understood by `translate-toolkit`, however each format being slightly different, some issues with formats that are not well tested can arise.

**Дивись також:**

[Translation Related File Formats](#)

---

**Примітка:** When choosing a file format for your application, it's better to stick some well established format in the toolkit/platform you use. This way your translators can additionally use whatever tools they are used to, and will more likely contribute to your project.

---

### 1.10.1 Bilingual and monolingual formats

Both monolingual and bilingual formats are supported. Bilingual formats store two languages in single file—source and translation (typical examples are *GNU Gettext*, *XLIFF* or *Рядки Apple iOS*). On the other side, monolingual formats identify the string by ID, and each language file contains only the mapping of those to any given language (typically *Рядкові ресурси Android*). Some file formats are used in both variants, see the detailed description below.

For correct use of monolingual files, Weblate requires access to a file containing complete list of strings to translate with their source—this file is called *Основний файл для одномовного перекладу* within Weblate, though the naming might vary in your paradigm.

Additionally this workflow can be extended by utilizing *Проміжний мовний файл* to include strings provided by developers, but not to be used as is in the final strings.

### 1.10.2 Автоматичне виявлення

Weblate can automatically detect several widespread file formats, but this detection can harm your performance and will limit features specific to given file format (for example automatic addition of new translations).

### 1.10.3 Можливості типів перекладу

Можливості усіх підтримуваних форматів:

Формат	Linguality <sup>1</sup>	Форми множини <sup>2</sup>	Коментарі <sup>3</sup>	Контекст <sup>4</sup>	Розташування <sup>5</sup>	Відношення <sup>8</sup>	Додаткові стани <sup>6</sup>
<i>GNU Gettext</i>	двомовний	так	так	так	так	так <sup>9</sup>	потребує редагування
<i>Одномовний gettext</i>	моно	так	так	так	так	так <sup>9</sup>	потребує редагування
<i>XLIFF</i>	обидва	так	так	так	так	так <sup>10</sup>	потребує редагування, затверджено
<i>Властивості Java</i>	обидва	ні	так	ні	ні	ні	
<i>Властивості GWT</i>	моно	так	так	ні	ні	ні	
<i>Переклади Joomla</i>	моно	ні	так	ні	так	ні	
<i>Qt Linguist .ts</i>	обидва	так	так	ні	так	так <sup>10</sup>	потребує редагування
<i>Рядкові ресурси Android</i>	моно	так	так <sup>7</sup>	ні	ні	так <sup>10</sup>	
<i>Рядки Apple iOS</i>	двомовний	ні	так	ні	ні	ні	
<i>Рядки PHP</i>	моно	ні <sup>11</sup>	так	ні	ні	ні	
<i>файли JSON</i>	моно	ні	ні	ні	ні	ні	
<i>Файли JSON i18next</i>	моно	так	ні	ні	ні	ні	
<i>Файли JSON go-i18n</i>	моно	так	ні	ні	ні	ні	
<i>Файл ARB</i>	моно	так	так	ні	ні	ні	
<i>JSON WebExtension</i>	моно	так	так	ні	ні	ні	
<i>Файли ресурсів .XML</i>	моно	ні	так	ні	ні	так <sup>10</sup>	

continues on next page

Table 3 – continued from previous page

Формат	Linguality <sup>1</sup>	Форми множини <sup>2</sup>	Коментарі <sup>3</sup>	Контекст <sup>4</sup>	Розташування <sup>5</sup>	Прапорці <sup>8</sup>	Додаткові стани <sup>6</sup>
<i>Файли CSV</i>	МОНО	ні	так	так	так	ні	потребує редагування
<i>Файли YAML</i>	МОНО	ні	так	ні	ні	ні	
<i>Файли YAML Ruby</i>	МОНО	так	так	ні	ні	ні	
<i>Файли DTD</i>	МОНО	ні	ні	ні	ні	ні	
<i>Flat XML</i>	МОНО	ні	ні	ні	ні	так <sup>10</sup>	
<i>Файли RC Windows</i>	МОНО	ні	так	ні	ні	ні	
<i>Excel Open XML</i>	МОНО	ні	так	так	так	ні	потребує редагування
<i>Файли мета-даних крамниці застосунків</i>	МОНО	ні	ні	ні	ні	ні	
<i>Файли субтитрів</i>	МОНО	ні	ні	ні	так	ні	
<i>Файли HTML</i>	МОНО	ні	ні	ні	ні	ні	
<i>Формат OpenDocument</i>	МОНО	ні	ні	ні	ні	ні	
<i>Формат IDML</i>	МОНО	ні	ні	ні	ні	ні	
<i>Переклади INI</i>	МОНО	ні	ні	ні	ні	ні	
<i>Переклади INI Inno Setup</i>	МОНО	ні	ні	ні	ні	ні	

<sup>1</sup> Див. *Bilingual and monolingual formats*<sup>2</sup> Форми множини потрібні для належної локалізації рядків із змінними-лічильниками.<sup>3</sup> Коментарями можна скористатися для передавання додаткових відомостей щодо рядка, який слід перекласти.<sup>4</sup> Context is used to differentiate identical strings used in different scopes (for example *Sun* can be used as an abbreviated name of the day «Sunday» or as the name of our closest star).<sup>5</sup> Location of a string in source code might help proficient translators figure out how the string is used.<sup>8</sup> Див. *Налаштування поведінки*<sup>6</sup> Additional states supported by the file format in addition to «Not translated» and «Translated».<sup>9</sup> Коментарі типу gettext використовуються як прапорці.<sup>10</sup> 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*.<sup>7</sup> XML comment placed before the `<string>` element, parsed as a developer comment.<sup>11</sup> Підтримку форм множини передбачено лише для Laravel, який використовує синтаксис рядків для визначення цих форм, див. *Локалізація у Laravel*.

### 1.10.4 GNU Gettext

Найпоширеніший формат для перекладу вільного програмного забезпечення. Цей формат був першим, підтримку якого було реалізовано у Weblate. Його підтримка є і досі найповнішою.

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	Порожньо
Заготовка для нових перекладів	po/messages.pot
Формат файлу	файл PO Gettext

#### Дивись також:

*Translating software using GNU Gettext*, *Переклад документації за допомогою Sphinx*, *Gettext у Вікіпедії*, *PO Files*, *Оновити змінну ALL\_LINGUAS у файлі «configure»*, *Налаштувати виведення gettext*, *Оновити файл LINGUAS*, *Генерувати то-файли*, *Оновити файли po*, *порівнявши із pot (msgmerge)*

### Одномовний gettext

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

Типовий одномовний файл PO gettext виглядає ось так:

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

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

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

А базовий мовний файл ось такий:

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

(continues on next page)

(continued from previous page)

```
#: 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 файла	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>

### 1.10.5 XLIFF

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

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

#### Дивись також:

Специфікація *XML Localization Interchange File Format (XLIFF)*

### Стани перекладу

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

The `state` attribute in the file is partially processed and mapped to the «Needs edit» state in Weblate (the following states are used to flag the string as needing edit if there is a target present: `new`, `needs-translation`, `needs-adaptation`, `needs-l10n`). Should the `state` attribute be missing, a string is considered translated as soon as a `<target>` element exists.

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. *Налаштовування 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 на `Source_Sans_Pro`. Будь ласка, зверніть увагу на це при іменуванні групи шрифтів (див. *Керування шрифтами*).

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

### 1.10.6 Властивості Java

Власний формат Java для перекладів.

Властивості Java, зазвичай, використовуються як одномовні переклади.

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



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

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 Java (ISO-8859-1)</i>

**Дивись також:**

[Властивості Java у Вікіпедії](#), [Mozilla and Java properties files](#), [Форматує файл властивостей Java](#), [Очищення файлів перекладу](#)

### 1.10.7 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](#)

### 1.10.8 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](#)

### 1.10.9 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 замінників `%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](#)

### 1.10.10 Переклади Joomla

Нове в версії 2.12.

Власний формат Joomla для перекладів.

Переклади Joomla, зазвичай, використовуються як одномовні переклади.

Типовий <i>Налаштовування складників Weblate</i>	
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](#)

### 1.10.11 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 <i>Налаштовування складників Weblate</i> 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 <i>Налаштування складників Weblate</i> 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, *Bilingual and monolingual formats*

### 1.10.12 Рядкові ресурси Android

Специфічний для Android формат файлів для перекладу програм.

Рядкові ресурси Android є одномовними. Файл *Одномовний базовий мовний файл* зберігається у іншому місці, якщо порівнювати із іншими `res/values/strings.xml`.

Типовий <i>Налаштування складників Weblate</i>	
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>

---

### 1.10.13 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>Налаштовування складників</i> Weblate	
Маска файла	Resources/*.lproj/Localizable.strings
Основний файл для одномовного перекладу	Resources/en.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

### 1.10.14 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</i> Weblate	
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 [
    'apples' => 'There is one apple|There are many apples',
];
```

**Дивись також:**

PHP, Локалізація у Laravel

### 1.10.15 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.

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.
- 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": ""
  }
}
```

**Попередження:** Поточна версія Weblate обробляє JSON із вкладеністю шляхом спрощення ключів. Це призводить до вад із перетворенням у послідовну форму, якщо у ключах використано спеціальні символи, зокрема . або [ ], оскільки Weblate вважає такі символи позначками вкладеності.

Див. <<https://github.com/WeblateOrg/weblate/issues/2149>>

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

Дивись також:

*JSON, Налаштувати виведення JSON, Очищення файлів перекладу,*

### 1.10.16 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 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 i18next JSON</i>

**Дивись також:**

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

### 1.10.17 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 2. Підтримки простих файлів JSON версії 1 не реалізовано.

---

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](#),

### 1.10.18 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</i> Weblate	
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*

### 1.10.19 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:** While this format is called JSON, its specification allows to include comments, which are not part of JSON specification. Weblate currently does not support file with comments.

---

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>Налаштування складників</i> Weblate	
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

### 1.10.20 Файли ресурсів .XML

Нове в версії 2.3.

A .XML resource (.resx) file employs a monolingual XML file format used in Microsoft .NET applications. It is interchangeable with .resw, when using identical syntax to .resx.

Типовий <i>Налаштування складників</i> Weblate	
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,

### 1.10.21 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:

- Files with header defining fields (source, translation, location, ...). This is the recommended approach, as it is the least error prone.
- Files with two fields—source and translation (in this order), choose *Simple CSV file* as file format
- Файли із полями, які визначено у translate-toolkit: location, source, target, ID, fuzzy, context, translator\_comments, developer\_comments

**Попередження:** The CSV format currently automatically detects the dialect of the CSV file. In some cases the automatic detection might fail and you will get mixed results. This is especially true for CSV files with newlines in the values. As a workaround it is recommended to omit quoting characters.

Приклад файла:

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

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



Дивись також:

[CSV](#)

### 1.10.22 Файли YAML

Нове в версії 2.9.

The plain YAML files with string keys and values. Weblate also extract strings from lists or dictionaries.

Приклад файла YAML:

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

Типовий <i>Налаштовування складників</i> Weblate	
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](#)

### 1.10.23 Файли YAML Ruby

Нове в версії 2.9.

Ruby i18n YAML files with language as root node.

Приклад файла перекладу YAML Ruby:

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

Типовий <i>Налаштовування складників</i> Weblate	
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](#)

## 1.10.24 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](#)

## 1.10.25 Прості файли XML

Нове в версії 3.9.

Приклад простого файла 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 Weblate</i>	
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](#)

## 1.10.26 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 файла RC Windows:

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT

STRINGTABLE
BEGIN
```

(continues on next page)

(continued from previous page)

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

**Дивись також:**[Windows RC files](#)

### 1.10.27 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 Weblate</i>	
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“

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

**Дивись також:**

---

Subtitles

### 1.10.29 Excel Open XML

Нове в версії 3.2.

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

When uploading XLSX files for translation, be aware that only the active worksheet is considered, and there must be at least a column called `source` (which contains the source string) and a column called `target` (which contains the translation). Additionally there should be the column called `context` (which contains the context path of the translation string). If you use the XLSX download for exporting the translations into an Excel workbook, you already get a file with the correct file format.

### 1.10.30 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](#)

### 1.10.31 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](#)

### 1.10.32 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.

### 1.10.33 Інше

Most formats supported by translate-toolkit which support serializing can be easily supported, but they did not (yet) receive any testing. In most cases some thin layer is needed in Weblate to hide differences in behavior of different translate-toolkit storages.

**Дивись також:**

Translation Related File Formats

### 1.10.34 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.

Some formats expect to start with an empty file and only translated strings to be included (for example *Рядкові пелюстки Android*), while others expect to have all keys present (for example *GNU Gettext*). In some situations this really doesn't depend on the format, but rather on the framework you use to handle the translation (for example with *файли JSON*).

When you specify *Заготовка для нових перекладів* in *Налаштування складників*, Weblate will use this file to start new translations. Any exiting translations will be removed from the file when doing so.

When *Template for new translations* is empty and the file format supports it, an empty file is created where new strings will be added once they are translated.

The *Language code style* allows you to customize language code used in generated filenames:

**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.

---

**Примітка:** Weblate recognizes any of these when parsing translation files, the above settings only influences how new files are created.

---

### 1.10.35 Рядки лише для читання

Нове в версії 3.10.

Read-only strings from translation files will be included, but can not be edited in Weblate. This feature is natively supported by few formats (*XLIFF* and *Рядкові ресурси Android*), but can be emulated in others by adding a read-only flag, see *Налаштування поведінки*.

## 1.11 Налаштування інтеграції із керуванням версіями

Weblate currently supports *Git* (with extended support for *GitHub*, *Gerrit* and *Subversion*) and *Mercurial* as version control backends.

### 1.11.1 Доступ до сховищ

The VCS repository you want to use has to be accessible to Weblate. With a publicly available repository you just need to enter the correct URL (for example `https://github.com/WeblateOrg/weblate.git`), but for private repositories or for push URLs the setup is more complex and requires authentication.

#### Доступ до сховищ з Hosted Weblate

For Hosted Weblate there is a dedicated push user registered on GitHub, Bitbucket, Codeberg and GitLab (with username *weblate* named *Weblate push user*). You need to add this user as a collaborator and give it appropriate permission to your repository (read only is okay for cloning, write is required for pushing). Depending on service and your organization settings, this happens immediately or requires confirmation from Weblate side.

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

Once the *weblate* user is added, you can configure *Сховище з джерелами* and *Адреса для записування до сховища* using SSH protocol (for example `git@github.com:WeblateOrg/weblate.git`).

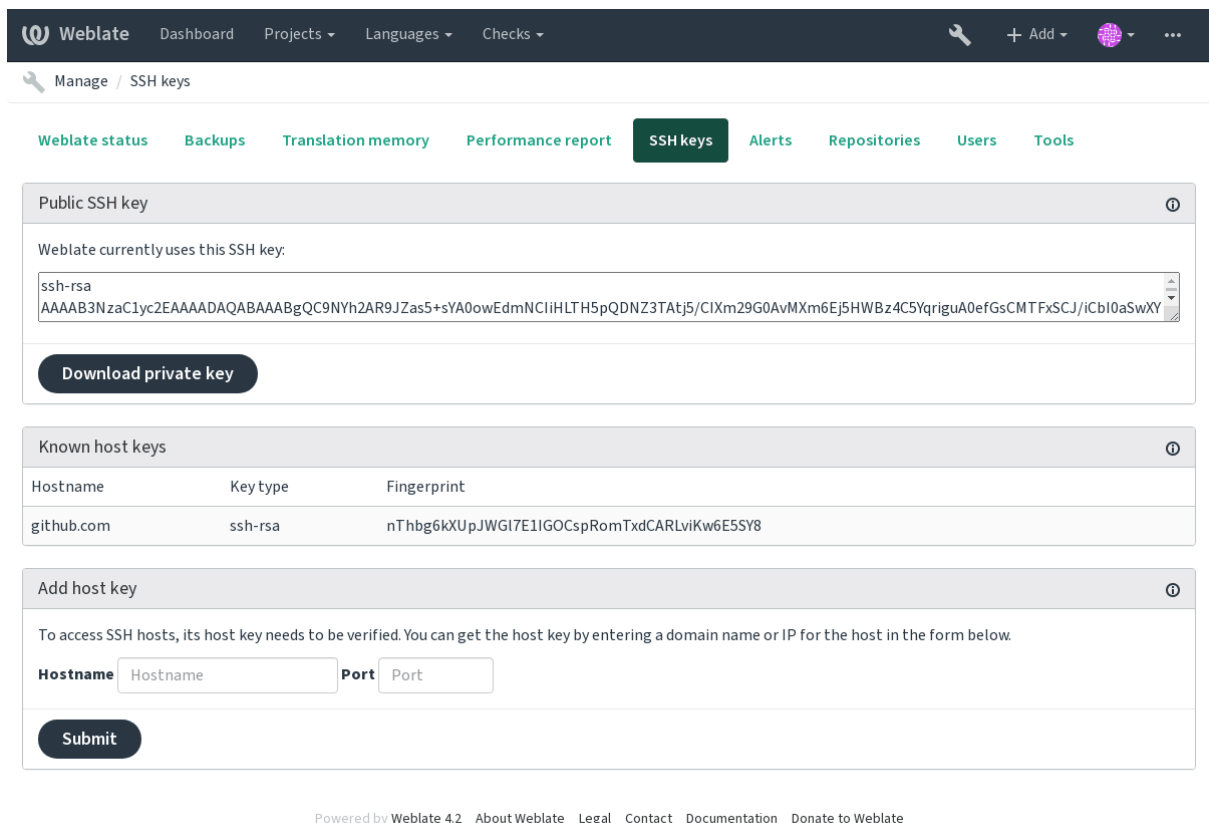
#### Сховища із доступом за SSH

The most frequently used method to access private repositories is based on SSH. Authorize the public Weblate SSH key (see *Ключ SSH Weblate*) to access the upstream repository this way.

**Попередження:** On GitHub, each key can be added to only one repository, see *Сховища GitHub* and *Доступ до сховищ з Hosted Weblate*.

Weblate also stores the host key fingerprint upon first connection, and fails to connect to the host should it be changed later (see *Перевірка ключів SSH вузла*).

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



Public SSH key

Weblate currently uses this SSH key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGC9NYh2AR9JZas5+sYA0owEdmNCIiLTH5pQDNZ3TAtj5/CIXm29G0AvMXm6Ej5HWBz4C5YqriguA0efGsCMTFxSCJ/iCbI0aSwXY
```

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

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

Hostname  Port

Submit

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

## Ключ SSH Weblate

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

Admins can generate or display the public key currently used by Weblate in the (from *SSH keys*) on the admin interface landing page.

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

**Підказка:** Make a backup of the generated private Weblate SSH key.

## Перевірка ключів SSH вузла

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

In case you want to verify them before connecting to the repository, verify the SSH host keys of the servers you are going to access in *Add host key*, from the same section of the admin interface. Enter the hostname you are going to access (e.g. `gitlab.com`), and press *Submit*. Verify its fingerprint matches the server you added. They are shown in the confirmation message:

Dashboard
Projects ▾
Languages ▾
Checks ▾
⚙️
+ Add ▾

...

Manage / SSH keys

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

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

Public SSH key

Weblate currently uses this SSH key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGC9NYh2AR9JZas5+sYA0owEdmNCIiHLTH5pQDNZ3TAtj5/CIXm29G0AvMXm6Ej5HWBz4C5YqriguA0efGsCMTFxSCJ/iCbl0aSwXY
```

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

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

Hostname

github.com

Port

Port

Submit

Powered by Weblate 4.2
 [About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

## Сховища GitHub

Access via SSH is possible (see *Сховища із доступом за SSH*), but in case you need to access more than one repository, you will hit a GitHub limitation on allowed SSH key usage (since one key can be used only for one repository).

In case the *Гілка для запису* is not set, the project is forked and changes pushed through a fork. In case it is set, changes are pushed to the upstream repository and chosen branch.

For smaller deployments, use HTTPS authentication with a personal access token and your GitHub account, see *Creating an access token for command-line use*.

For bigger setups, it is usually better to create a dedicated user for Weblate, assign it the public SSH key generated in Weblate (see *Ключ SSH Weblate*) and grant it access to all the repositories you want to translate. This approach is also used for Hosted Weblate, there is dedicated *weblate* user for that.

### Дивись також:

*Доступ до сховищ з Hosted Weblate*



### Внутрішні адреси Weblate

To share one repository between different components you can use a special URL like `weblate://project/component`. This way, the component will share the VCS repository configuration with the referenced component (`project/component` in the example).

Weblate automatically adjusts repository URL when creating component when it finds component with matching repository setup. You can override this in last step of component configuration.

Reasons to use this:

- Заощаджує місце на диску сервера — сховище зберігається лише один раз.
- Пришвидшує оновлення — оновлюється лише одне сховище.
- There is just single exported repository with Weblate translations (see [Засіб експортування Git](#)).
- Some addons can operate on more components sharing single repository, for example [Сполучити Git подання](#).

### Сховища HTTPS

To access protected HTTPS repositories, include the username and password in the URL. Don't worry, Weblate will strip this info when the URL is shown to users (if even allowed to see the repository URL at all).

For example the GitHub URL with authentication added might look like: `https://user:your_access_token@github.com/WeblateOrg/weblate.git`.

---

**Примітка:** If you username or password contains special characters, those have to be URL encoded, for example `https://user%40example.com:%24password%23@bitbucket.org/....`

---

### Використання проксі-сервера

If you need to access HTTP/HTTPS VCS repositories using a proxy server, configure the VCS to use it.

This can be done using the `http_proxy`, `https_proxy`, and `all_proxy` environment variables, (as described in the [cURL documentation](#)) or by enforcing it in the VCS configuration, for example:

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

---

**Примітка:** The proxy configuration needs to be done under user running Weblate (see also [Права доступу у файловій системі](#)) and with `HOME=$DATA_DIR/home` (see [DATA\\_DIR](#)), otherwise Git executed by Weblate will not use it.

---

**Дивись також:**

[Сторінка підручника cURL](#), [Документація з налаштувань Git](#)

### 1.11.2 Git

#### Дивись також:

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

#### Git з примусовим «push»

This behaves exactly like Git itself, the only difference being that it always force pushes. This is intended only in the case of using a separate repository for translations.

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

#### Налаштовування Git

Weblate invokes all VCS commands with `HOME=$DATA_DIR/home` (see *DATA\_DIR*), therefore editing the user configuration needs to be done in `DATA_DIR/home/.git`.

#### Git remote helpers

You can also use Git *remote helpers* for additionally supporting other version control systems, but be prepared to debug problems this may lead to.

At this time, helpers for Bazaar and Mercurial are available within separate repositories on GitHub: *git-remote-hg* and *git-remote-bzr*. Download them manually and put somewhere in your search path (for example `~/bin`). Make sure you have the corresponding version control systems installed.

Once you have these installed, such remotes can be used to specify a repository in Weblate.

Клонування проєкту `gnuhello` з Launchpad за допомогою Bazaar:

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

For the `hello` repository from selenic.com using Mercurial:

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

**Попередження:** The inconvenience of using Git remote helpers is for example with Mercurial, the remote helper sometimes creates a new tip when pushing changes back.

### 1.11.3 GitHub

Нове в версії 2.3.

This adds a thin layer atop *Git* using the *hub* tool to allow pushing translation changes as pull requests, instead of pushing directly to the repository.

*Git* pushes changes directly to a repository, while *GitHub* creates pull requests. The latter is not needed for merely accessing Git repositories.

#### Дивись також:

*Записування змін з Weblate*

### Записування змін до GitHub у формі запитів щодо об'єднання

If not wanting to push translations to a GitHub repository, they can be sent as either one or many pull requests instead.

#### Дивись також:

`GITHUB_USERNAME`, *Налаштовування концентратора* — настанови з налаштування

### Налаштовування концентратора

*Записування змін до GitHub у формі запитів щодо об'єднання* requires a configured `hub` installation on your server. Follow the installation instructions at <https://hub.github.com/> use `hub` to finish the configuration, for example:

```
# Use DATA_DIR as configured in Weblate settings.py, it is /app/data in the Docker
HOME=${DATA_DIR}/home hub clone octocat/Spoon-Knife
```

The `hub` will ask you for your GitHub credentials, retrieve a token and store it in `~/.config/hub`. This file has to be readable by the user running Weblate.

---

**Примітка:** Use the username you configured `hub` with, as `GITHUB_USERNAME` (`WEBLATE_GITHUB_USERNAME` for the Docker image).

---

### 1.11.4 GitLab

Нове в версії 3.9.

This just adds a thin layer atop *Git* using the `lab` tool to allow pushing translation changes as merge requests instead of pushing directly to the repository.

There is no need to use this access Git repositories, ordinary *Git* works the same, the only difference is how pushing to a repository is handled. With *Git* changes are pushed directly to the repository, while *GitLab* creates merge request.

#### Дивись також:

*Записування змін з Weblate*

### Записування змін до GitLab у формі запитів щодо злиття

If not wanting to push translations to a GitLab repository, they can be sent as either one or many merge requests instead.

Configure the `lab` command line tool and set `GITLAB_USERNAME` for this to work.

#### Дивись також:

`GITLAB_USERNAME`, *Налаштовування Lab* щодо настанов з налаштування

## Налаштовування Lab

Записування змін до *GitLab* у формі запитів щодо злиття requires a configured *lab* installation on your server. Follow the installation instructions at *lab* and run it without any arguments to finish the configuration, for example:

```
# Use DATA_DIR as configured in Weblate settings.py, it is /app/data in the Docker
$ HOME=${DATA_DIR}/home lab
Enter GitLab host (default: https://gitlab.com):
Create a token here: https://gitlab.com/profile/personal_access_tokens
Enter default GitLab token (scope: api):
(Config is saved to ~/.config/lab.hcl)
```

The *lab* will ask you for your GitLab access token, retrieve it and store it in `~/.config/lab.hcl`. The file has to be readable by the user running Weblate.

---

**Примітка:** Use the username you configured *lab* with, as `GITLAB_USERNAME` (`WEBLATE_GITLAB_USERNAME` for the Docker image).

---

### 1.11.5 Gerrit

Нове в версії 2.2.

Adds a thin layer atop *Git* using the *git-review* tool to allow pushing translation changes as Gerrit review requests, instead of pushing a directory to the repository.

The Gerrit documentation has the details on the configuration necessary to set up such repositories.

### 1.11.6 Mercurial

Нове в версії 2.1.

Mercurial is another VCS you can use directly in Weblate.

---

**Примітка:** It should work with any Mercurial version, but there are sometimes incompatible changes to the command-line interface which breaks Weblate integration.

---

#### Дивись також:

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

### 1.11.7 Subversion

Нове в версії 2.8.

Weblate uses *git-svn* to interact with *subversion* repositories. It is a Perl script that lets subversion be used by a Git client, enabling users to maintain a full clone of the internal repository and commit locally.

---

**Примітка:** Weblate tries to detect Subversion repository layout automatically - it supports both direct URLs for branch or repositories with standard layout (branches/, tags/ and trunk/). More info about this is to be found in the *git-svn documentation*. If your repository does not have a standard layout and you encounter errors, try including the branch name in the repository URL and leaving branch empty.

---

Змінено в версії 2.19: Before this, there was only support for standard layout repositories.

### Реєстраційні дані Subversion

Weblate expects you to have accepted the certificate up-front and if needed, your credentials. It will look to insert them into the DATA\_DIR directory. Accept the certificate by using *svn* once with the *\$HOME* environment variable set to the DATA\_DIR:

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

Дивись також:

`DATA_DIR`

### 1.11.8 Локальні файли

Нове в версії 3.8.

Weblate can also operate without a remote VCS. The initial translations are imported by uploading them. Later you can replace individual files by file upload, or add translation strings directly from Weblate (currently available only for monolingual translations).

In the background Weblate creates a Git repository for you and all changes are tracked in. In case you later decide to use a VCS to store the translations, you already have a repository within Weblate can base your integration on.

## 1.12 Weblate's REST API

Нове в версії 2.6: The REST API is available since 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*.

### 1.12.1 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 i 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*) – результати 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

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.

### 1.12.2 Точка входу до програмного інтерфейсу

#### GET /api/

Коренева точка входу до програмного інтерфейсу.

##### Приклад запиту:

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

##### Приклад відповіді:

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

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

### 1.12.3 Користувачі

Нове в версії 4.0.

#### GET /api/users/

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

##### Дивись також:

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

#### POST /api/users/

Створює обліковий запис користувача.

##### Параметри

- **username** (*string*) – Ім'я користувача
- **full\_name** (*string*) – Повне ім'я користувача
- **email** (*string*) – Адреса електронної пошти користувача
- **is\_superuser** (*boolean*) – Чи є користувач 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/"
}
```

**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) /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** – Назва зареєстрованого сповіщення
- **subscription\_id** – int

**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** – Назва зареєстрованого сповіщення
- **subscription\_id** – int

#### 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** – Назва зареєстрованого сповіщення
- **subscription\_id** – int

#### Надіслати запит щодо об'єкта JSON

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

**DELETE** `/api/users/(str: username)/notifications/`  
**int:** *subscription\_id*/ Вилучити передплату, яку пов'язано із користувачем.

#### Параметри

- **username** (*string*) – Назва облікового запису користувача
- **subscription\_id** – Назва зареєстрованого сповіщення
- **subscription\_id** – int

### 1.12.4 Групи

Нове в версії 4.0.

#### **GET** `/api/groups/`

Повертає список груп, якщо у вас є права доступу для перегляду групи керування. Якщо прав немає, ви зможете переглянути лише групи, частиною яких є користувач.

Дивись також:

Атрибути об'єкта групи описано у `GET /api/groups/(int:id)/`.

#### **POST** `/api/groups/`

Створює групу.

#### Параметри

- **name** (*string*) – Назва групи
- **project\_selection** (*int*) – Вибір групи 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

### 1.12.5 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

## 1.12.6 Мови

**GET** `/api/languages/`

Повертає список усіх мов.

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

Створює мову.

**Параметри**

- **code** (*string*) – Назва мови
- **name** (*string*) – Назва мови
- **direction** (*string*) – Напрямок 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
  },
}
```

(continues on next page)

(continued from previous page)

```

"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

### 1.12.7 Проєкти

**GET** /api/projects/

Повертає список усіх проєктів.

Дивись також:

Атрибути об'єкта проєкту описано у [GET /api/projects/\(string:project\)/](#).

**POST** /api/projects/

Нове в версії 3.9.

Створює проєкт.

#### Параметри

- **name** (*string*) – Назва проєкту
- **slug** (*string*) – Ідентифікатор проєкту
- **web** (*string*) – Вебсайт проєкту
- **source\_language** (*string*) – Код початкової мови проєкту (необов'язковий)

**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
- **source\_language** (*object*) – об'єкт початкової мови, див. [GET /api/languages/\(string:language\)/](#)
- **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/](#)

Приклад даних JSON:

```
{
  "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/"
}
```

**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 проєкту

#### 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 проекті.

#### Параметри

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

#### 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.

Створює складники перекладу у вказаному проекті.

#### Параметри

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

#### 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",
```

(continues on next page)

(continued from previous page)

```

"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/"
}

```

**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

### 1.12.8 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 проекту
- **component** (*string*) – Ідентифікатор адрес складника

Об'єкт відповіді JSON

- **project** (*object*) – проект перекладу, див. `GET /api/projects/(string:project)/`
- **name** (*string*) – Назва складника
- **slug** (*string*) – Ідентифікатор складника
- **vcs** (*string*) – Система керування версіями
- **repo** (*string*) – 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
- **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 пропозиції
- **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/`

#### 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",

```

(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",
"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

#### 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

```

(continues on next page)



(continued from previous page)

```
{
  "branch": "master",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "new name",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "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,
```

(continues on next page)

(continued from previous page)

```

"share_url": "http://example.com/engage/hello/cs/",
"total": 4,
"total_words": 15,
"translate_url": "http://example.com/translate/hello/weblate/cs/",
"translated": 0,
"translated_percent": 0.0,
"translated_words": 0,
"url": "http://example.com/api/translations/hello/weblate/cs/",
"web_url": "http://example.com/projects/hello/weblate/cs/"
}

```

**GET /api/components/(string: project) /**  
**string:** *component/statistics/* Повертає поділену на сторінки статистику для 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/*

## 1.12.9 Переклади

**GET /api/translations/**  
 Повертає список перекладів.

Дивись також:

Атрибути об'єкта перекладу документовано у *GET /api/translations/(string:project)/(string:component)/(string:language)/*.

**GET /api/translations/(string: project) /**  
**string:** *component/string: language/* Повертає відомості щодо перекладу.

#### Параметри

- **project** (*string*) – Ідентифікатор адреси 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 файла 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",
```

(continues on next page)

(continued from previous page)

```

        "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",
"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

**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)

### 1.12.10 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) /`

Повертає відомості щодо модуля перекладу.

#### Параметри

- **id** (*int*) – Ід. модуля

#### Об'єкт відповіді JSON

- **translation** (*string*) – Адреса пов'язаного об'єкта перекладу
- **source** (*string*) – початковий рядок
- **previous\_source** (*string*) – попередній початковий рядок, який використано для встановлення приблизної відповідності
- **target** (*string*) – 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
- **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
- **web\_url** (*string*) – адреса, за якою можна редагувати модуль
- **source\_info** (*string*) – Посилання на дані початкового рядка; див. `GET /api/units/(int:id)/`

### 1.12.11 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

### 1.12.12 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
- **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)/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 складників

**Об'єкт відповіді JSON**

- **name** (*string*) – назва знімка вікна
- **component** (*string*) – адреса пов'язаного об'єкта складника
- **file\_url** (*string*) – адреса для отримання файла; див. *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – пов'язати із відповідними даними початкового рядка, див. *GET /api/units/(int:id)/*

**PATCH /api/screenshots/(int: id) /**

Редагувати часткові відомості щодо знімка вікна.

**Параметри**

- **id** (*int*) – Ідентифікатор знімка вікна

**Об'єкт відповіді JSON**

- **name** (*string*) – назва знімка вікна
- **component** (*string*) – адреса пов'язаного об'єкта складника
- **file\_url** (*string*) – адреса для отримання файла; див. *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – пов'язати із відповідними даними початкового рядка, див. *GET /api/units/(int:id)/*

**PUT /api/screenshots/(int: id) /**

Редагувати повні відомості щодо знімка вікна.

**Параметри**

- **id** (*int*) – Ідентифікатор знімка вікна

**Об'єкт відповіді JSON**

- **name** (*string*) – назва знімка вікна
- **component** (*string*) – адреса пов'язаного об'єкта складника
- **file\_url** (*string*) – адреса для отримання файла; див. *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – пов'язати із відповідними даними початкового рядка, див. *GET /api/units/(int:id)/*

**DELETE /api/screenshots/(int: id) /**

Вилучити знімок.

**Параметри**

- **id** (*int*) – Ідентифікатор знімка вікна

### 1.12.13 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

### 1.12.14 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/github/extending-github/about-webhooks> Загальні відомості щодо веб-скриптів 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 веб-скриптів 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://confluence.atlassian.com/bitbucket/manage-webhooks-735643732.html> Загальні відомості щодо вебскриптів Bitbucket

**ENABLE\_HOOKS** Вмикання скриптів для усього екземпляра Weblate

#### **POST /hooks/pagure/**

Нове в версії 3.3.

Особливий скрипт для обробки сповіщень Pagure і автоматичного оновлення відповідних компонентів.

**Дивись також:**

*Автоматичне отримання змін з Pagure* Настанови щодо налаштування інтеграції із Pagure

[https://docs.pagure.org/pagure/usage/using\\_webhooks.html](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> 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

## 1.12.15 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 \u010ciha\u0159",
    "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/"
  },
  {
```

(continues on next page)

(continued from previous page)

```

    "code": "nl",
    "failing": 21,
    "failing_percent": 4.8,
    "fuzzy": 11,
    "fuzzy_percent": 2.5,
    "last_author": null,
    "last_change": null,
    "name": "Dutch",
    "total": 436,
    "total_words": 15271,
    "translated": 319,
    "translated_percent": 73.2,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/nl/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/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/
→"
  },
]

```

### 1.12.16 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](#)

## 1.13 Клієнт Weblate

Нове в версії 2.7: There has been full `wlc` utility support ever since Weblate 2.7. If you are using an older version some incompatibilities with the API might occur.

### 1.13.1 Встановлення

The Weblate Client is shipped separately and includes the Python module. To use the commands below, you need to install `wlc`:

```
pip3 install wlc
```

### 1.13.2 Перші кроки

The `wlc` configuration is stored in `~/.config/weblate`, please create it to match your environment:

```
[weblate]
url = https://hosted.weblate.org/api/

[keys]
https://hosted.weblate.org/api/ = APIKEY
```

Далі, ви можете викликати команди на типовому сервері:

```
wlc ls
wlc commit sandbox/hello-world
```

**Дивись також:**

*Файли налаштувань*

### 1.13.3 Короткий опис

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

Команди вказують на те, яку дію має бути виконано.

### 1.13.4 Опис

Weblate Client is a Python library and command-line utility to manage Weblate remotely using [API](#). The command-line utility can be invoked as **wlc** and is built-in on `wlc`.

#### Параметри усього сайту

Програма приймає вказані нижче параметри для усього екземпляра Weblate. Ці параметри можна ввести перед будь-якою допоміжною командою.

**--format** {csv,json,text,html}  
Вказати формат виведення даних.

**--url** URL  
Specify the API URL. Overrides any value found in the configuration file, see *Файли налаштувань*. The URL should end with `/api/`, for example `https://hosted.weblate.org/api/`.

**--key** KEY

Specify the API user key to use. Overrides any value found in the configuration file, see [Файли налаштувань](#). You can find your key in your profile on Weblate.

**--config** PATH

Перевизначає шлях до файла налаштувань, див. [Файли налаштувань](#).

**--config-section** SECTION

Перевизначає розділ файла налаштувань для використання, див. [Файли налаштувань](#).

**Допоміжні команди**

Можна скористатися такими допоміжними 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.

Removes any untracked changes in a Weblate object to match the remote repository (translation, component or project).

**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.

## 1.13.5 Файли налаштувань

**.weblate** Файл налаштувань окремого проєкту

**~/.config/weblate** Файл налаштувань користувача

**/etc/xdg/weblate** Загальносистемний файл налаштувань

The program follows the XDG specification, so you can adjust placement of config files by environment variables `XDG_CONFIG_HOME` or `XDG_CONFIG_DIRS`.

Following settings can be configured in the `[weblate]` section (you can customize this by `--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 проєкту.

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
```

This allows you to store keys in your personal settings, while using the `.weblate` configuration in the VCS repository so that `wlc` knows which server it should talk to.

### 1.13.6 Приклади

Вивести версію поточної програми:

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

Список усіх проєктів:

```
$ wlc list-projects
name: Hello
slug: hello
source_language: en
url: http://example.com/api/projects/hello/
web: https://weblate.org/
web_url: http://example.com/projects/hello/
```

You can also designate what project `wlc` should work on:

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

$ wlc show
branch: master
file_format: po
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
```

## 1.14 Програмний інтерфейс Python Weblate

### 1.14.1 Встановлення

The Python API is shipped separately, you need to install the *Клієнт Weblate*: (wlc) to have it.

```
pip install wlc
```

### 1.14.2 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.

Access class to the API, define API key and optionally API URL.

**get** (*path*)

**Параметри** **path** (*str*) – Шлях запити

**Тип повернення** *object*

Виконує один виклик GET програмного інтерфейсу.

**post** (*path*, *\*\*kwargs*)

**Параметри** **path** (*str*) – Шлях запити

**Тип повернення** *object*

Виконує один виклик GET програмного інтерфейсу.

### 1.14.3 wlc.config

#### WeblateConfig

**class** wlc.config.**WeblateConfig** (*section='wlc'*)

**Параметри** **section** (*str*) – Розділ налаштувань, яким слід скористатися

Configuration file parser following XDG specification.

**load** (*path=None*)

**Параметри** **path** (*str*) – Шлях, з якого слід завантажувати налаштування.

Завантажує налаштування з файла. Якщо файл не вказано, завантаження відбуватиметься з файла налаштувань *wlc* (*~/.config/wlc*), який розташовано у вашому каталозі налаштувань XDG (*/etc/xdg/wlc*).

### 1.14.4 wlc.main

wlc.main.**main** (*settings=None, stdout=None, args=None*)

**Параметри**

- **settings** (*list*) – Параметри для перевизначення у форматі списку кортежів
- **stdout** (*object*) – Файловий об'єкт *stdout* для виведення даних. Типовим є *sys.stdout*
- **args** (*list*) – Аргументи командного рядка для обробки. Типовими є *sys.args*

Головна точка входу для інтерфейсу командного рядка.

@wlc.main.**register\_command** (*command*)

Decorator to register *Command* class in main parser used by *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.

---

**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 загально-доступній 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. Див. [ваду 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 дозволені 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 посиланнях 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 виключення адрес для облікового запису, потрібного для усього 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 змінною [GOOGLE\\_ANALYTICS\\_ID](#).

#### WEBLATE\_GITHUB\_USERNAME

Налаштовує ім'я користувача GitHub для запитів щодо об'єднання GitHub змінною [GITHUB\\_USERNAME](#).

Дивись також:

[GitHub](#), [Налаштовування концентратора](#)

#### WEBLATE\_GITLAB\_USERNAME

Налаштовує ім'я користувача GitLab для запитів щодо об'єднання GitLab змінною [GITLAB\\_USERNAME](#)

Дивись також:

[GitLab Налаштовування Lab](#)

#### WEBLATE\_GITLAB\_HOST

Налаштовує вузол GitLab для запитів щодо об'єднання GitLab

Дивись також:

[GitLab Налаштовування Lab](#)

#### WEBLATE\_GITLAB\_TOKEN

Налаштовує ключ доступу GitLab для запитів щодо об'єднання GitLab

Дивись також:

[GitLab Налаштовування Lab](#)

#### 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\_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, результати яких ви не хочете бачити, див. *SILENCED\_SYSTEM\_CHECKS*.

### WEBLATE\_CSP\_SCRIPT\_SRC

### WEBLATE\_CSP\_IMG\_SRC

### WEBLATE\_CSP\_CONNECT\_SRC

### WEBLATE\_CSP\_STYLE\_SRC

### WEBLATE\_CSP\_FONT\_SRC

Надає змогу налаштовувати заголовок HTTP Content-Security-Policy.

**Дивись також:**

*Правила щодо безпеки даних, CSP\_SCRIPT\_SRC, CSP\_IMG\_SRC, CSP\_CONNECT\_SRC, CSP\_STYLE\_SRC, CSP\_FONT\_SRC*

## Параметри комп'ютерного перекладу

### 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

Вмикає *Когнітивні служби перекладу Microsoft* і встановлює *MT\_MICROSOFT\_ENDPOINT\_URL*

### WEBLATE\_MT\_MICROSOFT\_BASE\_URL

Вмикає *Когнітивні служби перекладу Microsoft* і 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**

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 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](#)

**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).

## 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\_TRANSLATE\_OPTIONS**

**CELERY\_MEMORY\_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
```

### 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*

## Налаштування Hub

Щоб мати змогу користуватися можливістю зі створення запитів щодо об'єднання GitHub, вам слід ініціалізувати налаштування Hub введенням контейнера Weblate і виконанням довільної команди Hub. Приклад:

```
docker-compose exec --user weblate weblate bash
cd
HOME=/app/data/home hub clone octocat/Spoon-Knife
```

Ім'я користувача, яке передається для реєстрації, має бути тим самим, що і у `GITHUB_USERNAME`.

**Дивись також:**

*GitHub, Налаштовування концентратора*

### Налаштування Lab

Щоб мати змогу користуватися можливістю зі створення запитів щодо злиття GitLab, вам слід ініціалізувати налаштування lab введенням контейнера Weblate і виконанням довільної команди lab. Приклад:

```
docker-compose exec --user weblate weblate bash
cd
HOME=/app/data/home lab
```

Ви також можете скористатися змінними середовища для налаштування lab при кожному запуску контейнера. Просто додайте `WEBLATE_GITLAB_USERNAME`, `WEBLATE_GITLAB_HOST` та `WEBLATE_GITLAB_TOKEN` до налаштувань вашого середовища.

```
weblate:
  environment:
    WEBLATE_GITLAB_USERNAME: translations_bot
    WEBLATE_GITLAB_HOST: https://gitlab.example.com
    WEBLATE_GITLAB_TOKEN: personal_access_token_of_translations_bot
```

Значення `access_token`, яке передається при налаштуванні lab, має збігатися зі значенням `GITLAB_USERNAME`.

**Дивись також:**

*GitLab Налаштовування Lab*

### 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.

**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 libfreetype6-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

# 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. 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
```

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 інтерфейс (`http://localhost:8000/create/project/`) і створіть проєкт, який ви хочете перекладати. Див. *Налаштування проєкту*, щоб дізнатися більше.

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

2. Створіть складник, який є справжнім об'єктом для перекладу — він вказує на 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.

---

**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-
    ↪devel \
    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

# Database server: PostgreSQL
zypper install postgresql postgresql-contrib

# SMTP server
zypper install postfix
```

### Модулі Python

---

**Підказка:** Ми використовуємо virtualenv для встановлення Weblate в окремому середовищі з вашої системи. Якщо ви не при звичаєні до такого встановлення, ознайомтеся із *User Guide* до virtualenv.

---

1. Створіть virtualenv для Weblate:

```
virtualenv --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 інтерфейс (`http://localhost:8000/create/project/`) і створіть проєкт, який ви хочете перекладати. Див. *Налаштування проєкту*, щоб дізнатися більше.

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

2. Створіть складник, який є справжнім об'єктом для перекладу — він вказує на сховище системи керування версіями і визначає, які файли слід перекладати. Див. *Налаштування 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 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.

---

**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

# Database server: PostgreSQL
dnf install postgresql postgresql-contrib

# SMTP server
dnf install postfix
```

## Модулі Python

**Підказка:** Ми використовуємо virtualenv для встановлення Weblate в окремому середовищі з вашої системи. Якщо ви не при звичаєні до такого встановлення, ознайомтеся із [User Guide](#) до virtualenv.

1. Створіть virtualenv для Weblate:

```
virtualenv --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 інтерфейс (`http://localhost:8000/create/project/`) і створіть проєкт, який ви хочете перекладати. Див. *Налаштування проєкту*, щоб дізнатися більше.

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

2. Створіть складник, який є справжнім об'єктом для перекладу — він вказує на 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.

---

**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
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
```

- 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`.

- Скопіюйте файл `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` до `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`
- 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*.
- 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

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

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

2. Створіть складник, який є справжнім об'єктом для перекладу — він вказує на сховище системи керування версіями і визначає, які файли слід перекладати. Див. *Налаштування 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 з початкового коду

1. Будь ласка, спочатку виконайте настанові 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. 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*.
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

**Примітка:** Для написання цієї частини підручника нам потрібні автори, які мають досвід роботи із OpenShift, див. <https://github.com/WeblateOrg/weblate/issues/2889>.

У Weblate передбачено підтримку OpenShift. Потрібні для інтеграції файли зберігаються в основному сховищі коду у підкаталозі `openshift3`.

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

- *Встановлення за допомогою 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*
- *Встановлення з початкового коду*, рекомендовано для розробки.
- *Встановлення на OpenShift*.

### 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*

**hub** (необов'язкова, призначено для запитів щодо об'єднання на GitHub) <https://hub.github.com/>

**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>

## 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. *Promoting the translation*) 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>.

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 (Brains) <michal.cihar@brains.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 (Brains) <michal.cihar@brains.cz>" [ultimate]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:      There is no indication that the signature belongs to the owner.
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE  5A32 9C27 B313 42B7 511D
```

Проблема полягає у тому, що випустити ключ для цього імені може будь-хто. Вам слід переконатися, що ключ справді належить згаданій особі. У підручнику з конфіденційності GNU цю тему висвітлено у розділі [Validating other keys on your public keyring](#). Найнадійнішим способом встановлення автентичності ключа є особиста зустріч із розробником і обмін відбитками ключів. Втім, ви також можете покластися на мережу довіри. У такій мережі ви можете довіритися ключ опосередковано — через підписи інших осіб, які зустрічалися із розробником особисто.

Щойно ключ стане довіреним, має бути показано попередження:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:      using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:      aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:      aka "[jpeg image of size 8848]" [ultimate]
gpg:      aka "Michal Čihař (Brains) <michal.cihar@brains.cz>" [ultimate]
```

Якщо підпис є некоректним (до архіву було внесено зміни), ви отримаєте ясне повідомлення щодо помилки, незалежно від того, є ключ довіреним чи ні:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:      using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: BAD signature from "Michal Čihař <michal@cihar.com>" [ultimate]
```

### 2.1.4 Права доступу у файловій системі

Процес Weblate повинен мати можливість читати дані з каталогу, де зберігає дані, і записувати дані до нього — `DATA_DIR`. Усі файли у цьому каталозі мають належати користувачеві, від імені якого запущено 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.

**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',
        # Database password
        'PASSWORD': 'password',
        # Set to empty string for localhost
        'HOST': 'database.example.com',
        # Set to empty string for default
        'PORT': '',
    }
}
```

### MySQL і MariaDB

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.

---

**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.

Через це, ми рекомендуємо використовувати для нових встановлених екземплярів [PostgreSQL](#).

---

Рекомендовано такі налаштування для Weblate:

- Скористайтеся набором символів `utf8mb4`, щоб уможливити обробку вищих блоків Unicode (зокрема емоджі).
- Налаштуйте сервер за допомогою `InnoDB_large_prefix` так, щоб уможливити довші індекси у текстових полях.
- Встановіть рівень ізоляції `READ COMMITTED`.
- Режим SQL має бути встановлено у значення `STRICT_TRANS_TABLES`.

## 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_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.

**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).

## 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 видалення застарілих даних сеансу з бази даних.

Якщо ви використовуєте Redis як кеш (див. [Увімкніть кешування](#)) рекомендуємо скористатися ним і для сеансів:

```
SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
```

**Дивись також:**

[Configuring the session engine](#), `SESSION_ENGINE`

### DATABASES

Можливість з'єднання із сервером бази даних. Будь ласка, ознайомтеся із документацією до Django, щоб дізнатися більше.

**Дивись також:**

[Налаштування бази даних для Weblate](#), `DATABASES`, [Databases](#)

### DEBUG

Вимкніть цю можливість для будь-яких промислових серверів. Якщо діагностичний режим увімкнено, Django показуватиме користувачам дані зворотного трасування при помилці. Якщо ви вимкнете її, помилки буде надіслано електронною поштою до ADMINS (див. вище).

Режим діагностики також уповільнює роботу Weblate, оскільки у такому режимі Django на внутрішньому рівні зберігає набагато більше даних.

**Дивись також:**

`DEBUG`

### 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`.

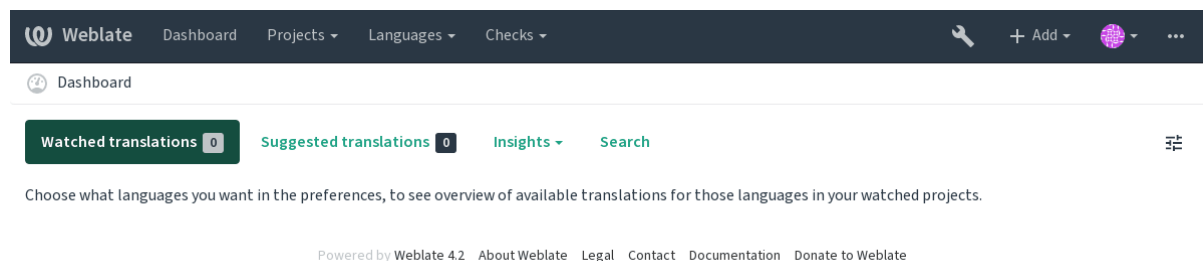
Щойно потрібні дії буде виконано, вам слід також ознайомитися зі *Звітом щодо швидкодії* в інтерфейсі адміністратора, там ви зможете знайти підказки щодо потенційних недоліків у налаштуваннях на вашому сайті.

**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 результати 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:**

*Налаштування 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, Налаштування 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. *«Лінії» внески* і *commit\_pending*.
- Оновлення нагадувань щодо складників (щоденно).
- Оновлення віддалених гілок (щодня), див. *AUTO\_UPDATE*.
- Резервного копіювання пам'яті перекладів до JSON (щодня), див. *dump\_memory*.
- Завдань із супроводу обробки 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**

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/
    Alias /media/ /home/weblate/data/media/
```

(continues on next page)

(continued from previous page)

```

<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
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/
    <Directory /home/weblate/data/media/>

```

(continues on next page)

(continued from previous page)

```

    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.

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
    WSGIApplicationGroup %{GLOBAL}

    WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.7/site-packages/
↪weblate/wsgi.py process-group=weblate

```

(continues on next page)

(continued from previous page)

```

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
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}'

```

(continues on next page)

(continued from previous page)

```
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 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 --concurrency:celery=4 --queues:celery=celery --
↪prefetch-multiplier:celery=4 \
--concurrency:notify=4 --queues:notify=notify --prefetch-multiplier:notify=10 \
--concurrency:translate=4 --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 «*Лінійні*» *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 дамів 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*

### 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. *Встановлення за допомогою 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 версій* 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 непідтримуваними — вони не працюватимуть як слід.

**Дивись також:**

[Документація із оновлення з 2.20 до 3.0 у документації до Weblate 3.0](#)

#### Оновлення з версії 3.x

Якщо ви оновлюєте систему з версії 3.x, завжди спочатку оновіться до версії 4.0.4, а потім продовжіть оновлення у межах послідовності версій 4.x. Оновлення, у яких пропущено цей крок, вважаються непідтримуваними — вони не працюватимуть як слід.

**Дивись також:**

[Документація із оновлення з 3.11 до 4.0 у документації до Weblate 4.0](#)

#### Оновлення з 4.0 до 4.1

Будь ласка, виконайте настанови із розділу *Загальні настанови щодо оновлення* для виконання оновлення.

Помітні зміни у налаштуваннях або залежностях:

- До `settings_example.py` внесено декілька змін. Найпомітнішими є зміна проміжного програмного забезпечення. Будь ласка, скоригуйте параметри відповідним чином.
- Передбачено нові формати файлів. Можливо, вам варто включити їх до списку зміненого `WEBLATE_FORMATS`.
- Передбачено нові перевірки якості. Можливо, вам варто включити їх до списку зміненого `CHECK_LIST`.
- Змінено параметр `DEFAULT_THROTTLE_CLASSES` так, щоб уможливити звітування щодо обмеження частоти у програмному інтерфейсі.
- Маємо деякі нові і оновлені вимоги щодо програмного забезпечення.
- Змінено `INSTALLED_APPS`.
- 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 покладаетесь 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.

**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',
        # 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 резервне копіювання

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, оскільки у випуску 4.0 передбачено підтримку лише цього рушія. Див. *Перенесення даних з інших баз даних до PostgreSQL*.

Dashboard
Projects ▾
Languages ▾
Checks ▾
+ Add ▾
...

Manage / Backups

Backup process triggered

Backup service credentials Aug. 18, 2020

Backup repository /tmp/tmp7e9uy3d1weblate

Passphrase 3AYd5(It9NmOjF0n56@zVHaZ5Ab(a&bcE4gf5pBjxoDS1IrPVM

The passphrase is used to encrypt the backups and is necessary to restore them.

SSH key

Download private key

Deleted the oldest backups Aug. 18, 2020

Backup performed Aug. 18, 2020

Repository initialization Aug. 18, 2020

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

Add

Powered by Weblate 4.2
 [About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

## Використання передбаченого у Weblate сховища резервних копій

Найпростішим способом резервного копіювання вашого екземпляра Weblate є придбання служби резервного копіювання на [weblate.org](https://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 забезпечення ваших даних. Без вашої згоди під час процедури реєстрації до сховища резервних копій не буде надіслано жодних даних.

2.4. Резервне копіювання і пересування Weblate

205



### 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

### 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 зберегти його у безпечному місці.

Якщо ви користуєтеся *Використання передбаченого у Weblate сховища 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`

### 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 CXO-ВИЩЕ : : АРХІВ`.
4. Відновіть базу даних із дампу SQL із розташування у каталозі `backup` каталогу даних Weblate (див. *Дампи даних для резервних копій*).
5. Скопіюйте налаштування Weblate і каталог даних до поточного розташування.

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`

## 2.4.2 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 повнішої резервної копії. Файли оновлюються щоденно (це потребує запуску сервера Celery, див. *Фонові завдання з використанням Celery*). У поточній версії сюди включено:

- Параметри Weblate у файлі `settings.py` (розширена версія зберігається у `settings-expanded.py`).
- Резервна копія бази даних PostgreSQL у файлі `database.sql`.

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*).

### 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
```

### 2.4.3 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*

### 2.4.4 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
```

### 2.4.5 Пересування встановленого екземпляра Weblate

Перенесіть ваш екземпляр на іншу систему за допомогою настанов зі створення і відновлення резервних копій, які наведено вище.

**Дивись також:**

*Оновлення з Python 2 до Python 3, Перенесення даних з інших баз даних до PostgreSQL*

## 2.5 Розпізнавання

### 2.5.1 Реєстрація користувача

За типових налаштувань Weblate використовує python-social-auth, форму на сайті для обробки реєстрації нових користувачів. Після підтвердження адреси електронної пошти новий користувач може робити внески або проходити розпізнавання за допомогою однієї зі сторонніх служб.

Ви також можете вимкнути реєстрацію нових користувачів за допомогою параметра `REGISTRATION_OPEN`.

Спроби розпізнавання описано у *Обмеження частоти*.

### 2.5.2 Модулі розпізнавання

Для розпізнавання використовується вбудоване рішення Django, у якому передбачено різноманітні варіанти розпізнавань для соціальних мереж. Використання цього рішення означає, що ви можете імпортувати базу даних користувачів інших заснованих на Django проєктів (див. *Перенесення даних з Pootle*).

Django можна додатково налаштувати для розпізнавання в інших системах розпізнавання.

**Дивись також:**

У розділі *Параметри розпізнавання* описано, як налаштувати розпізнавання у офіційному образі Docker.

### 2.5.3 Розпізнавання за соціальними мережами

Завдяки [Welcome to Python Social Auth's documentation!](#), у Weblate передбачено підтримку розпізнавання за допомогою сторонніх служб, зокрема GitLab, Ubuntu, Fedora тощо.

Будь ласка, ознайомтеся із документацією щодо типових настанов з налаштування [Django Framework](#).

---

**Примітка:** Типово, Weblate покладається на сторонні служби розпізнавання для забезпечення коректності адреси електронної пошти. Якщо у якихось із служб, якими ви хочете скористатися, підтримки перевірки коректності адреси електронної пошти не передбачено, будь ласка, скористайтеся примусовою перевіркою адреси електронної пошти на боці Weblate, налаштувавши для таких служб параметр `FORCE_EMAIL_VALIDATION`. Приклад:

```
SOCIAL_AUTH_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. Будь ласка, 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 сайта*.

Дивись також:

[GitHub](#)

## Розпізнавання за Bitbucket

Вам слід зареєструвати програму на Bitbucket, а потім повідомити Weblate усі її реєстраційні дані:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    'social_core.backends.bitbucket.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 сайта*.

Дивись також:

[Bitbucket](#)

## Google OAuth 2

Щоб скористатися OAuth 2 Google, вам слід зареєструвати програму на <https://console.developers.google.com/> і увімкнути програмний інтерфейс Google+.

Адреса переспрямування — `https://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>](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 login
# 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 login
# 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 зможу 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 із анонімними користувачами. Цей параметр також надає вам змогу входити до системи за допомогою локального адміністративного облікового запису, якщо ви його створили (наприклад, за допомогою `createadmin`).

---

## Використання пароля bind

Якщо ви не можете скористатися безпосередньою прив'язкою для розпізнавання, вам доведеться скористатися пошуком і вказати користувача для прив'язки пошуку. Приклад:

```
import ldap
from django_auth_ldap.config import LDAPSearch

AUTH_LDAP_BIND_DN = ""
AUTH_LDAP_BIND_PASSWORD = ""
AUTH_LDAP_USER_SEARCH = LDAPSearch("ou=users,dc=example,dc=com",
    ldap.SCOPE_SUBTREE, "(uid=%(user)s)")
```

## Інтеграція із Active Directory

```
import ldap
from django_auth_ldap.config import LDAPSearch, NestedActiveDirectoryGroupType

AUTH_LDAP_BIND_DN = "CN=ldap,CN=Users,DC=example,DC=com"
AUTH_LDAP_BIND_PASSWORD = "password"

# User and group search objects and types
AUTH_LDAP_USER_SEARCH = LDAPSearch("CN=Users,DC=example,DC=com", ldap.SCOPE_
↳SUBTREE, "(sAMAccountName=%(user)s)")

# Make selected group a superuser in Weblate
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    # is_superuser means user has all permissions
    "is_superuser": "CN=weblate_AdminUsers,OU=Groups,DC=example,DC=com",
}

# Map groups from AD to Weblate
AUTH_LDAP_GROUP_SEARCH = LDAPSearch("OU=Groups,DC=example,DC=com", ldap.SCOPE_
↳SUBTREE, "(objectClass=group)")
AUTH_LDAP_GROUP_TYPE = NestedActiveDirectoryGroupType()
AUTH_LDAP_FIND_GROUP_PERMS = True

# Optionally enable group mirroring from LDAP to Weblate
# AUTH_LDAP_MIRROR_GROUPS = True
```

### Дивись також:

Django Authentication Using LDAP, Authentication

## 2.5.7 Розпізнавання за допомогою CAS

Щоб скористатися розпізнаванням за допомогою CAS, встановіть пакунок подібний до *django-cas-ng*.

Першим кроком є розкриття поля адреси електронної пошти користувача за допомогою CAS. Цю можливість має бути налаштовано на самому сервері CAS, вона потребує роботи із принаймні версією 2 CAS, оскільки у версії 1 CAS не передбачено підтримки атрибутів взагалі.

Другим кроком є оновлення Weblate для використання вашого сервера CAS і атрибутів.

Щоб встановити *django-cas-ng*, виконайте такі дії:

```
pip install django-cas-ng
```

Після встановлення пакунка, ви можете вставити його до скрипту розпізнавання Django внесенням змін до файла `settings.py`:

```
# Add CAS backed, keep the Django one if you want to be able to sign in
# even without LDAP for the admin account
AUTHENTICATION_BACKENDS = (
    'django_cas_ng.backends.CASBackend',
    'weblate.accounts.auth.WeblateUserBackend',
)

# CAS server address
CAS_SERVER_URL = 'https://cas.example.net/cas/'

# Add django_cas_ng somewhere in the list of INSTALLED_APPS
INSTALLED_APPS = (
    ... ,
```

(continues on next page)

(continued from previous page)

```
'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 = (
    ...
    'weblate',
    # 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. Прискіпніше налаштування можна виконати за допомогою розділу [Нетипове керування доступом](#)

## 2.6.1 Типові налаштування

### Блокування Weblate

Щоб повністю заблокувати встановлений Weblate, ви можете скористатися `LOGIN_REQUIRED_URLS`, щоб змусити користувачів входити до системи, і `REGISTRATION_OPEN`, щоб запобігти новим реєстраціям користувачів.

### Права доступу на рівні сайту

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 налаштують як *Public* (див. [Керування доступом на рівні проєкту](#)).

### Права доступу на рівні проєктів

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

### Додавання прав доступу до мов, компонентів або проєктів

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.

---

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](#).

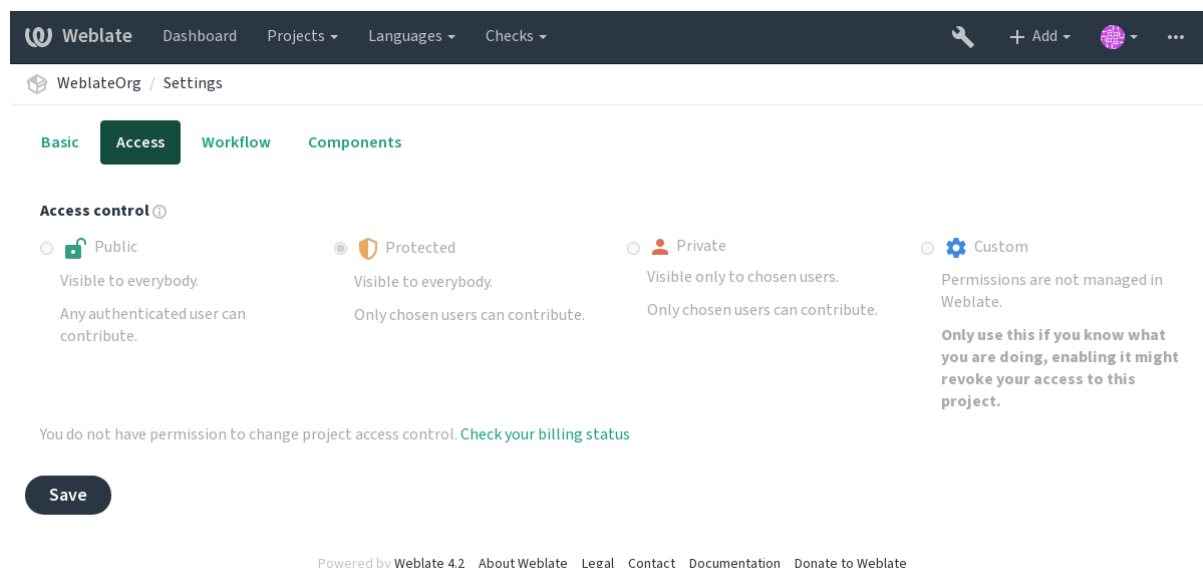
Для [Керування доступом](#) передбачено такі варіанти:

**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

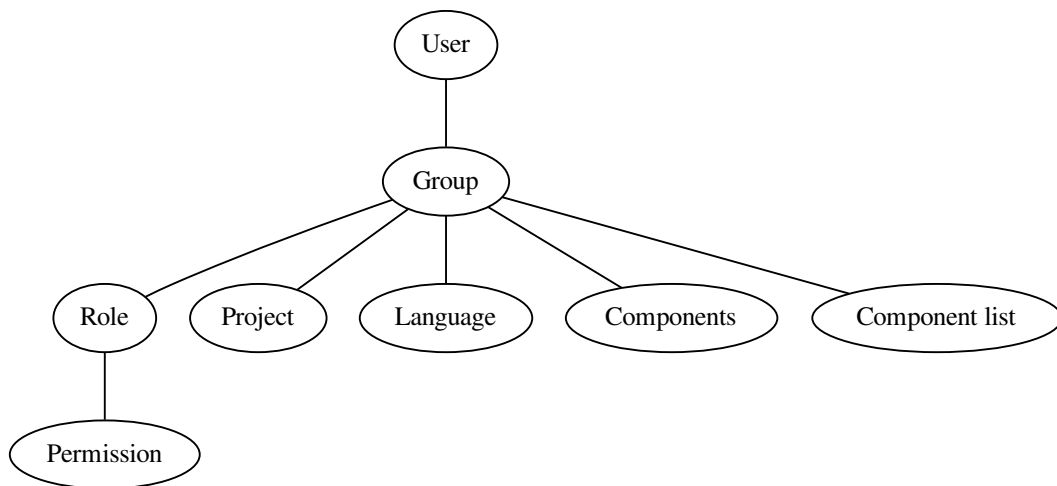
Моделі розпізнавання складаються з декількох об'єктів:

**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 (проєкти, 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 значення параметра *Список складників* складнику або проекту.
2. Встановлюється відповідність значення *Складники* складнику або проекту.
3. Встановлюється відповідність значення *Проекти* проекту.

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. *Languages* are matched against the scope of translations if set, if not set, this does not match any language.

---

**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. Учасі 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 проєктів, див. *Керування доступом на рівні проєкту*.


---

Користувачі з правами доступу *Може керувати правилами ACL проєкту* (див. *Керування доступом*) можуть також керувати записами користувачів у проєктах із увімкненим за допомогою сторінки проєкту керуванням доступом. За допомогою інтерфейсу ви можете виконувати такі дії:

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

Керувати користувачами можна за допомогою меню *Керувати проєкту*:


Dashboard
Projects ▾
Languages ▾
Checks ▾
+ Add ▾

...


Manage users

Users ⓘ

Username	Full name	E-mail	Last login	Administration	Billing	Glossary	Languages	Memory	Screenshots	Template	Translate	VCS
testuser	Weblate Test	weblate@example.org	13 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 4.2
[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.

**Template**

Може редагувати шаблони перекладу 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

**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*]

**Автоматизація** Використовувати автоматизацію [*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

Додати визначення мов

Керувати визначеннями мов

Додавати групи

Керувати групами

Додавати користувачів

Управління користувачами

Керувати оголошеннями

Керувати пам'яттю перекладів

---

**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 проєктів у встановленому вами Weblate.

---

## Список груп

Під час встановлення (або після виконання *setupgroups*) буде створено такі групи:

**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 системі 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 суперкористувачів і, якщо у вашому екземплярі використовується облік (наприклад, він використовується на <https://hosted.weblate.org/>, див. *Оплата*), 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'. The main content area is titled 'Your profile / Billing'. It is divided into two columns. The left column, 'Billing plan', shows the current plan as 'Basic plan (Active)' with a monthly price of 19 EUR and a yearly price of 199 EUR. It also shows usage limits for strings and languages (0 used) and a projects limit (0 of 1). The right column, 'Invoices', shows a table with one invoice for the period 08/17/2020 - 08/19/2020 for 19.0 EUR. At the bottom, there is a button to 'Add new translation project' and a 'Terminate billing plan' button.

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	2020-08-17 - 2020-08-19
Projects limit	Used 0 of 1
Projects	No projects currently assigned!

Invoices		
Invoice period	Invoice amount	Download invoice
08/17/2020 - 08/19/2020	19.0 EUR	Not available

Powered by Weblate 4.2 [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.

**Source language** ⓘ

English

Language used for source strings in all components


**Billing** ⓘ

Weblate Test (Basic plan)

Save

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Після створення проєкту ви перейдете безпосередньо до сторінки проєкту:

 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 4.2 [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 допомогою 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

**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 4.2 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 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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



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](#).

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 з перекладу

Адреса до додаткового сайту із 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.

## Джерельна мова

Мова, яку буде використано для початкових рядків в усіх складниках. 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*.

---

## Дивись також:

*Проміжний мовний файл, Шлюз якості для початкових рядків, Bilingual and monolingual formats, Визначення мов*

## Альтернативні назви мови

Визначте прив'язку кодів мов при імпортуванні перекладів до Weblate. Скористайтеся цим, якщо коди мов у ваших сховищах є нестандартними, і ви хочете мати однорідний вигляд у Weblate.

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`

---

**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.

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 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:

*Bilingual and monolingual formats, What does mean «There are more files for the single language (en)»?*

### Основний файл для одномовного перекладу

Базовий файл, у якому містяться визначення рядків для *Одномовні складники*.

#### Дивись також:

*Bilingual and monolingual formats, What does mean «There are more files for the single language (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, Bilingual and monolingual formats, What does mean «There are more files for the single language (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, Bilingual and monolingual formats, What does mean «There are more files for the single language (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:

*Додавання нових перекладів.*

### Стиль коду мови

Налаштуйте код мови, використаний для створення назви файлу для перекладів, створених Weblate, див. *Додавання нових перекладів*, щоб дізнатися більше.

### 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_COMMITTER_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_COMMITTER_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.

## 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>
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. [Варіанти рядків](#).

---

**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?, [Translation component 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 до-
датка.
```

Доступними у навігаторі сховища або шаблонах редактора є такі змінні:

```
{{branch}} поточна гілка
{{line}} рядок у файлі
{{filename}} назва файла, 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|stripext }}
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 швидкодію. Будь ласка, ознайомтеся із розділом *Промислові налаштування*, щоб дізнатися більше, особливо з такими підрозділами:

- Налаштуйте Celery для виконання фонових завдань (див. *Фонові завдання з використанням Celery*)
- *Увімкніть кешування*
- *Використання потужного рушія бази даних*
- *Вимкнути режим діагностики*

### Перевірка обмежень на ресурси

Якщо ви імпортуєте великі об'єми перекладів або 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 швидкодію.
- Вузьким місцем можуть бути дії із диском, якщо потрібно обробляти багато рядків — дані на диск записуватимуть одразу Weblate і база даних.
- Додаткові ядра процесора можуть підвищити швидкодію виконання фонових завдань (див. *Фонові завдання з використанням Celery*).

## Вимкніть непотрібні перевірки

Деякі перевірки якості є доволі витратними щодо ресурсів. Якщо вони є непотрібними, ви можете заощадити певний час під час імпортування, якщо ви пропустите їх. Див. [CHECK\\_LIST](#), щоб дізнатися більше про налаштування.

## 2.7.7 Автоматичне створення складників

Якщо у проєкт є з 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. До типового пакунка включено визначення для близько 350 мов.

### 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`).

Якщо і це не допоможе, нове визначення мови буде створено за допомогою типових значень (напрям запису тексту зліва праворуч, одна форма множини) із назвою мови `xx_XX` (*generated*). Ви можете змінити цю назву пізніше за допомогою адміністративного інтерфейсу (див. [Зміна визначень мов](#)) і повідомити про помилку у системі стеження за вадами (see [Contributing to Weblate](#)).

---

**Підказка:** Якщо у списку мов ви бачите якісь небажані записи, вам варто скоригувати [Фільтр мов](#) для ігнорування відповідного файла при обробці 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 напрямок запису тексту). Якщо ви цього не зробите, перекладачі не зможуть належним чином редагувати відповідні переклади.

## 2.8.3 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*

### 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:**

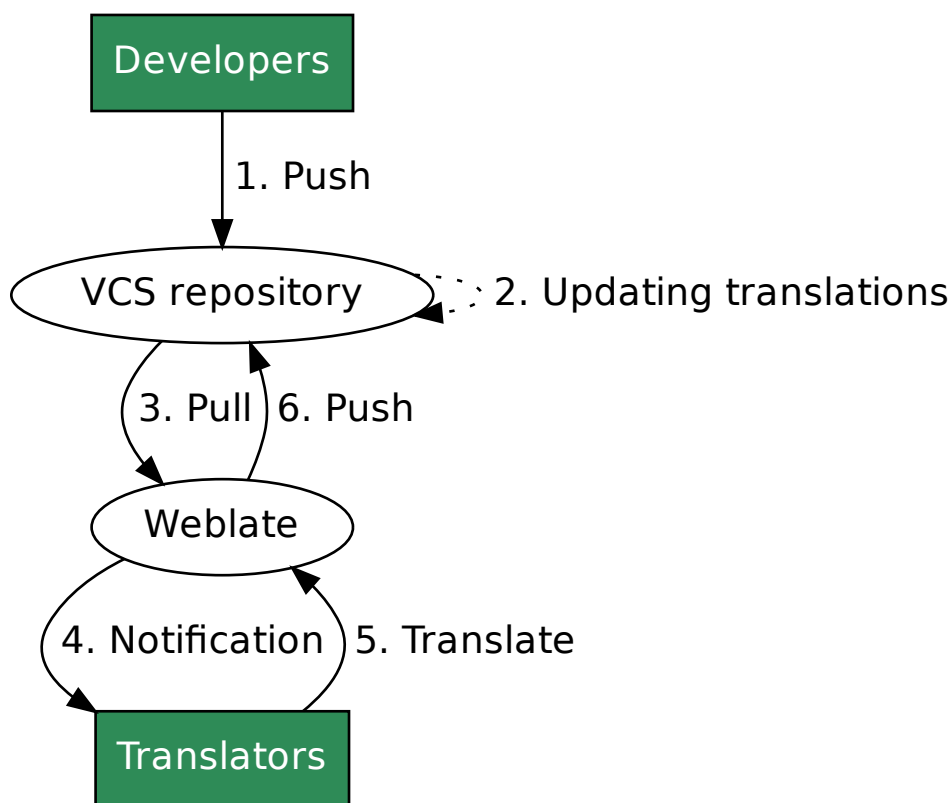
*Множина*, GNU gettext utilities: Plural forms, Language Plural Rules by the Unicode Consortium

## 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.

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. *Why does Weblate still show old translation strings when I've updated the template?*).
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 усування 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 передують 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 користуєтеся 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:

The screenshot shows the GitHub 'Add webhook' page for the repository 'WeblateOrg / hello'. The left sidebar contains navigation links: Options, Collaborators & teams, Branches, Webhooks (selected), Integrations & services, Deploy keys, and Alerts. The main content area is titled 'Webhooks / Add webhook' and contains the following fields and options:

- Webhooks / Add webhook**: Section header.
- Introduction**: A paragraph explaining that a POST request will be sent to the URL with details of subscribed events. It mentions that the data format can be JSON, x-www-form-urlencoded, etc., and refers to developer documentation.
- Payload URL \***: A text input field containing 'https://hosted.weblate.org/hooks/github/'.
- Content type**: A dropdown menu set to 'application/x-www-form-urlencoded'.
- Secret**: An empty text input field.
- SSL verification**: A checkbox labeled 'By default, we verify SSL certificates when delivering payloads.' with a red button 'Disable SSL verification'.
- Which events would you like to trigger this webhook?**: Three radio button options:
  - ☒ Just the push event.
  - ☐ Send me everything.
  - ☐ Let me select individual events.
- Active**: A checked checkbox with the text 'We will deliver event details when this hook is triggered.'
- Add webhook**: A green button at the bottom.

The footer of the page shows copyright information for 2018 GitHub, Inc., and links to Terms, Privacy, Security, Status, Help, Contact GitHub, API, Training, Shop, Blog, and About.

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

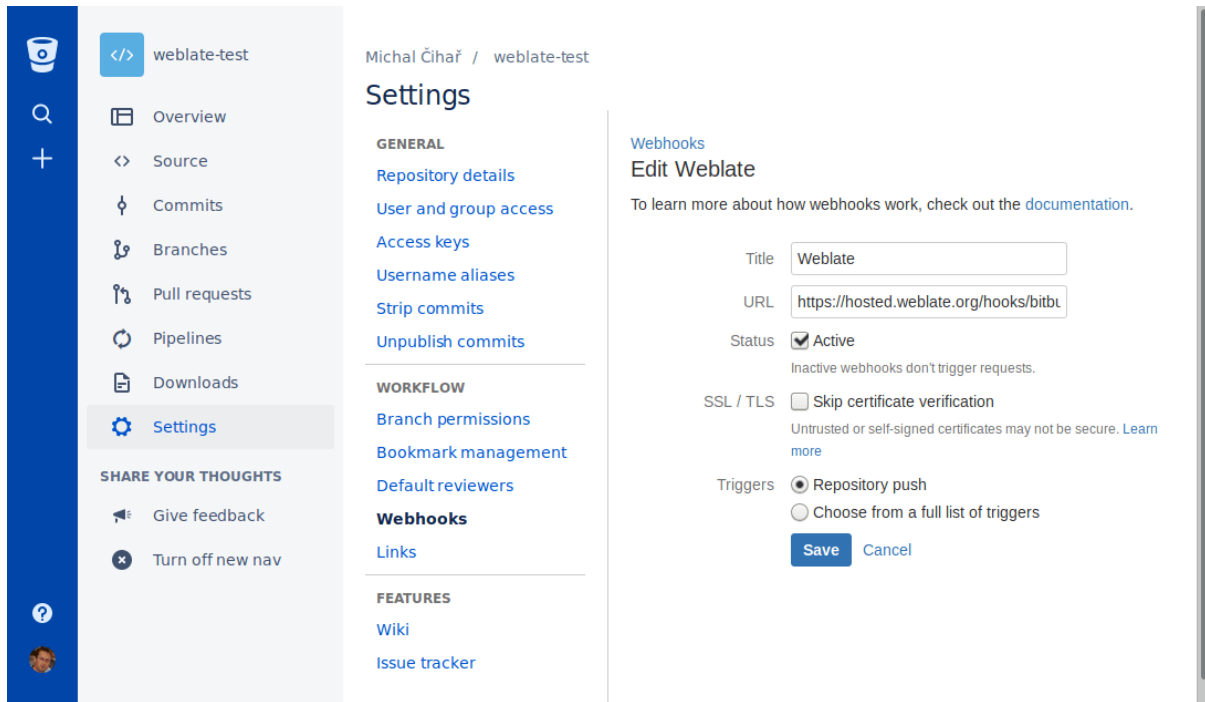
Ви можете лишити типові значення для решти параметрів (Weblate може одночасно обробляти обидва типи даних і споживатиметься лише подією *push*).

### 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 проекту 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* або рецензувань *Gerrit*. Ви можете активувати їх вибором *GitHub*, *GitLab* або *Gerrit* 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 <sup>1</sup>	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 <sup>1</sup>	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:

☒ **Require pull request reviews before merging**  
When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.  


Required approving reviews: 1 ▾


☐ **Dismiss stale pull request approvals when new commits are pushed**  
New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

☐ **Require review from Code Owners**  
Require an approved review in pull requests including files with a designated code owner.

☒ **Restrict who can dismiss pull request reviews**  
Specify people or teams allowed to dismiss pull request reviews.  

**People and teams that can dismiss reviews.**

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

 **weblate**  
Weblate push user ×

---

<sup>1</sup> Can be empty in case *Сховище з джерелами* supports pushing.

### 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 проекту за допомогою програмного інтерфейсу.

**Дивись також:**

[API](#)

### 2.9.5 «Лінійні» внески

Weblate групує внески від одного автора так, щоб остаточний внесок був якомога більшим. Це значно зменшує кількість внесків, але призводить до того, що у вас може виникнути потреба явним чином ініціювати внески, якщо ви хочете підтримувати синхронізацію зі сховищем системи керування версіями, наприклад, для злиття (таке злиття типово увімкнено для групи керівників проекту, 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 Налаштування 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:

Contribution to this translation requires you to agree with a contributor agreement. [View contributor agreement](#)

Language	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Czech <small>GPL-3.0</small>	✓					
Hebrew <small>GPL-3.0</small>	✓					
Hungarian <small>GPL-3.0</small>	81%	4	5			
English <small>GPL-3.0</small>	✓					

[Start new translation](#)

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Введений текст буде форматовано за абзацами. Може бути включено зовнішні посилання. Не можна використовувати розмітку HTML.

## 2.10.3 Ліцензування для користувачів

Будь-який користувач може ознайомитися із усіма ліцензійними угодами щодо перекладу усіх відкритих проєктів екземпляра Weblate за допомогою сторінки профілю:

[Your profile](#)

[Languages](#) [Preferences](#) [Notifications](#) [Account](#) [Avatar](#) **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 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)



## 2.11 Процес перекладу

### 2.11.1 Голосування за пропозицію

Типово, надсилати пропозиції може кожен, а приймати пропозиції можуть зареєстровані користувачі. Можна скористатися голосуванням за пропозиції, коли рядок використовується, лише якщо за нього проголосувала більшість зареєстрованих користувачів. Для цього у налаштуваннях *складника* слід увімкнути *Голосування за пропозиції* і встановити порогове значення *Автоматичне прийняття пропозиції* для визначення порогового значення для прийнятих пропозицій (до голосів буде включено голос користувача, який вніс пропозицію, якщо він може голосувати).

---

**Примітка:** Щойно буде налаштовано автоматичне прийняття, звичайні користувачі втратять право безпосередньо зберігати переклади або приймати пропозиції. Це можна перевизначити за допомогою прав доступу *Може перевизначати стан пропозиції* (див. *Керування доступом*).

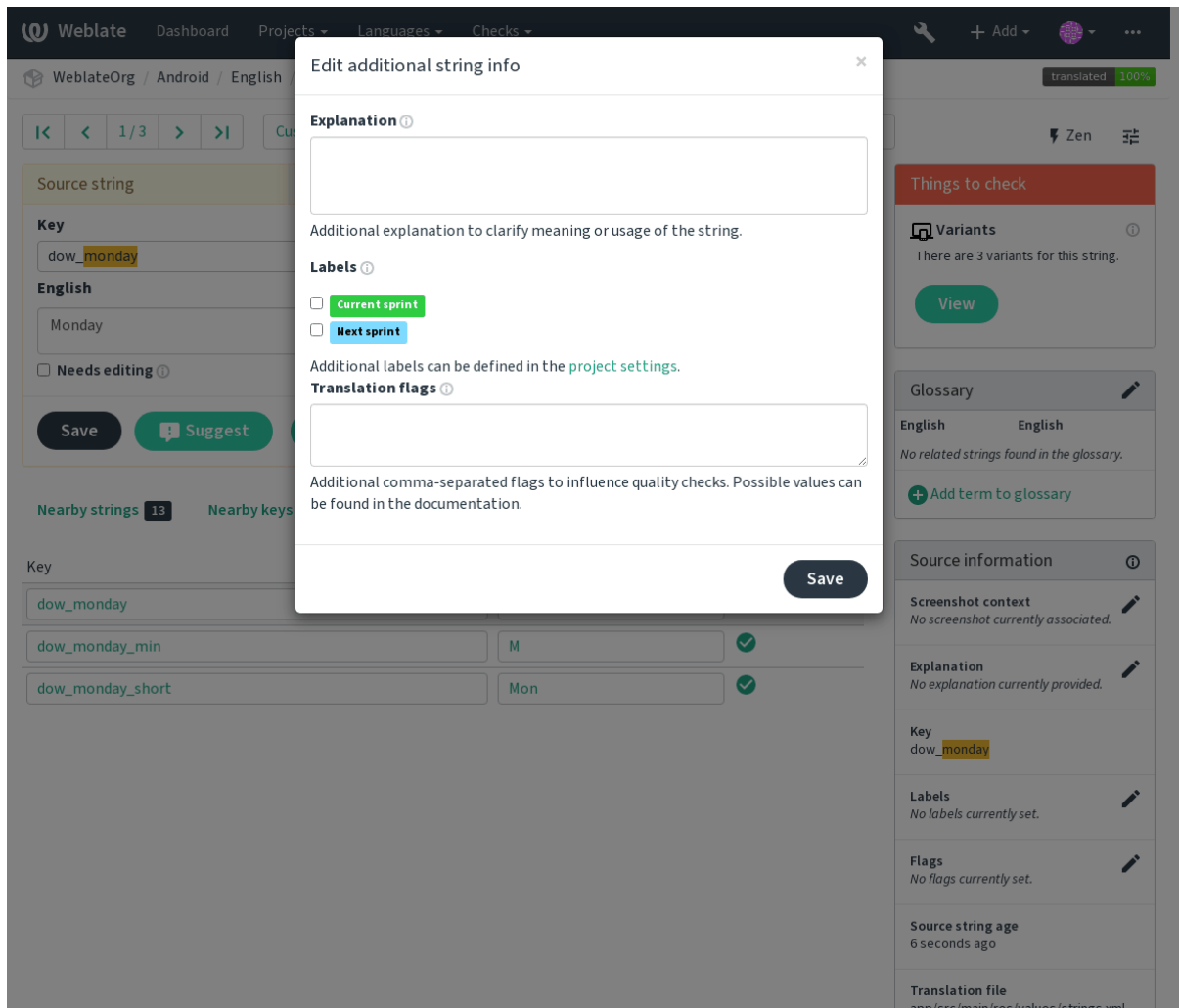
---

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 і голосують за переклади із 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 у розділі *Reviewing strings*:



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 ▾

WeblateOrg / Django / Czech / Translate

Translated 96%

<

>

11 / 26

All strings ▾

Position and priority ▾

Zen

Translation

Explanation

Help text for automatic translation tool

English

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

Czech

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

Needs editing

Save Suggest Skip

Nearby strings 26

Comments

Machinery

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	✓
	<b>Singular</b> %(count)s word <b>Plural</b> %(count)s words	<b>One</b> %(count)s slovo <b>Few</b> %(count)s slova <b>Other</b> %(count)s slov	✓
	Other components	Další součásti	✓
	Translation file	Soubor s překladem	✓
	Download	Stáhnout	✓
	Browse all translation changes	Procházet všechny změny v překladu.	✓ ⚠
	Automatic translation takes existing translations in this project and applies them to the current component. It can be used to push translations to a different branch, to fix inconsistent translations or to translate a new component using translation memory.	Automatický překlad použije stávající překlady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí překladové paměti.	✓
	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.	✓
	You can add new translation string here, it will automatically appear in all translations.	Zde můžete přidat nový řetězec k překladu, automaticky se objeví ve všech jazycích.	✓
	The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.	Nahraný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nepamenejte to povolit.	✓
	The uploaded file will be merged with the current translation.	Nahraný 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	Kolik to je?	✓ ⚠
	The string uses three dots (...) instead of an ellipsis character (...)		📄

Glossary

English

machine translation

project

Czech

strojový překlad

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

Source string age

2 seconds ago

Translation file

weblate/locale/cs/LC\_MESSAGES/djan go.po, string 11

## Встановлення пріоритетності рядків

Нове в версії 2.0.

Пріоритетність рядків може бути змінено так, щоб рядки з вищою пріоритетністю було перекладено раніше, за допомогою прапорця пріоритет

---

**Підказка:** Цим можна скористатися для логічного упорядкування перекладу.

---

**Дивись також:**

[Перевірки якості](#)

## Прапорці перекладу

Нове в версії 2.4.

Змінено в версії 3.3: Раніше ця можливість мала назву *Прапорці перевірок якості*. У поточній версії ця можливість налаштовує не лише перевірки.

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.

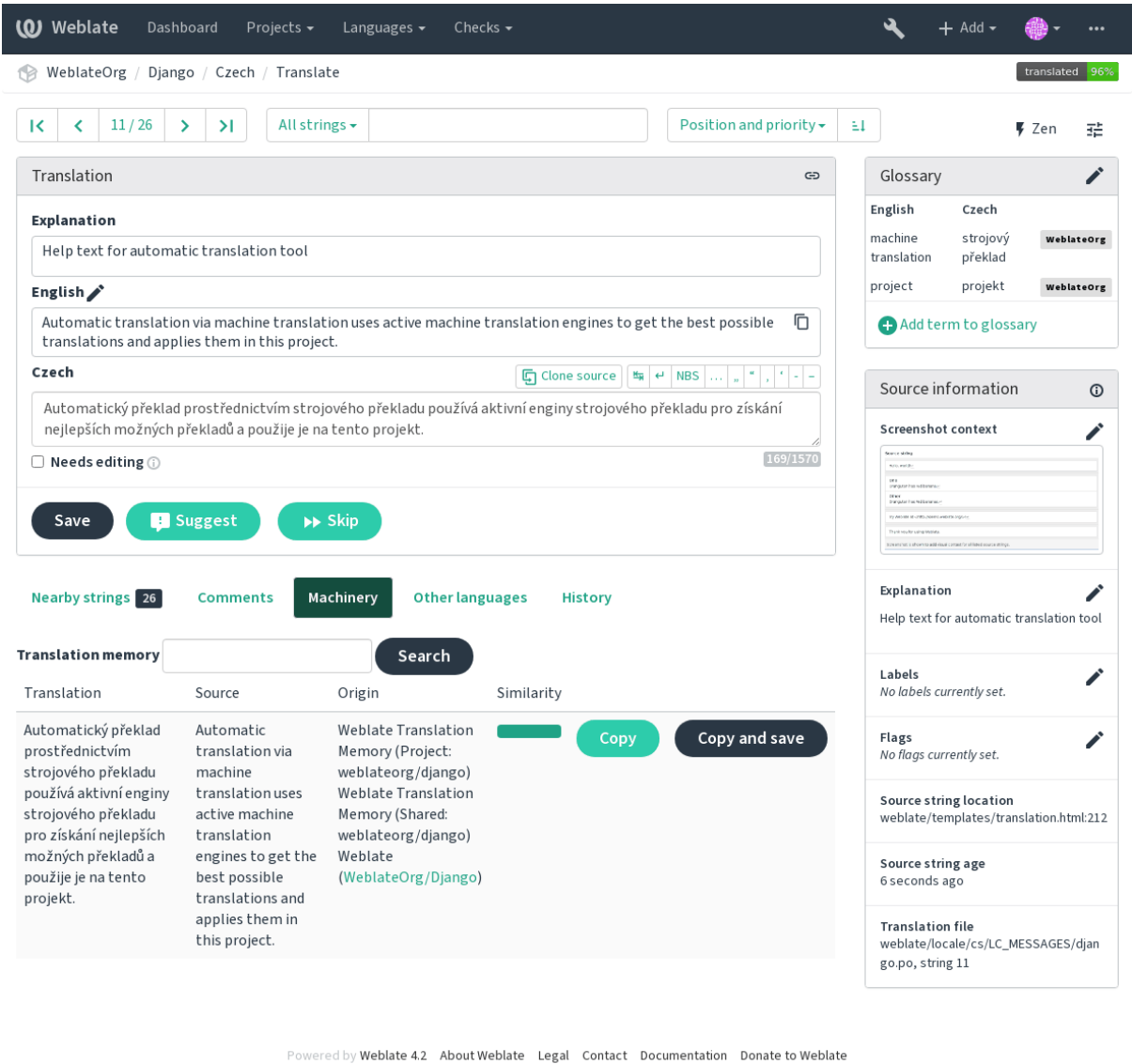
Скористайтесь поясненням для того, що зробити зрозумілішою область або використання перекладу. Ви можете скористатися кодом Markdown для включення посилань та визначення розмітки.

## 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 *Reviewing strings*, знімки вікон маю окремий інтерфейс керування за допомогою меню *Інструменти*. 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:

Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

+

Add ▾

WeblateOrg

/

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></div>testuser</div>

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

Delete

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to Weblate

2.11. Процес перекладу

259

## 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 [Налаштовування складників \(Прапорці перекладу\)](#). Для деяких форматів файлів можна вказати прапорці безпосередньо у записі формату (див. [Підтримувані формати файлів](#)).

Прапорці слід відокремлювати комами, а параметри — двокрапками. Ви можете скористатися лапками для включення до рядка пробілів або спеціальних символів. Приклад:

```
placeholders:"special:value":"other value", regex:.*
```

Ось список прапорців, які приймаються у поточній версії:

**rst-text** Вважати текст документом RST, 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:НАЗВА** Визначає сімейство шрифтів для перевірок обробки, див. [Керування шрифтами](#).

**font-weight:ВАГА** Визначити вагу шрифту для перевірок обробки, див. [Керування шрифтами](#).

**font-size:РОЗМІР** Визначити розмір шрифту для перевірок обробки, див. [Керування шрифтами](#).

**font-spacing:ІНТЕРВАЛ** Визначити інтервал 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 замінників 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, java-format** Обробляти усі рядки як рядки форматування, впливає на [Форматовані рядки](#), [Форматовані рядки](#), [Форматовані рядки](#), [Форматовані рядки](#), [Форматовані рядки](#), [Форматовані рядки](#), [Форматовані рядки](#), [Форматовані рядки](#), [Форматовані рядки](#), [Форматовані рядки](#), [Переклад не змінено](#).

**strict-same** Наказати перевірці «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-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. Вибір можна здійснити за допомогою засобу керування шрифтами Weblate у розділі *Шрифти* меню *Керування* вашого проєкту перекладу.

Можна вивантажувати шрифти TrueType або OpenType, налаштовувати групи шрифтів і використовувати ці групи у перевірці.

Групи шрифтів надають вам змогу визначити різні шрифти для різних мов, що типово знадобиться для мов, які записують нелатинськими символами:

W Weblate

Dashboard


Projects ▾

Languages ▾

Checks ▾

⚙

+ Add ▾



...

WebateOrg / 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 4.2 [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 4.2
 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 4.2
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

Ви можете завантажувати до Weblate декілька шрифтів:

Font family

Font style

Droid Sans Fallback Regular Edit

Source Sans Pro Bold Edit

Add font

Font file

Choose File No file chosen

OpenType and TrueType fonts are supported.

Upload

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to Weblate

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
```

(continues on next page)

(continued from previous page)

```
# (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/>.
#
"""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 проекту](#).

### 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, Apertium website, Apertium APy documentation](#)

### 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.google.v3.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"}

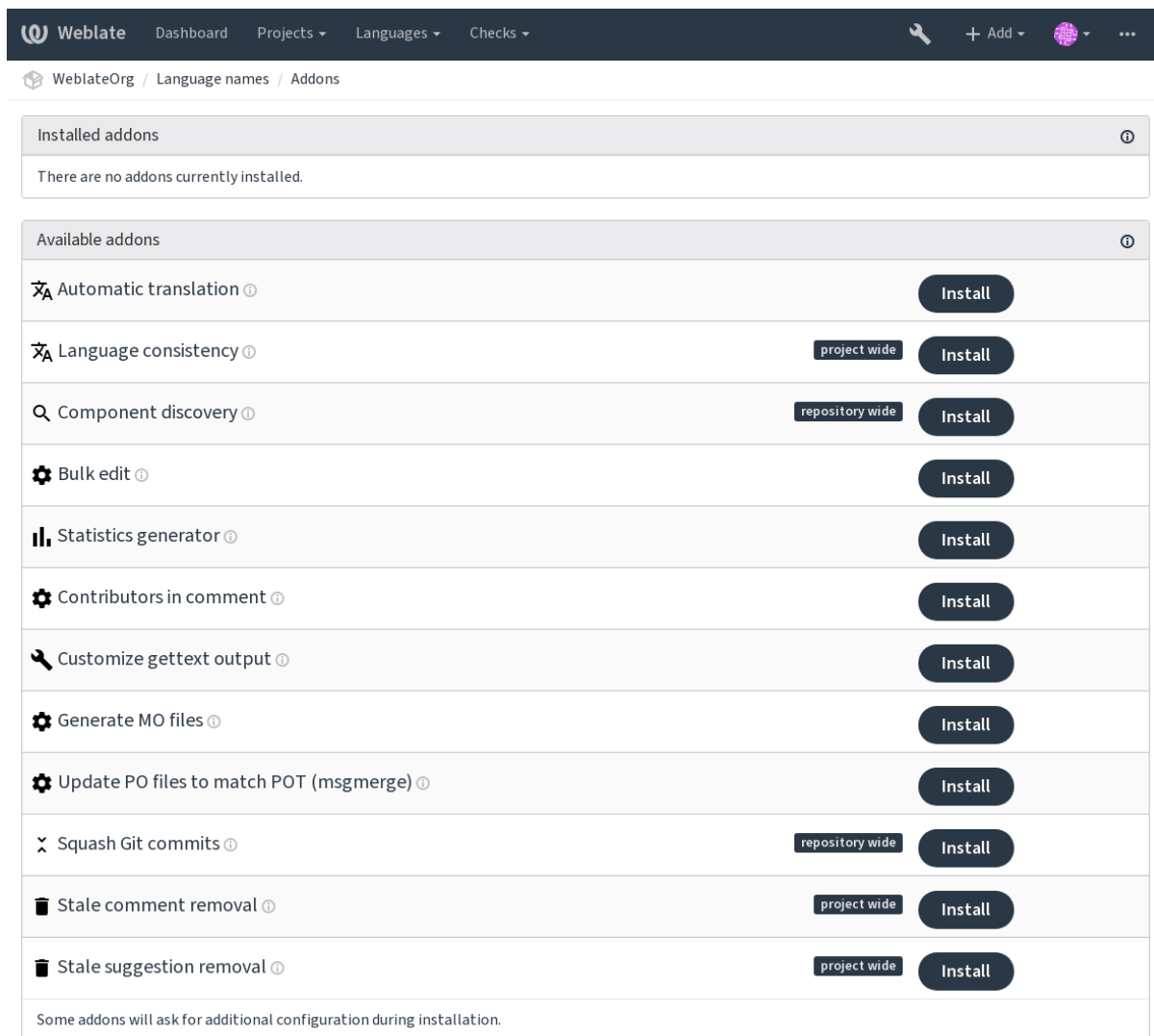
    def download_translations(self, source, language, text, unit, user, search):
        """Return tuple with translations."""
        for t in dictionary.translate(text):
            yield {"text": t, "quality": 100, "service": self.name, "source": text}
```

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 4.2 [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

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. [Переклад HTML і JavaScript за допомогою CDN Weblate](#), щоб дізнатися більше.

#### Дивись також:

*Налаштування додатка CDN Weblate, Переклад HTML і JavaScript за допомогою CDN Weblate, Вибудування рядків для CDN Weblate, Локалізація HTML за допомогою CDN Weblate*

### Очищення файлів перекладу

Оновіть всі файли перекладу, щоб вони відповідали одномовному основному файлу. Для більшості форматів файлів це означає видалення застарілих перекладацьких ключів, які вже відсутні у основному файлі.

### Мовна однорідність

Гарантує, що всі складники в межах одного проекту 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 проекту.

---

**Підказка:** Автоматичний переклад нових доданих рядків за допомогою [Автоматичний переклад](#).

---

### Виявлення складників

Автоматично додає або вилучає складники до проекту на основі змін файла у системі керування версіями.

Вмикається під час кожного оновлення у системі керування версіями. Загалом, подібне до команди керування `import_project`. Один зі способів стежити за декількома складниками перекладу у одній системі керування версіями.

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:

Powered by Weblate 4.2 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)



**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.

**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 джерельні 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 про переклад.

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 РО, щоб включити імена перекладачів та роки перекладу.

Заголовок файла РО 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 mo-файли

Автоматично створює mo-файл для кожного зміненого ро-файла.

## Оновити файли ро, порівнявши із pot (msgmerge)

Оновлює всі файли ро, порівнявши з відповідним файлом pot за допомогою msgmerge. Це спрацює щоразу, коли нові зміни витягуються із основного сховища даних.

## 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.

---

## 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 `BaseAddon`, визначити метадані додатка і реалізувати зворотний виклик, який виконуватиме обробку.

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.
#
# 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
```

## 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 проекту, який використовується для побудови адреси.

**WL\_COMPONENT\_NAME**

Нове в версії 3.9.

Назва складника.

**WL\_PROJECT\_NAME**

Нове в версії 3.9.

Назва проекту.

**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

Скористайтесь 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 *Машинний переклад* 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:

*Weblate Translation Memory Schema*



Translation memory status

Number of your entries	0	Download as JSON	Download as TMX	Delete
Total number of entries	0			

Import translation memory

File

Choose File No file chosen

You can upload a TMX or JSON file.

Upload

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to 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](https://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 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)*

## 2.16.12 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*

### 2.16.13 COMMENT\_CLEANUP\_DAYS

Нове в версії 3.6.

Вилучати коментарі за вказану кількість днів. Типове значення — `None`, тобто не вилучати коментарі.

### 2.16.14 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`*

### 2.16.15 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*

### 2.16.16 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*

### 2.16.17 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 покладаетесь на внутрішні механізми керування Weblate.

**Дивись також:**

*Керування доступом на рівні проєкту, Керування доступом, Керування доступом*

### 2.16.18 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*

### 2.16.19 DEFAULT\_ADD\_MESSAGE, DEFAULT\_COMMIT\_MESSAGE, DEFAULT\_MERGE\_MESSAGE

### DEFAULT\_ADDON\_MESSAGE, DEFAULT\_DELETE\_MESSAGE,

Типові повідомлення про внесок для різних дій. Докладніше про це у розділі *Налаштування складників*.

**Дивись також:**

*Розмітка шаблонів, Налаштовування складників, Повідомлення щодо внеску, додавання, вилучення, об'єднання та повідомлення додатків*

## 2.16.20 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*

## 2.16.21 DEFAULT\_COMMITTER\_EMAIL

Нове в версії 2.4.

Адреса електронної пошти автора внеску для створених складників перекладу. Типове значення — `noreply@weblate.org`.

**Дивись також:**

*DEFAULT\_COMMITTER\_NAME*, *Налаштування складників*, *Ел. пошта подавача*

## 2.16.22 DEFAULT\_COMMITTER\_NAME

Нове в версії 2.4.

Ім'я автора внеску для створених складників перекладу. Типове значення — `Weblate`.

**Дивись також:**

*DEFAULT\_COMMITTER\_EMAIL*, *Налаштування складників*, *Ім'я подавача*

## 2.16.23 DEFAULT\_MERGE\_STYLE

Нове в версії 3.4.

Стиль злиття для усіх нових складників.

- *rebase* — типовий
- *merge*

**Дивись також:**

*Налаштування складників*, *Стиль злиття*

## 2.16.24 DEFAULT\_TRANSLATION\_PROPAGATION

Нове в версії 2.5.

Типовий параметр для надсилання перекладу, типове значення — True.

**Дивись також:**

*Налаштування складників, Дозволити поширення перекладу*

## 2.16.25 DEFAULT\_PULL\_MESSAGE

Заголовок для нових запитів щодо об'єднання, типове значення — 'Update from Weblate'.

## 2.16.26 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*

## 2.16.27 ENABLE\_HOOKS

Чи слід вмикати доступу до анонімних віддалених скриптів обробки.

**Дивись також:**

*Обробники сповіщень*

## 2.16.28 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 сайта*

## 2.16.29 ENABLE\_SHARING

Увімкнути або вимкнути меню *Поділитися*, щоб користувачі могли оприлюднювати поступ перекладу у соціальних мережах.



### 2.16.30 GITLAB\_USERNAME

Ім'я користувача GitLab, яке буде використано для надсилання запитів щодо об'єднання для оновлень перекладів.

**Дивись також:**

*GitLab, Налаштовування Lab*

### 2.16.31 GITHUB\_USERNAME

Ім'я користувача GitHub, яке буде використано для надсилання запитів щодо об'єднання для оновлень перекладів.

**Дивись також:**

*GitHub, Налаштовування концентратора*

### 2.16.32 GOOGLE\_ANALYTICS\_ID

Ідентифікатор Google Analytics для вмикання спостереження за Weblate за допомогою Google Analytics.

### 2.16.33 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.

---

### 2.16.34 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*

### 2.16.35 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`*

### 2.16.36 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`*

### 2.16.37 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/"
```

### 2.16.38 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",
    ),
]
```

### 2.16.39 LICENSE\_FILTER

Необов'язкове додавання до показаних умов ліцензування.

---

**Примітка:** У цьому фільтрі використовуються скорочені назви ліцензій.

---

Приклад:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

### 2.16.40 LICENSE\_REQUIRED

Визначає, чи є обов'язковим атрибут ліцензування у записі *Налаштовування складників*.

---

**Примітка:** Типово вимкнено.

---

### 2.16.41 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`.

---

### 2.16.42 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*

### 2.16.43 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.

### 2.16.44 LOGIN\_REQUIRED\_URLS\_EXCEPTIONS

Список виключень для `LOGIN_REQUIRED_URLS`. Якщо не вказано, користувачам буде відкрито доступу до сторінки входу до системи.

Деякі виключення, які вам варто включити:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r'/accounts/(.*)$', # Required for login
    r'/static/(.*)$',   # 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
)
```

### 2.16.45 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`

## 2.16.46 MATOMO\_URL

Повна адреса (разом із кінцевою похилою рискою) 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*

## 2.16.47 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 ref: *machine*, 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, Машинний переклад*

### 2.16.48 MT\_APERTIUM\_APY

Адреса сервера Apertium-APY, <https://wiki.apertium.org/wiki/Apertium-apy>

**Дивись також:**

*Apertium, Машинний переклад, Машинний переклад*

### 2.16.49 MT\_AWS\_ACCESS\_KEY\_ID

Ідентифікатор ключа доступу для Amazon Translate.

**Дивись також:**

*AWS, Машинний переклад, Машинний переклад*

### 2.16.50 MT\_AWS\_SECRET\_ACCESS\_KEY

Реєстраційний ключ до програмного інтерфейсу Amazon Translate.

**Дивись також:**

*AWS, Машинний переклад, Машинний переклад*

### 2.16.51 MT\_AWS\_REGION

Назва регіону, якою слід скористатися у Amazon Translate.

**Дивись також:**

*AWS, Машинний переклад, Машинний переклад*

### 2.16.52 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, Машинний переклад*

### 2.16.53 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, Машинний переклад*

## 2.16.54 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, Машинний переклад*

## 2.16.55 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, Машинний переклад*

## 2.16.56 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, Машинний переклад*

## 2.16.57 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.

## 2.16.58 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

### 2.16.59 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)*

### 2.16.60 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`.

### 2.16.61 MT\_MICROSOFT\_COGNITIVE\_KEY

Ключ клієнта до програмного інтерфейсу когнітивних служб перекладу Microsoft.

**Дивись також:**

*Когнітивні служби перекладу Microsoft, Машинний переклад, Машинний переклад, Когнітивні служби — програмний інтерфейс для перекладу текстів, Портал Microsoft Azure*

### 2.16.62 MT\_MICROSOFT\_REGION

Префікс регіону, як його визначено у «[Multi service subscription](#)».

### 2.16.63 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.

### 2.16.64 MT\_MODERNMT\_KEY

Ключ до програмного інтерфейсу для рушія комп'ютерного перекладу ModernMT.

**Дивись також:**

*ModernMT MT\_MODERNMT\_URL*



### 2.16.65 MT\_MODERNMT\_URL

Адреса ModernMT. Типовою для «хмарної» служби є адреса <https://api.modernmt.com/>.

**Дивись також:**

*ModernMT MT\_MODERNMT\_KEY*

### 2.16.66 MT\_MYMOMETRY\_EMAIL

Ідентифікаційна адреса електронної пошти MyMemory. Приймає 1000 запитів на день.

**Дивись також:**

*MyMemory, Машинний переклад, Машинний переклад, MyMemory: технічна специфікація програмного інтерфейсу*

### 2.16.67 MT\_MYMOMETRY\_KEY

Ключ доступу MyMemory для приватної пам'яті перекладів. Слід використовувати разом із *MT\_MYMOMETRY\_USER*.

**Дивись також:**

*MyMemory, Машинний переклад, Машинний переклад, MyMemory: генератор ключів до програмного інтерфейсу*

### 2.16.68 MT\_MYMOMETRY\_USER

Ідентифікатор користувача MyMemory для приватної пам'яті перекладів. Слід використовувати разом із *MT\_MYMOMETRY\_KEY*.

**Дивись також:**

*MyMemory, Машинний переклад, Машинний переклад, MyMemory: генератор ключів до програмного інтерфейсу*

### 2.16.69 MT\_NETEASE\_KEY

Програмний ключ до програмного інтерфейсу NetEase Sight. Зареєструватися можна на <https://sight.netease.com/>

**Дивись також:**

*Програмний інтерфейс комп'ютерного перекладу NetEase Sight, Машинний переклад, Машинний переклад*

### 2.16.70 MT\_NETEASE\_SECRET

Програмний пароль до програмного інтерфейсу NetEase Sight. Зареєструватися можна на <https://sight.netease.com/>

**Дивись також:**

*Програмний інтерфейс комп'ютерного перекладу NetEase Sight, Машинний переклад, Машинний переклад*

### 2.16.71 MT\_TMSERVER

Адреса, за якою працює tmserver.

**Дивись також:**

*tmserver, Машинний переклад, Машинний переклад, tmserver*

### 2.16.72 MT\_YANDEX\_KEY

Ключ до програмного інтерфейсу Yandex Translate. Зареєструватися можна на <https://tech.yandex.com/translate/>

**Дивись також:**

*Переклад Yandex, Машинний переклад, Машинний переклад*

### 2.16.73 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, Машинний переклад*

### 2.16.74 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, Машинний переклад*

### 2.16.75 MT\_SAP\_BASE\_URL

Адреса програмного інтерфейсу до служби вузла перекладу SAP.

**Дивись також:**

*Концентратор перекладів SAP, Машинний переклад, Машинний переклад*

### 2.16.76 MT\_SAP\_SANDBOX\_APIKEY

Ключ до програмного інтерфейсу для використання програмного інтерфейсу пісочниці

**Дивись також:**

*Концентратор перекладів SAP, Машинний переклад, Машинний переклад*

### 2.16.77 MT\_SAP\_USERNAME

Ваше ім'я користувача SAP

**Дивись також:**

*Концентратор перекладів SAP, Машинний переклад, Машинний переклад*

### 2.16.78 MT\_SAP\_PASSWORD

Ваш пароль SAP

**Дивись також:**

*Концентратор перекладів SAP, Машинний переклад, Машинний переклад*

### 2.16.79 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, Машинний переклад*

### 2.16.80 NEARBY\_MESSAGES

Кількість показаних рядків, які розташовано поруч із поточним рядком перекладу. Це лише типові значення, користувачі можуть змінювати його за допомогою *Профіль користувача*.

### 2.16.81 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*

### 2.16.82 RATELIMIT\_WINDOW

Нове в версії 3.2.

Тривалість періоду прийняття розпізнавання після застосування обмеження частоти.

Проміжок часу у секундах, типові значення — 300 (5 хвилин).

**Дивись також:**

*Обмеження частоти, RATELIMIT\_ATTEMPTS, RATELIMIT\_LOCKOUT*

### 2.16.83 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`

### 2.16.84 REGISTRATION\_ALLOW\_BACKENDS

Нове в версії 4.1.

Список модулів розпізнавання, з яких можна реєструватися у випадку, якщо реєстрацію вимкнено за допомогою `REGISTRATION_OPEN`.

Приклад:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

---

**Підказка:** Назви модулів відповідають назвам, використаним у адресі для розпізнавання.

---

**Дивись також:**

`REGISTRATION_OPEN`

### 2.16.85 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.

### 2.16.86 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$'
```

## 2.16.87 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`

## 2.16.88 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:**

*Translation component alerts*

## 2.16.89 SENTRY\_DSN

Нове в версії 3.9.

Sentry DSN, яким слід скористатися для *Збирання звітів щодо помилок*.

**Дивись також:**

Інтеграція Django із Sentry

## 2.16.90 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.

## 2.16.91 SITE\_DOMAIN

Налаштовує домен сайту. Домен потрібен для створення правильних абсолютних посилань у багатьох областях застосування (наприклад, для повідомлень активації, сповіщень та стрічок RSS).

Якщо Weblate запущено на нестандартному порту, допишіть його до вказаної адреси.

**Приклади:**

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

**Примітка:** Цей параметр має містити лише назву домену. Для налаштованого протоколу (увімкнутого і примусово встановленого HTTPS) скористайтеся `ENABLE_HTTPS`, а для зміни адреси скористайтеся `URL_PREFIX`.

**Підказка:** У контейнері Docker домен сайта можна налаштувати за допомогою `WEBLATE_ALLOWED_HOSTS`.

**Дивись також:**

faq-site, *Налаштовування дозволених вузлів*, *Належне налаштування HTTPS* `ENABLE_HTTPS`, changesite, The “sites” framework

## 2.16.92 SITE\_TITLE

Назва сайта, яку буде використано для сайта і надсилання повідомлень електронною поштою.

## 2.16.93 SPECIAL\_CHARS

Додаткові символи, які слід включити на візуальну клавіатуру, *Візуальна клавіатура*.

Типове значення:

```
SPECIAL_CHARS = ('\t', '\n', '...')
```

## 2.16.94 SINGLE\_PROJECT

Нове в версії 3.8.

Переспрямовує користувачів безпосередньо до проекту або складника замість показу панелі приладів. Ви можете встановити значення `True` — у цьому випадку це працюватиме, лише якщо на Weblate є лише один проект. У протилежному випадку встановіть ідентифікатор проекту, і він безумовно переспрямовуватиме до цього проекту.

Змінено в версії 3.11: У поточній версії цього параметра також приймається ідентифікатор проекту для примусового показу лише цього проекту.

Приклад:

```
SINGLE_PROJECT = "test"
```

## 2.16.95 STATUS\_URL

Адреса, на яку екземпляр Weblate повідомляє про свій стан.

## 2.16.96 SUGGESTION\_CLEANUP\_DAYS

Нове в версії 3.2.1.

Автоматично вилучає пропозицій після вказаної кількості днів. Типовим є значення `None`, тобто пропозиції не вилучаються.

## 2.16.97 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.

---

## 2.16.98 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*

## 2.16.99 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
```

## 2.16.100 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 збергти 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.LanguageConsistencyAddon",
    "weblate.addons.discovery.DiscoveryAddon",
    "weblate.addons.flags.SourceEditAddon",
    "weblate.addons.flags.TargetEditAddon",
    "weblate.addons.flags.SameEditAddon",
    "weblate.addons.flags.BulkEditAddon",
    "weblate.addons.generate.GenerateFileAddon",
    "weblate.addons.json.JSONCustomizeAddon",
    "weblate.addons.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",
)
```

**Дивись також:**

*Доповнення*



### 2.16.101 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*

### 2.16.102 WEBLATE\_FORMATS

Нове в версії 3.0.

Список форматів файлів, якими ви можете скористатися.

---

**Примітка:** У типовому списку вже є загальновживані формати.

---

**Дивись також:**

*Підтримувані формати файлів*

### 2.16.103 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.
#
```

(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/>.
#

import os
import platform
from logging.handlers import SysLogHandler

#
# Django settings for Weblate project.
#

DEBUG = True

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # 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"

```

(continues on next page)

(continued from previous page)

```

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

LANGUAGES = (
    ("ar", "العربية"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@latin", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("da", "Dansk"),
    ("de", "Deutsch"),
    ("en", "English"),
    ("el", "Ελληνικά"),
    ("en-gb", "English (United Kingdom)"),
    ("es", "Español"),
    ("fi", "Suomi"),
    ("fr", "Français"),
    ("gl", "Galego"),
    ("he", "עברית"),
    ("hu", "Magyar"),
    ("hr", "Hrvatski"),
    ("id", "Indonesia"),
    ("is", "Íslenska"),
    ("it", "Italiano"),
    ("ja", "日本語"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("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

```

(continues on next page)

(continued from previous page)

```

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(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,
        },
    },
]

```

(continues on next page)

(continued from previous page)

```

]

# GitHub username for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None

# GitLab username for sending merge requests.
# Please see the documentation for more details.
GITLAB_USERNAME = 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_FACEBOOK_API_VERSION = "3.1"

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",

```

(continues on next page)

(continued from previous page)

```

"weblate.accounts.pipeline.verify_username",
"social_core.pipeline.user.create_user",
"social_core.pipeline.social_auth.associate_user",
"social_core.pipeline.social_auth.load_extra_data",
"weblate.accounts.pipeline.cleanup_next",
"weblate.accounts.pipeline.user_full_name",
"weblate.accounts.pipeline.store_email",
"weblate.accounts.pipeline.notify_connect",
"weblate.accounts.pipeline.password_reset",
)
SOCIAL_AUTH_DISCONNECT_PIPELINE = (
    "social_core.pipeline.disconnect.allowed_to_disconnect",
    "social_core.pipeline.disconnect.get_entries",
    "social_core.pipeline.disconnect.revoke_tokens",
    "weblate.accounts.pipeline.cycle_session",
    "weblate.accounts.pipeline.adjust_primary_mail",
    "weblate.accounts.pipeline.notify_disconnect",
    "social_core.pipeline.disconnect.disconnect",
    "weblate.accounts.pipeline.cleanup_next",
)

# Custom authentication strategy
SOCIAL_AUTH_STRATEGY = "weblate.accounts.strategy.WeblateStrategy"

# Raise exceptions so that we can handle them later
SOCIAL_AUTH_RAISE_EXCEPTIONS = True

SOCIAL_AUTH_EMAIL_VALIDATION_FUNCTION = "weblate.accounts.pipeline.send_validation"
SOCIAL_AUTH_EMAIL_VALIDATION_URL = "{0}/accounts/email-sent/".format(URL_PREFIX)
SOCIAL_AUTH_LOGIN_ERROR_URL = "{0}/accounts/login/".format(URL_PREFIX)
SOCIAL_AUTH_EMAIL_FORM_URL = "{0}/accounts/email/".format(URL_PREFIX)
SOCIAL_AUTH_NEW_ASSOCIATION_REDIRECT_URL = "{0}/accounts/profile/#account".format(
    URL_PREFIX
)

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")
    #     }
    # },
]

```

(continues on next page)

(continued from previous page)

```

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
    "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.langdata",
    "weblate.memory",
    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "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",

```

(continues on next page)

(continued from previous page)

```

    "compressor",
    "rest_framework",
    "rest_framework.auth_token",
    "django_filters",
]

# Custom exception reporter to include some details
DEFAULT_EXCEPTION_REPORTER_FILTER = "weblate.trans.debug.
↳ WeblateExceptionReporterFilter"

# Default logging of Weblate messages
# - to syslog in production (if available)
# - otherwise to console
# - 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 IOError:
        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[%s] %s %s" % (process)d]: %(levelname)s %(message)s"},
        "simple": {"format": "%(levelname)s %(message)s"},
        "logfile": {"format": "%(asctime)s %(levelname)s %(message)s"},
        "django.server": {
            "(): "django.utils.log.ServerFormatter",
            "format": "[%s] %s" % (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",

```

(continues on next page)



(continued from previous page)

```

        "class": "logging.StreamHandler",
        "formatter": "simple",
    },
    "django.server": {
        "level": "INFO",
        "class": "logging.StreamHandler",
        "formatter": "django.server",
    },
    "syslog": {
        "level": "DEBUG",
        "class": "logging.handlers.SysLogHandler",
        "formatter": "syslog",
        "address": "/dev/log",
        "facility": SysLogHandler.LOG_LOCAL2,
    },
    # 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},
    },
}

# 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",

```

(continues on next page)

(continued from previous page)

```

# "weblate.machinery.google.GoogleTranslation",
# "weblate.machinery.google.v3.GoogleV3Translation",
# "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
# "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
# "weblate.machinery.modernmt.ModernMTTranslation",
# "weblate.machinery.mymemory.MyMemoryTranslation",
# "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

```

(continues on next page)

(continued from previous page)

```

# tmserver URL
MT_TMSERVER = None

# SAP Translation Hub
MT_SAP_BASE_URL = None
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 referrrer 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 = 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 * 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

```

(continues on next page)

(continued from previous page)

```

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = "{0}/accounts/login/".format(URL_PREFIX)

# URL of logout
LOGOUT_URL = "{0}/accounts/logout/".format(URL_PREFIX)

# Default location for login
LOGIN_REDIRECT_URL = "{0}/".format(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 = "[{0}] ".format(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",

```

(continues on next page)

(continued from previous page)

```

# "weblate.checks.format.CFormatCheck",
# "weblate.checks.format.PperlFormatCheck",
# "weblate.checks.format.JavaScriptFormatCheck",
# "weblate.checks.format.CSharpFormatCheck",
# "weblate.checks.format.JavaFormatCheck",
# "weblate.checks.format.JavaMessageFormatCheck",
# "weblate.checks.format.PercentPlaceholdersCheck",
# "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",

```

(continues on next page)

(continued from previous page)

```

# "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.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 = [SITE_DOMAIN]

# 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"
    ]
}

```

(continues on next page)

(continued from previous page)

```

    ],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework.authentication.TokenAuthentication",
        "weblate.api.authentication.BearerAuthentication",
        "rest_framework.authentication.SessionAuthentication",
    ),
    "DEFAULT_THROTTLE_CLASSES": (
        "weblate.api.throttling.UserRateThrottle",
        "weblate.api.throttling.AnonRateThrottle",
    ),
    "DEFAULT_THROTTLE_RATES": {"anon": "100/day", "user": "5000/hour"},
    "DEFAULT_PAGINATION_CLASS": ("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

```

(continues on next page)

(continued from previous page)

```

CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

# Celery settings, it is not recommended to change these
CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_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"},
}

# 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 файлової системи*, 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*, рекомендовано для розробки.

## 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 проєкту.

**--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, Weblate Translation Memory Schema*

**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, Weblate Translation Memory Schema*

### 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 проекту на основі маски файлів.

`<project>` є назвою наявного проекту, 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. [Налаштовування 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 проекту 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 проекти.

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 with internationalization*. 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 проєкт 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.l10n.cz/Jazyky>`.

### 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 скрипти 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 сайту, проєкту, складника або мови.

Оголошуйте про мету, контрольні дати, стан або вказуйте цілі для перекладу.

Користувачі отримають сповіщення при появі оголошень у проєктах, за якими вони спостерігають (якщо таке отримання не було вимкнено явним чином).

Це може бути корисним для різних речей — від оголошення мети на рівні сайту до визначення цілей для перекладу.

Оголошення можна розмішувати на будь-якому рівні меню *Управління* за допомогою *Повідомлення оголошення*:

Weblate
 Dashboard Projects Languages Checks

WeblateOrg
 translated 90%

Translations will be used only if they reach 60%.

Components Languages Info Search Glossaries Insights Files Tools **Manage** Share Not watching

Post announcement

**Message**
  
  
 You can use Markdown and mention users by @username.

**Category**
  
 Info (light blue)
   
 Category defines color used for the message.

**Expiry date**
  
 mm/dd/yyyy
   
 The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ **Notify users**
  
 The message is shown for all translations within the project, until its given expiry, or permanently until it is deleted.

Add

Powered by Weblate 4.2 About Weblate Legal Contact Documentation Donate to Weblate

Додавання також можливе за допомогою адміністративного інтерфейсу:

Weblate administration
 WELCOME, WEBLATE TEST RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / LOG 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.

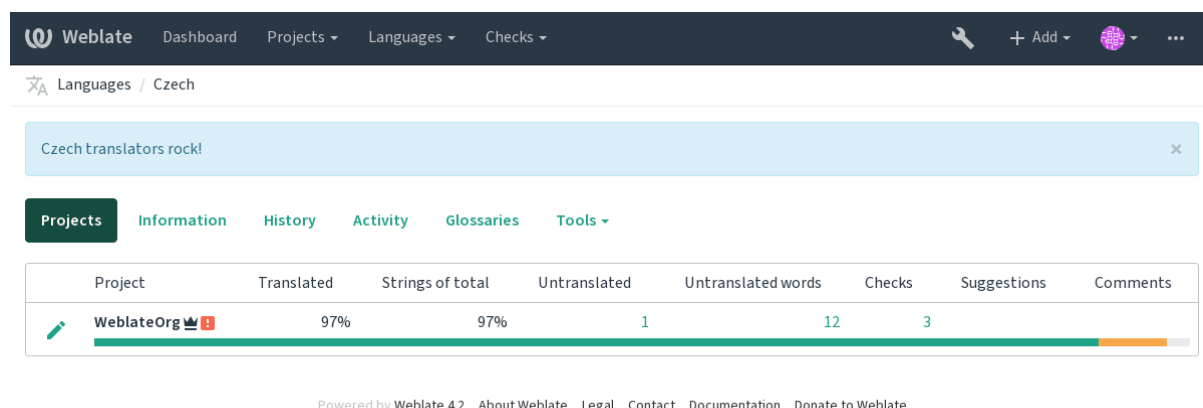
Вказано складник

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:



Project	Translated	Strings of total	Untranslated	Untranslated words	Checks	Suggestions	Comments
WeblateOrg	97%	97%	1	12	3		

## 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 сонові 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](#) / [LOG 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 ?

Filter

WeblateOrg/Django

WeblateOrg/Language names

Choose all ?

Chosen components ?

Remove all

Hold down "Control", or "Command" on a Mac, to select more than one.

AUTOMATIC COMPONENT LIST ASSIGNMENTS		
PROJECT REGULAR EXPRESSION <small>?</small>	COMPONENT REGULAR EXPRESSION <small>?</small>	DELETE?
<input type="text" value="^.*\$"/>	<input type="text" value="^.*\$"/>	<a href="#">✕</a>
<a href="#">+ Add another Automatic component list assignment</a>		

Save and add another Save and continue editing SAVE

## 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

Модуль автоматично обробляє дані Weblate і встановлює адресу експортованого сховища у *компоненті*. Доступ до сховищ можна отримати за частиною `/git/` адреси Weblate, наприклад `https://example.org/git/weblate/master/`:

```
git clone 'https://example.org/git/weblate/master/'
```

Доступ до сховищ є анонімним, якщо не увімкнено *Керування доступом на рівні проєкту*. Це потребує розпізнавання за допомогою ключа програмного інтерфейсу (його можна отримати у вашому *профілі 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 *профілі 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 проекту користувач може вибрати, до якого рахунка слід включати проект, 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:** Юридичні документи для служби Hosted Weblate доступні у цьому сховищі Git: <https://github.com/WeblateOrg/hosted/tree/master/wlhosted/legal/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 login
SOCIAL_AUTH_PIPELINE += (
    'weblate.legal.pipeline.tos_confirm',
)

# Middleware to enforce TOS confirmation of signed in users
MIDDLEWARE += [
    'weblate.legal.middleware.RequireTOSMiddleware',
]
```

2. Запустіть перенесення бази даних, щоб додатково встановити структури бази даних для модуля:



```
weblate migrate
```

- Змініть юридичні документи у теці `weblate/legal/templates/legal/` так, щоб вони відповідали вашій службі.

## Користування

Після встановлення і редагування юридичні документи буде показано у інтерфейсі користувача Weblate.

### 2.21.4 Аватари

Аватари завантажуються і кешуються на боці сервера для зменшення витоку відомостей для сайтів, які їх типово обслуговують. Вбудовану підтримку для отримання аватарів з адрес електронної пошти, які для них налаштовано, можна вимкнути за допомогою `ENABLE_AVATARS`.

У поточній версії Weblate передбачено підтримку таких служб:

- Gravatar

**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](https://akismet.com).

- Встановіть модуль Python `akismet`
- 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.

- 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»:

Webplate

Dashboard

Projects

Languages

Checks

Register

Sign in

About Weblate

Webplate keys

About Weblate

Statistics

Keys

SSH key

SSH key not available.

Commit signing

All commits made with Weblate are signed with the GPG key 658522D589546E92261884D4FC059CF323AA14AC, for which the corresponding public key is found below.

-----BEGIN PGP PUBLIC KEY BLOCK-----  
  
mQGNBF87kzcBDACtJrlJfV9f96vRgtRMovfFo1mwVVOyhxKMivDIOJXsJHYZ05Lh  
IRORX/BUNwn+DDstQTRSoHkcC9UjahBmfrAxlUSdQsKE4aVzyQK0jk/Wrzvqdlpn  
iVESAbL7Gv48VjR5CdMQJ5Z7eKGDf/WZt+3SH6S6viECivWBPCsk6FNvyUTwdpZ0  
vS0xvLdK8GyLH9rdFKNhjRTBKO/7FURz9iukT7YQr8/i44a/SyqUBKvZR2oivjm  
WM2dW6Nmp6IIYZ8tkmAo7bVraJbUysceXukU5tAxtjZaaUdunXEN7o3s6+tyGaD6  
9vrQc/OAjdAcuNKATVW8yNGSbF0lr4wsluvGJpCEfP562iAMxloS6t6kB81bC3HR  
tS4/72DagaS+dkju199gZerSfSerE25VIOszPFTLzeHjJxs1Z7Ula/laByXLIeV  
UGxaRwxX6PZxqueCsleB6n6OsKAEvucUIUA5CMD2jaJSn6dfiZfnlgt2luAjMKA  
aYsH6RUWpOysGqUAEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J  
Ac4EEwEKADgWIQRihSLVIRukiYhNT8BZzzI6oUrAUCXzuTNwibAwULCQHAgYV  
CgklCwIEFgIDAQIeAQIXgAAKCRD8BZzzI6oUrGxjC/9Am+oiMJvUs8tDm1hSeTKG  
vBIP/PTfxdJbSRkl77nWmVShcfLLhcvlKa+TL0SnpdWqzLIT/YcRo6Owd7FIWW2  
yQfgdJVVcgdbzuadzjpZKrhJcfJg5xB1CM03ByPatmD2rLaYK53fQHqKZR72Gyrx  
\_0nT702s6YeWERMcnnoCPNzihAfrJn2aXhdisHL1EO70aWU+D76mAVfiLRzi3Tlw/

Powered by Weblate 4.2

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 вимкнено розпізнавання за паролем, аж доки не буде виконано процедуру скидання пароля облікового запису.

Дивись також:

*Обмеження частоти, Робота за реверсивним проксі-сервером*

## 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:

*Contributing to 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 використовуватимемо `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. Create a simple Django app containing the static files you want to overwrite (see *Створення модуля Python*).

2. Додайте її до `INSTALLED_APPS`:

```

INSTALLED_APPS = (
    # Add your customization as first
    'weblate_customization',

    # Weblate apps are here...
)

```

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.

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',
)

```

(continues on next page)

(continued from previous page)

```

)

# Addons
WEBLATE_ADDONS += (
    'weblate_customization.addons.ExamplePreAddon',
)

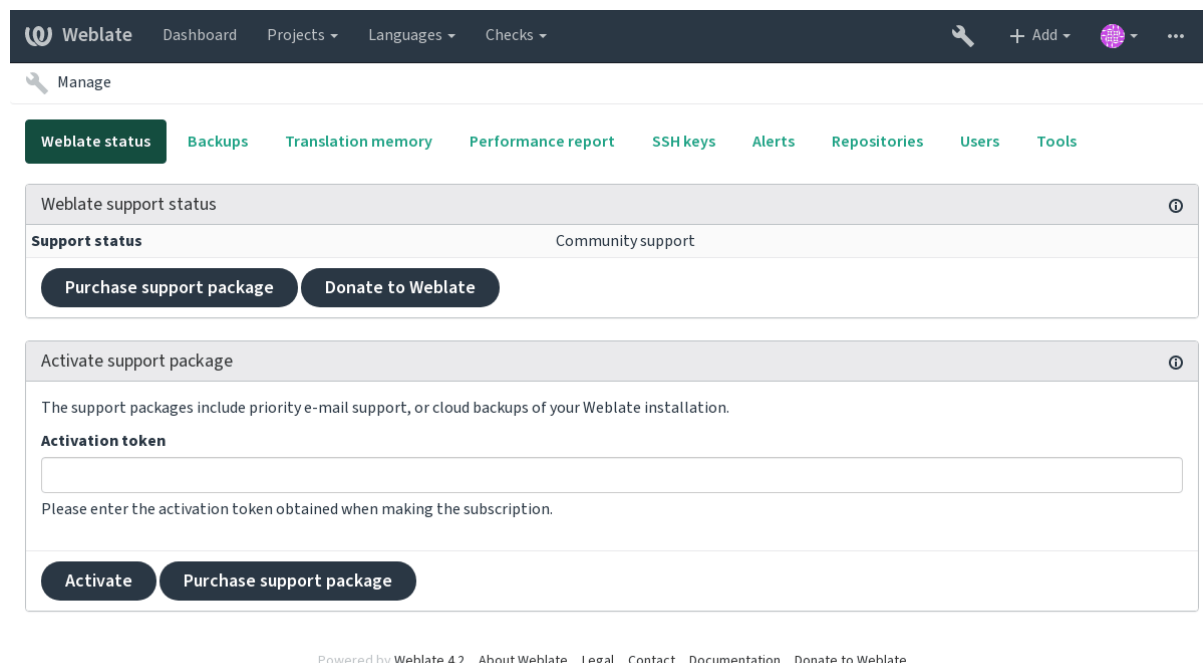
```

**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](#) / [LOG OUT](#)

Site administration

REPORTS		
Weblate support status		
Status of repositories		
SSH keys		
Performance report		
Translation memory		
ACCOUNTS		
Audit logs	<a href="#">+ Add</a>	<a href="#">Change</a>
Profiles	<a href="#">+ Add</a>	<a href="#">Change</a>
Verified emails	<a href="#">+ Add</a>	<a href="#">Change</a>
AUTH TOKEN		
Tokens	<a href="#">+ Add</a>	<a href="#">Change</a>
AUTHENTICATION		
Groups	<a href="#">+ Add</a>	<a href="#">Change</a>
Roles	<a href="#">+ Add</a>	<a href="#">Change</a>
Users	<a href="#">+ Add</a>	<a href="#">Change</a>
BILLING		
Billings	<a href="#">+ Add</a>	<a href="#">Change</a>
Invoices	<a href="#">+ Add</a>	<a href="#">Change</a>
Plans	<a href="#">+ Add</a>	<a href="#">Change</a>
FONTS		
Font groups	<a href="#">+ Add</a>	<a href="#">Change</a>
Fonts	<a href="#">+ Add</a>	<a href="#">Change</a>
GLOSSARIES		
Glossaries	<a href="#">+ Add</a>	<a href="#">Change</a>
LEGAL		
Agreements	<a href="#">+ Add</a>	<a href="#">Change</a>
PYTHON SOCIAL AUTH		
Associations	<a href="#">+ Add</a>	<a href="#">Change</a>
Nonces	<a href="#">+ Add</a>	<a href="#">Change</a>
User social auths	<a href="#">+ Add</a>	<a href="#">Change</a>
SCREENSHOTS		
Screenshots	<a href="#">+ Add</a>	<a href="#">Change</a>
TRANSLATION MEMORY		
Memorys	<a href="#">+ Add</a>	<a href="#">Change</a>
WEBLATE LANGUAGES		
Languages	<a href="#">+ Add</a>	<a href="#">Change</a>
WEBLATE TRANSLATIONS		
Announcements	<a href="#">+ Add</a>	<a href="#">Change</a>
Component lists	<a href="#">+ Add</a>	<a href="#">Change</a>
Components	<a href="#">+ Add</a>	<a href="#">Change</a>
Contributor agreements	<a href="#">+ Add</a>	<a href="#">Change</a>
Projects	<a href="#">+ Add</a>	<a href="#">Change</a>

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):

Webplate administration

WELCOME, WEPLATE TEST, RETURN TO WEPLATE / DOCUMENTATION / CHANGE PASSWORD / LOG OUT

Home · Weblate translations · Projects · Add Project

Add Project

Required fields are marked in bold.

Project name:

WebplateOrg

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.

☒ Set "Language-Team" header

Lets Weblate update the "Language-Team" file header of your project.

☒ Use shared translation memory

Uses the pool of shared translations between projects.

☒ Contribute to shared translation memory

Contributes to the pool of shared translations between projects.

Access control:

Protected

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.

Source language:

English

Language used for source strings in all components

Language aliases:

Comma-separated list of language code mappings, for example: en\_GB,en\_US,en

Save and add another

Save and continue editing

SAVE

**Дивись також:**

*[Налаштування проєкту](#)*

### **Двомовні складники**

Після додавання проєкту до нього можна додати складники перекладу. (Див. *[Налаштування складників](#)*, щоб дізнатися більше про окремі параметри):



Webiate administration

HELLO! WEBLATE TESTRETURN TO WEBLATEDOCUMENTATIONCHANGEPASSWORDLOG OUT

[Home](#)[Webiate Translations](#)[Components](#)[Add Component](#)

Add Component

Secret: 00000000000000000000

Required fields are marked in bold.

Component name

Language names

Display name

URL slug

language-names

Name used in URLs and namespaces

Project

weblateOrg

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/learn.git

URL of a repository, use weblate://project/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/learn/branches/[[branch]]/[[filename]](file)

Link to repository browser, use [[branch]] for branch, [[filename]] and [[file]] as filename and the placeholders.

Exported repository URL

URL of repository where users can fetch changes from Weblate.

Source editing log 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/strings/translate/[[LC\_MESSAGES]].

Path of files to translate relative to repository root, use "" instead of language code, for example path for locale "LC\_MESSAGES" strings as.

Monolingual base language file

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, in most cases this is a translation file provided by developers and is used when creating actual source strings.

Template for new translations

weblate/strings/translate/strings.pot

Filename of file used for creating new translations. For gettext choose .pot file.

File format

gettext PO file

☐ Locked

Whether component will not get any translation updates.

☒ Allow translation propagation

Whether translation updates in other components will cause automatic translation in this one.

☒ Turn on suggestions

Whether to allow translation suggestions at all.

☐ Suggestion voting

Whether users can vote for suggestions.

Autoscript suggestions

0

Automatically accept 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

1

List of checks which can not 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.

Commit message when translating

Translated using Weblate ([ language\_name ])
Currently translated at [[ stats.translated\_percent |% ]] ([ stats.translated |] of [[ stats.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.

Commit 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.

Commit 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.

Commit 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.

Commit message when admin makes a change

Update translation files
Updated by "[[ admin\_name |]]" hook 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

no-reply@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.

Language filter

^[a-z]{2}\$

Regular expression used to filter translation when searching for strings.

Variables regular expression

Regular expression 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 continue editingSave

**Дивись також:**

*Налаштування складників, Bilingual and monolingual formats*

### **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):

## 352

## Розділ 2. Документація для адміністраторів

Website administration

HELLO! WEBSITE TEST Returns to HELPATE | DOCUMENTATION | CHANGE FEEDBACK | 236 hits

Home

Website translations

Components

Add Component

Support

Direct documentation

Feedback

Add Component

Required fields are marked by bold

Component name:

Android

URL slug:

android

Project:

TranslateCg

Version control system:

Git

Source code repository:

website/weblate/language-names

Repository push URL:

Repository browser:

Exported repository URL:

Source string reporting address:

Repository branch:

Repository branch to translate

Push branch:

Filters:

app/src/main/res/values \*/strings.xml

Mandatory base language file:

app/src/main/res/values strings.xml

Use base file

Whether users will be able to edit the base file for mandatory translations.

Intermediate translation file:

Template for new translations:

File format:

Android String Resource

Locked

Which component will not get an immediate updates

Allow translation propagation

Whether translation updates in other components will cause automatic translation or this one

Turn on suggestions

Whether to allow translation suggestions at all

Suggestion voting

Whether users can vote for suggestions

Accept/skip suggestions:

0

Translation Page:

Enforced checks:

Translation license:

Mit License

Contributor agreement:

Add new translation:

Create new language file

Language code style:

Default based on the file format

Merge style:

Rebase

Current message when translating:

Translated using Weblate (% language\_name %)  
Currently translated at (% states\_translated\_percent %) (% states\_translated %) of (% states\_all %) strings  
Translation of (% project\_name %) (% component\_name %)  
Translate info... (% url %)

Current message when adding translation:

Added translation using Weblate (% language\_name %)

Current message when removing translation:

Deleted translation using Weblate (% language\_name %)

Current message when merging translation:

Merge branch '% component\_remote\_branch' (% into Weblate).

Current message when add-on makes a change:

Update translation files  
Updated by '% add-on\_name %' hook in Weblate  
Translation of (% project\_name %) (% component\_name %)  
Translate info... (% url %)

Controller name:

Weblate

Push on commit

Whether the repository should be pushed upstream on every commit

Age of changes to commit:

24

Lock on error

Whether the component should be locked on repository errors

Language filter:

\*-id

Variants regular expression:

Priority:

Medium

Restricted component

Restrict access to the component to only those explicitly given permission

Save and add another

Save and continue editing

Save

Дивись також:

*Налаштовування складників, Bilingual and monolingual formats*

## 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 встановленого вами екземпляра.

The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo, 'Dashboard', 'Projects', 'Languages', and 'Checks' menus. On the right are icons for settings, adding new items, and a user profile. Below this is a 'Manage' section with a 'Weblate status' tab selected. The 'Weblate status' section shows 'Support status' as 'Community support' with buttons for 'Purchase support package' and 'Donate to Weblate'. Below this is the 'Activate support package' section, which includes a text box for an 'Activation token' and buttons for 'Activate' and 'Purchase support package'. The footer of the interface reads 'Powered by Weblate 4.2' followed by links for 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

### 2.24.2 Дані, які подаються до Weblate

- Адреса, за якою налаштовано ваш екземпляр Weblate
- Назва вашого сайту
- 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
- *Використання передбаченого у Weblate сховища резервних копій*

---

**Підказка:** Придбані пакунки підтримки є активованими з моменту придбання — ними можна користуватися без інтеграції.

---

## 2.25 Правові документи

---

**Примітка:** Herein you will find various legal information you might need to operate Weblate in certain legal jurisdictions. It is provided as a means of guidance, without any warranty of accuracy or correctness. It is ultimately your responsibility to ensure that your use of Weblate complies with all applicable laws and regulations.

---

### 2.25.1 ITAR та інші обмеження експортування

Weblate can be run within your own datacenter or virtual private cloud. As such, it can be used to store ITAR or other export-controlled information, however, end users are responsible for ensuring such compliance.

The Hosted Weblate service has not been audited for compliance with ITAR or other export controls, and does not currently offer the ability to restrict translations access by country.

### 2.25.2 US encryption controls

Weblate does not contain any cryptographic code, but might be subject export controls as it uses third party components utilizing cryptography for authentication, data-integrity and -confidentiality.

Most likely Weblate would be classified as ECCN 5D002 or 5D992 and, as publicly available libre software, it should not be subject to EAR (see [Encryption items NOT Subject to the EAR](#)).

Software components used by Weblate (listing only components related to cryptographic function):

**Python** Див. [https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is\\_Python\\_subject\\_to\\_export\\_laws.3F](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

The strength of encryption keys depend on the configuration of Weblate and the third party components it interacts with, but in any decent setup it will include all export restricted cryptographic functions:

- In excess of 56 bits for a symmetric algorithm
- Factorisation of integers in excess of 512 bits for an asymmetric algorithm
- Computation of discrete logarithms in a multiplicative group of a finite field of size greater than 512 bits for an asymmetric algorithm
- Discrete logarithms in a group different than above in excess of 112 bits for an asymmetric algorithm

Weblate doesn't have any cryptographic activation feature, but it can be configured in a way where no cryptography code would be involved. The cryptographic features include:

- 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 Contributing to Weblate

There are dozens of ways to contribute in Weblate. Any help is welcomed, be it coding, graphics design, documentation or sponsorship:

- *Надсилання звітів щодо вад у Weblate*
- *Starting contributing code to Weblate*
- *Переклад Weblate*
- *Фінансування розробки Weblate*

### 3.1.1 Переклад Weblate

Weblate is being [translated](#) using Weblate itself, feel free to take part in the effort of making Weblate available in as many human languages as possible.

### 3.1.2 Фінансування розробки Weblate

You can fund further Weblate development on the [donate page](#). Funds collected there are used to fund gratis hosting for libre software projects, and further development of Weblate. Please check the *donate page* for details, such as funding goals and rewards you can get for being a funder.

#### Backers who have funded Weblate

Список спонсорів Weblate:

- Yashiro Ccs
- Cheng-Chia Tseng
- Timon Reinhard
- [Cassidy James](#)
- Loic Dachary

- Marozed
- <https://freedombox.org/>
- GNU Solidario (GNU Health)

Do you want to be in the list? Please see options on the [Donate to Weblate](#).

## 3.2 Starting contributing code to Weblate

To understand Weblate source code, please first look into *Початковий код Weblate*, *Оболонка Weblate* and *Нумпуоуі Weblate*.

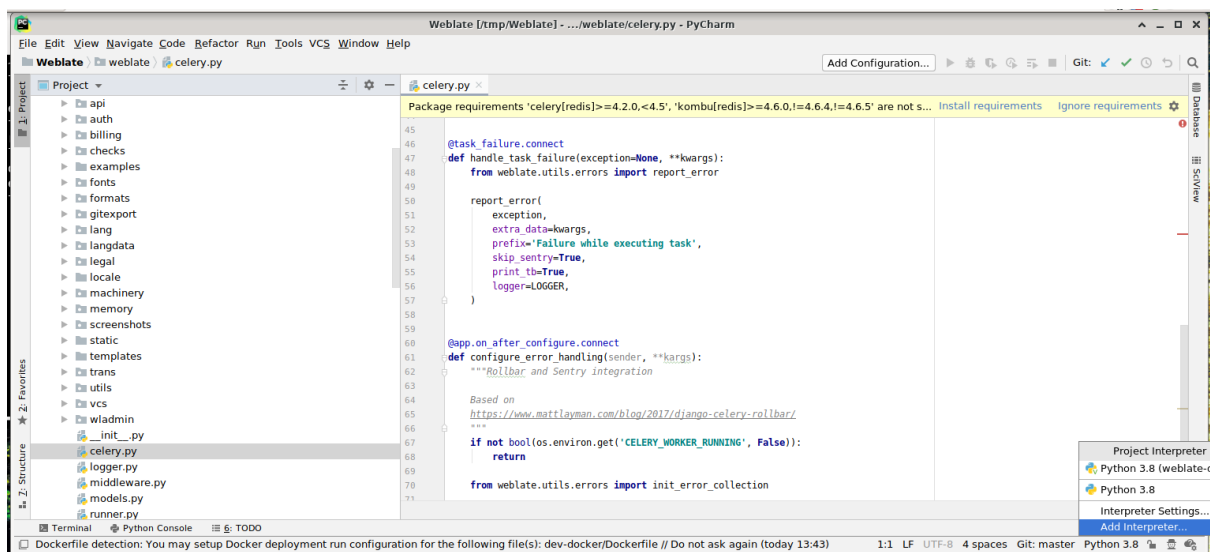
### 3.2.1 Starting with our codebase

If looking for some bugs to familiarize yourself with the Weblate codebase, look for ones labelled [good first issue](#).

### 3.2.2 Coding Weblate with PyCharm

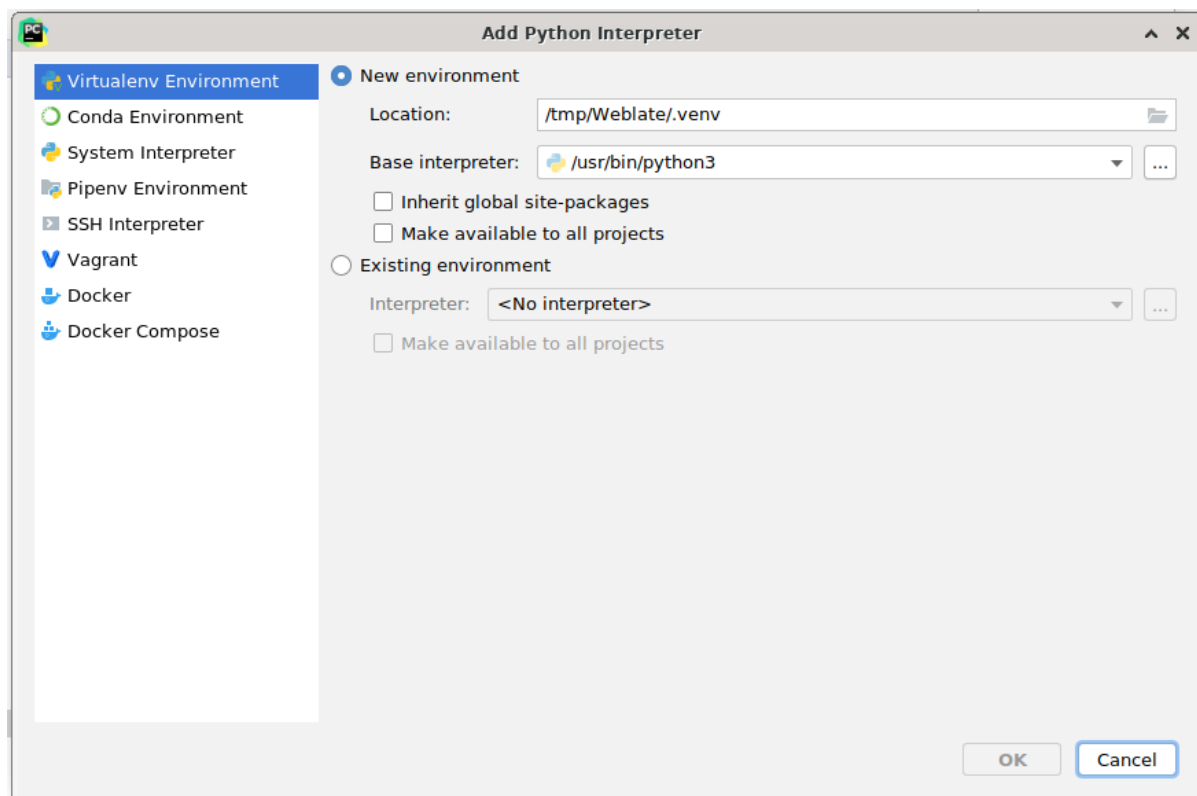
PyCharm is a known IDE for Python, here's some guidelines to help you setup Weblate project in it.

Considering you have just cloned the Github repository, just open the folder in which you cloned it in PyCharm. Once the IDE is open, the first step is to specify the interpreter you want:



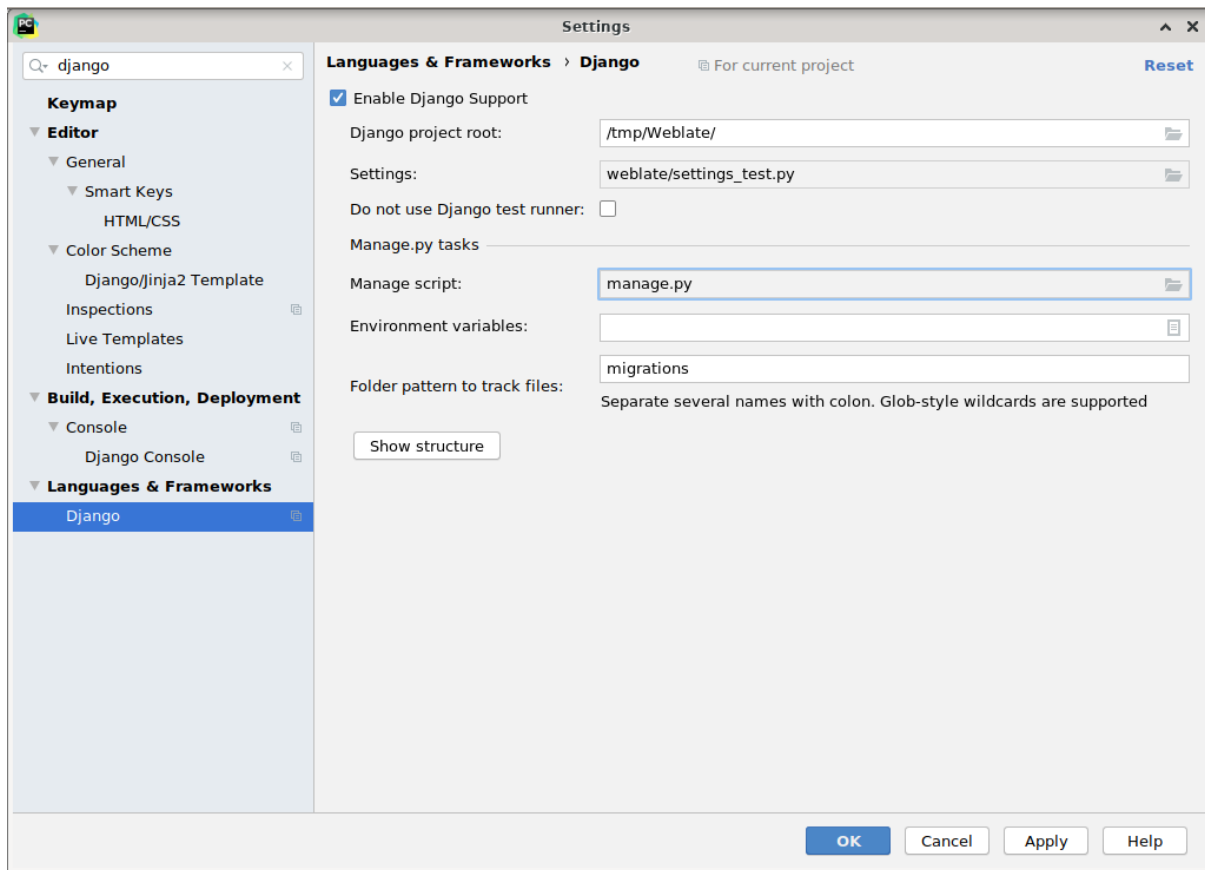
You can either chose to let PyCharm create the virtualenv for you, or select an already existing one:





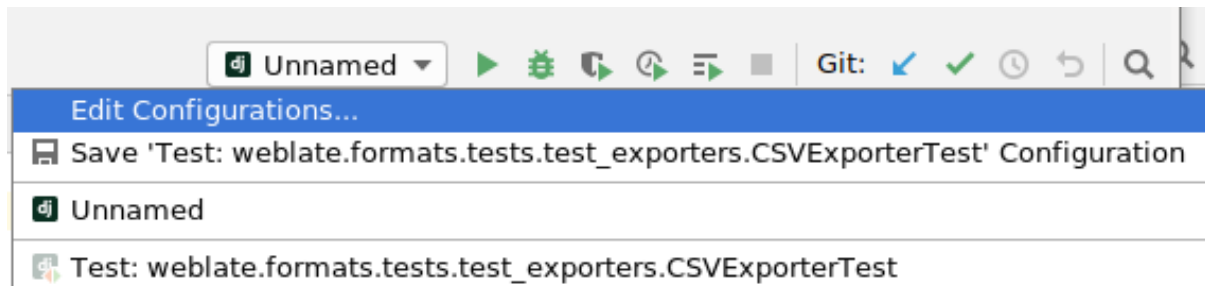
Don't forget to install the dependencies once the interpreter is set: you can do it, either through the console (the console from the IDE will directly use your virtualenv by default), or through the interface when you get a warning about missing dependencies.

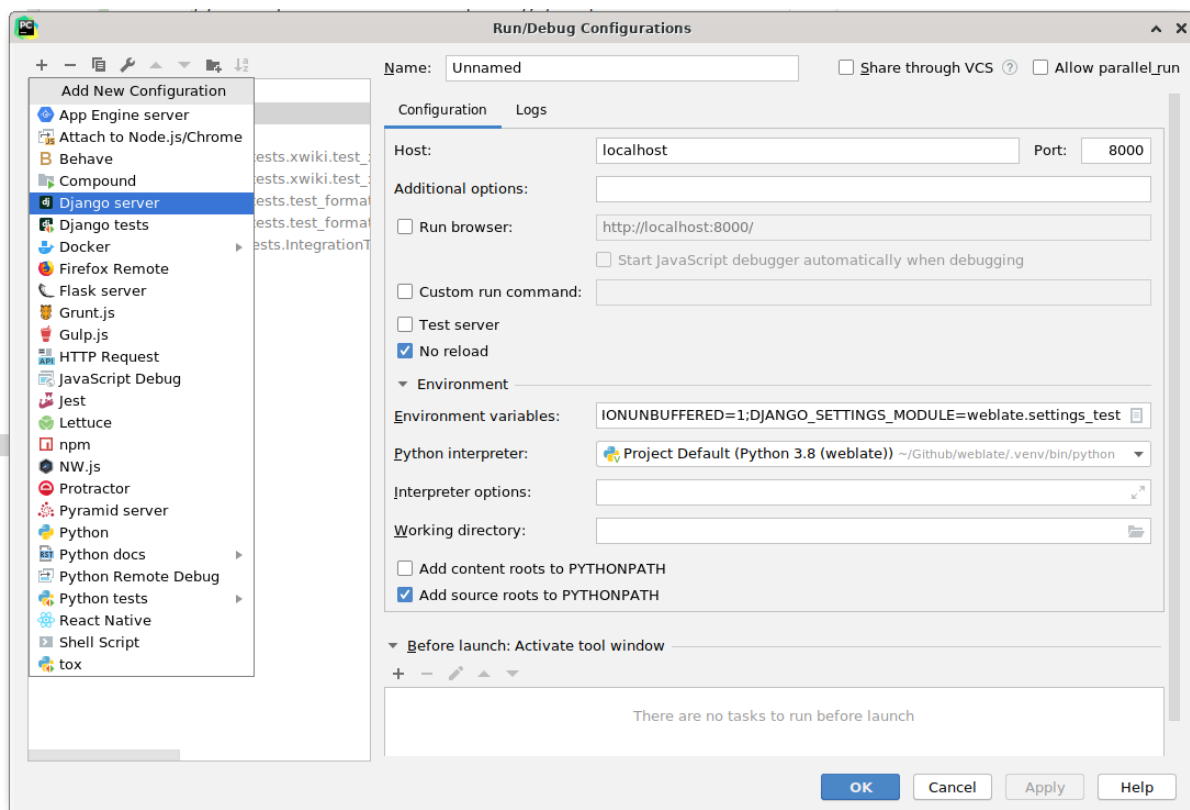
The second step is to set the right information to use natively Django inside PyCharm: the idea is to be able to immediately trigger the unit tests in the IDE. For that you need to specify the root path of Django and the path of one setting:



Be careful, the *Django project root* is the root of the repository, not the weblate sub-directory. About the settings, I personally use the *settings\_test* from the repository, but you could create your own setting and set it there.

Last step is to be able to run the server and to put breakpoints on the code to be able to debug it. This is done by creating a new *Django Server* configuration:





Be careful to properly checked «No reload»: you won't get anymore the server live reload if you modify some files, but the debugger will be stopped on the breakpoint you set.

### 3.2.3 Running Weblate locally

The most comfortable approach to get started with Weblate development is to follow *Встановлення з початкового коду*. It will get you a virtual env with editable Weblate sources.

To install all dependencies useful for development, do:

```
pip install -r requirements-dev.txt
```

To start a development server run:

```
weblate runserver
```

Depending on your configuration you might also want to start Celery workers:

```
./weblate/examples/celery start
```

### Running Weblate locally in Docker

If you have Docker and docker-compose installed, you can spin up the development environment simply by running:

```
./rundev.sh
```

It will create development Docker image and start it. Weblate is running on `<http://127.0.0.1:8080/>` and you can sign in with admin user and admin password. The new installation is empty, so you might want to continue with *Додавання проєктів і складників перекладу*.

The Dockerfile and docker-compose.yml for this are located in dev-docker directory.

The script also accepts some parameters, to execute tests run it with `test` parameter and then specify any `test` parameters, for example:

```
./rundevel.sh test --failfast weblate.trans
```

Be careful that your Docker containers are up and running before running the tests. You can check that by running the `docker ps` command.

To stop the background containers run:

```
./rundevel.sh stop
```

Running the script without args will recreate Docker container and restart it.

---

**Примітка:** This is not suitable setup for production, it includes several hacks which are insecure, but make development easier.

---

## 3.2.4 Bootstrapping your devel instance

You might want to use `import_demo` to create demo translations and `createadmin` to create admin user.

## 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 *Нумпуці 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

The code should follow PEP-8 coding guidelines and should be formatted using **black** code formatter.

To check the code quality, you can use **flake8**, the recommended plugins are listed in `.pre-commit-config.yaml` and it's configuration is placed in `setup.cfg`.

The easiest approach to enforce all this is to install [pre-commit](#). Weblate repository contains configuration for it to verify the committed files are sane. After installing it (it is already included in the `requirements-lint.txt`) turn it on by running `pre-commit install` in Weblate checkout. This way all your changes will be automatically checked.

Ви також можете запустити перевірку вручну. Щоб перевірити усі файли:

```
pre-commit run --all
```

## 3.4 Діагностика у Weblate

Bugs can behave as application crashes or as misbehavior. You are welcome to collect info on any such issue and submit it to the [issue tracker](#).

### 3.4.1 Режим налагодження

Turning on debug mode will make the exceptions show in the browser. This is useful to debug issues in the web interface, but not suitable for production environment as it has performance consequences and might leak private data.

**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).

Docker containers log to their output (as usual in the Docker world), so you can look at the logs using `docker-compose logs`.

**Дивись також:**

*[Зразок налаштувань](#) містить налаштування `LOGGING`.*

### 3.4.3 Аналіз аварійних завершень програми

In case the application crashes, it is useful to collect as much info about the crash as possible. The easiest way to achieve this is by using third-party services which can collect such info automatically. You can find info on how to set this up in [Збирання звітів щодо помилок](#).

### 3.4.4 Помилки без повідомлень

Lots of tasks are offloaded to Celery for background processing. Failures are not shown in the user interface, but appear in the Celery logs. Configuring [Збирання звітів щодо помилок](#) helps you to notice such failures easier.

### 3.4.5 Проблеми із швидкодією

In case Weblate performs badly in some situation, please collect the relevant logs showing the issue, and anything that might help figuring out where the code might be improved.

In case some requests take too long without any indication, you might want to install *dogslow* [<https://pypi.org/project/dogslow/>](https://pypi.org/project/dogslow/) along with [Збирання звітів щодо помилок](#) and get pinpointed and detailed tracebacks in the error collection tool.

## 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. [Running Weblate locally in 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.

`langdata`

Визначення даних мов.

`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 Оболонка Weblate

Поточну версію оболонки побудовано з використанням Bootstrap, jQuery і декількох сторонніх бібліотек.

### 3.6.1 Керування залежностями

The yarn package manager is used to update third party libraries. The configuration lives in `scripts/yarn` and there is a wrapper script `scripts/yarn-update` to upgrade the libraries, build them and copy to correct locations in `weblate/static/vendor`, where all third partly frontend code is located.

### 3.6.2 Стиль програмного коду

Weblate relies on [Prettier](#) for the code formatting for both JavaScript and CSS files.

We also use [ESLint](#) to check the JavaScript code.

### 3.6.3 Локалізація

Should you need any user visible text in the frontend code, it should be localizable. In most cases all you need is to wrap your text inside `gettext` function, but there are more complex features available:

```
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

### 3.6.4 Піктограми

Weblate currently uses material design icons, in case you are looking for new one, check [<https://materialdesignicons.com/>](https://materialdesignicons.com/).

Additionally, there is `scripts/optimize-svg` to reduce size of the SVG as most of the icons are embedded inside the HTML to allow styling of the paths.

## 3.7 Надсилання звітів щодо вад у Weblate

Нашу систему стеження за вадами розташовано на GitHub:

Feel welcome to report any issues with, or suggest improvement of Weblate there. If what you have found is a security issue in Weblate, please consult the «Security issues» section below.

### 3.7.1 Негаразди з безпекою

In order to give the community time to respond and upgrade you are strongly urged to report all security issues privately. HackerOne is used to handle security issues, and can be reported directly at [HackerOne](#).

Alternatively, report to [security@weblate.org](mailto:security@weblate.org), which ends up on HackerOne as well.

If you don't want to use HackerOne, for whatever reason, you can send the report by e-mail to [michal@cihar.com](mailto:michal@cihar.com). You can choose to encrypt it using this PGP key `3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D`.

---

**Примітка:** Weblate depends on third party components for many things. In case you find a vulnerability affecting one of those components in general, please report it directly to the respective project.

Some of these are:

- [Django](#)
  - [Бібліотека REST Django](#)
  - [Python Social Auth](#)
- 

## 3.8 Weblate testsuite and continuous integration

Testsuites exist for most of the current code, increase coverage by adding testcases for any new functionality, and verify that it works.



### 3.8.1 Continuous integration

Current test results can be found on [GitHub Actions](#) and coverage is reported on [Codecov](#).

There are several jobs to verify different aspects:

- Перевірка модулів
- Documentation build and external links
- Migration testing from all supported releases
- Code linting
- Setup verification (ensures that generated dist files do not miss anything and can be tested)

The configuration for the CI is in `.github/workflows` directory. It heavily uses helper scripts stored in `ci` directory. The scripts can be also executed manually, but they require several environment variables, mostly defining Django settings file to use and database connection. The example definition of that is in `scripts/test-database`:

```
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=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
```

The simple execution can look like:

```
. scripts/test-database
./ci/run-migrate
./ci/run-test
./ci/run-docs
./ci/run-setup
```

### 3.8.2 Local testing

To run a testsuite locally, use:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

---

**Підказка:** You will need a database (PostgreSQL) server to be used for tests. By default Django creates separate database to run tests with `test_` prefix, so in case your settings is configured to use `weblate`, the tests will use `test_weblate` database. See [Налаштування бази даних для Weblate](#) for setup instructions.

---

The `weblate/settings_test.py` is used in CI environment as well (see [Continuous integration](#)) and can be tuned using environment variables:

```
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=postgresql

# Database server configuration
```

(continues on next page)

(continued from previous page)

```
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
```

Prior to running tests you should collect static files as some tests rely on them being present:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

You can also specify individual tests to run:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

**Підказка:** The tests can also be executed inside developer docker container, see [Running Weblate locally in Docker](#).

Дивись також:

See [Testing in Django](#) for more info on running and writing tests for Django.

### 3.9 Data schemas

Weblate uses [JSON Schema](#) to define layout of external JSON files.

#### 3.9.1 Weblate Translation Memory Schema

type	array		
items			
•	The Translation Memory Item		
	type	object	
	properties		
	• category	The String Category	
		1 is global, 2 is shared, 10000000+ are project specific, 20000000+ are user specific	
		type	integer
		examples	1
		minimum	0
		default	1
	• origin	The String Origin	
		Filename or component name	
		type	string
		examples	test
		pattern	^(.*)\$
		default	
	• source	The Source String	
		type	string
		examples	Hello
		pattern	^(.+) \$
		default	
	• source_language	The Source Language	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
type		string	

continues on next page

Table 1 – continued from previous page

		examples	en
		pattern	^([ ^ ]+)\$
		default	
	• <b>target</b>	<i>The Target String</i>	
		type	<i>string</i>
		examples	Ahoj
		pattern	^(.+)\$
		default	
	• <b>target_language</b>	<i>The Target Language</i>	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		type	<i>string</i>
		examples	cs
		pattern	^([ ^ ]+)\$
		default	
	additionalProperties		False
definitions			

### 3.9.2 Weblate user data export

type	<i>object</i>		
properties			
• <b>basic</b>	<i>Basic</i>		
	type	<i>object</i>	
	properties		
	• <b>username</b>	<i>Username</i>	
		type	<i>string</i>
		examples	admin
		pattern	^.*\$
		default	
	• <b>full_name</b>	<i>Full name</i>	
		type	<i>string</i>
		examples	Weblate Admin
		pattern	^.*\$
		default	
	• <b>email</b>	<i>E-mail</i>	
		type	<i>string</i>
		examples	noreply@example.com
		pattern	^.*\$
		default	
	• <b>date_joined</b>	<i>Date joined</i>	
		type	<i>string</i>
		examples	2019-11-18T18:53:54.862Z
		pattern	^.*\$
		default	
• <b>profile</b>	<i>Profile</i>		
	type	<i>object</i>	
	properties		
	• <b>language</b>	<i>Language</i>	
		type	<i>string</i>
		examples	cs
		pattern	^.*\$
		default	
	• <b>suggested</b>	<i>Number of sugested strings</i>	
		type	<i>integer</i>

continues on next page

Table 2 – continued from previous page

		examples	1			
		default	0			
	• translated	Number of translated strings				
		type	integer			
		examples	24			
		default	0			
	• uploaded	Number of uploaded screenshots				
		type	integer			
		examples	1			
		default	0			
	• hide_completed	Hide completed translations on the dashboard				
		type	boolean			
		examples	False			
		default	True			
	• secondary_in_zen	Show secondary translations in the Zen mode				
		type	boolean			
		examples	True			
		default	True			
	• hide_source_secondary	Hide source if a secondary translation exists				
		type	boolean			
		examples	False			
		default	True			
	• editor_link	Editor link				
		type	string			
		examples				
		pattern	^.*\$			
		default				
	• translate_mode	Translation editor mode				
		type	integer			
		examples	0			
		default	0			
	• zen_mode	Zen editor mode				
		type	integer			
		examples	0			
		default	0			
	• special_chars	Special characters				
		type	string			
		examples				
		pattern	^.*\$			
		default				
	• dashboard_view	Default dashboard view				
		type	integer			
		examples	1			
		default	0			
	• dashboard_component_list	Default component list				
		default	None			
		anyOf	•	type	null	
			•	type	integer	
	• languages	Translated languages				
		type	array			
		default	[]			
items						

continues on next page

Table 2 – continued from previous page

			Language code		
			type	string	
			examples	cs	
			pattern	^.*\$	
			default		
	• secondary	Secondary languages			
		languages	array		
		default	[]		
		items			
			Language code		
			type	string	
			examples	sk	
			pattern	^.*\$	
			default		
		• watched	Watched projects		
			type	array	
			default	[]	
			items		
				Project slug	
				type	string
examples	weblate				
pattern	^.*\$				
default					
• auditlog	Audit log				
	type	array			
	default	[]			
	items				
		Items			
		type	object		
		properties			
		• address	IP address		
			type	string	
			examples	127.0.0.1	
			pattern	^.*\$	
			default		
		• user_agent	User agent		
			type	string	
			examples	PC / Linux / Firefox 70.0	
			pattern	^.*\$	
			default		
		• timestamp	Timestamp		
			type	string	
			examples	2019-11-18T18:58:30.845Z	
			pattern	^.*\$	
			default		
		• activity	Activity		
			type	string	
			examples	login	
			pattern	^.*\$	
			default		
		definitions			

Дивись також:

Пам'ять перекладів, *dump\_memory*, *import\_memory*

## 3.10 Створення випусків Weblate

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. Переконалися, 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. Once the Docker image is tested, add a tag and push it.
8. Update Helm chart to new version.
9. Include new version in `.github/workflows/migrations.yml` to cover it in migration testing.
10. Increase version in the repository by `./scripts/set-version`.

To create tags using the `./scripts/create-release` script you will need following:

- GnuPG із закритим ключем, який використовується для підписування випусків
- Push access to Weblate git repositories (it pushes tags)
- Configured **hub** tool and access to create releases on the Weblate repo
- SSH access to Weblate download server (the Website downloads are copied there)

## 3.11 Про Weblate

### 3.11.1 Цілі проєкту

Web-based continuous localization tool with tight *Налаштування інтеграції із керуванням версіями* supporting a wide range of *Підтримувані формати файлів*, making it easy for translators to contribute.

### 3.11.2 Назва проєкту

«Weblate» is a portmanteau of the words «web» and «translate».

### 3.11.3 Вебсайт проекту

The landing page is <<https://weblate.org/>> and a cloud hosted service at <<https://hosted.weblate.org/>>. This documentation can be found on <<https://docs.weblate.org/>>.

### 3.11.4 Логотипи проекту

Логотипи проекту та інші графічні роботи можна знайти у сховищі <<https://github.com/WeblateOrg/graphics/>>.

### 3.11.5 Лідерство

Супровід цього проекту виконується Міхалом Чихаржем (Michal Čihař) <[michal@cihar.com](mailto:michal@cihar.com)>.

### 3.11.6 Автори

Розробку Weblate було розпочато Міхалом Чихаржем (Michal Čihař) <[michal@cihar.com](mailto:michal@cihar.com)>. З часу появи програми у 2012 внесок до неї зробили тисячі людей.

## 3.12 Ліцензія

© Michal Čihař <[michal@cihar.com](mailto: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 4.2

Released on August 18th 2020.

- 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.

## 4.2 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.
- Added support for acting as SAML Service Provider, see [Розпізнавання за SAML](#).

## 4.3 Weblate 4.1

Released on June 15th 2020.

- Added support for creating new translations with included country code.
- Added support for searching source strings with screenshot.
- Extended info available in the stats insights.
- Improved search editing on «Translate» pages.
- Improve handling of concurrent repository updates.
- Include source language in project creation form.
- Include changes count in credits.
- Fixed UI language selection in some cases.
- Allow to whitelist registration methods with registrations closed.
- Improved lookup of related terms in glossary.
- Improved translation memory matches.
- Group same machinery results.
- Add direct link to edit screenshot from translate page.
- Improved removal confirmation dialog.

- Include templates in ZIP download.
- Add support for Markdown and notification configuration in announcements.
- Extended details in check listings.
- Added support for new file formats: *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*.
- Consistently use dismissed as state of dismissed checks.
- Add support for configuring default addons to enable.
- Fixed editor keyboard shortcut to dismiss checks.
- Improved machine translation of strings with placeholders.
- Show ghost translation for user languages to ease starting them.
- Improved language code parsing.
- Show translations in user language first in the list.
- Renamed shapings to more generic name variants.
- Added new quality checks: *Декілька змінних без назв*, *Тривалий час не перекладено*, *Послідовне повторення слів*.
- Reintroduced support for wiping translation memory.
- Fixed option to ignore source checks.
- Added support for configuring different branch for pushing changes.
- API now reports rate limiting status in the HTTP headers.
- Added support for Google Translate V3 API (Advanced).
- Added ability to restrict access on component level.
- Added support for whitespace and other special chars in translation flags, see *Налаштування поведінки*.
- Always show rendered text check if enabled.
- API now supports filtering of changes.
- Added support for sharing glossaries between projects.

## 4.4 Weblate 4.0.4

Released on May 07th 2020.

- Fixed testsuite execution on some Python 3.8 environments.
- Typo fixes in the documentation.
- Fixed creating components using API in some cases.
- Fixed JavaScript errors breaking mobile navigation.
- Fixed crash on displaying some checks.
- Fixed screenshots listing.
- Fixed monthly digest notifications.
- Fixed intermediate translation behavior with units non existing in translation.

## 4.5 Weblate 4.0.3

Released on May 02nd 2020.

- Fixed possible crash in reports.
- User mentions in comments are now case insensitive.
- Fixed PostgreSQL migration for non superusers.
- Fixed changing the repository URL while creating component.
- Fixed crash when upstream repository is gone.

## 4.6 Weblate 4.0.2

Released on April 27th 2020.

- Improved performance of translation stats.
- Improved performance of changing labels.
- Improved bulk edit performance.
- Improved translation memory performance.
- Fixed possible crash on component deletion.
- Fixed displaying of translation changes in some corner cases.
- Improved warning about too long celery queue.
- Fixed possible false positives in the consistency check.
- Fixed deadlock when changing linked component repository.
- Included edit distance in changes listing and CSV and reports.
- Avoid false positives of punctuation spacing check for Canadian French.
- Fixed XLIFF export with placeholders.
- Fixed false positive with zero width check.
- Improved reporting of configuration errors.
- Fixed bilingual source upload.
- Automatically detect supported languages for DeepL machine translation.
- Fixed progress bar display in some corner cases.
- Fixed some checks triggering on non translated strings.

## 4.7 Weblate 4.0.1

Released on April 16th 2020.

- Fixed package installation from PyPI.

## 4.8 Weblate 4.0

Released on April 16th 2020.

- Weblate now requires Python 3.6 or newer.
- Added management overview of component alerts.
- Added component alert for broken repository browser URLs.
- Improved sign in and registration pages.
- Project access control and workflow configuration integrated to project settings.
- Added check and highlighter for i18next interpolation and nesting.
- Added check and highlighter for percent placeholders.
- Показувати пропозиції, які не проходять перевірки.
- Record source string changes in history.
- Upgraded Microsoft Translator to version 3 API.
- Reimplemented translation memory backend.
- Added support for several `is:` lookups in *Пошук*.
- Allow to make *Переклад не змінено* avoid internal blacklist.
- Improved comments extraction from monolingual po files.
- Renamed whiteboard messages to announcements.
- Fixed occasional problems with registration mails.
- Improved LINGUAS update addon to handle more syntax variants.
- Fixed editing monolingual XLIFF source file.
- Added support for exact matching in *Пошук*.
- Extended API to cover screenshots, users, groups, componentlists and extended creating projects.
- Add support for source upload on bilingual translations.
- Added support for intermediate language from developers.
- Added support for source strings review.
- Extended download options for platform wide translation memory.

## 4.9 Weblate 3.x series

### 4.9.1 Weblate 3.11.3

Released on March 11th 2020.

- Fixed searching for fields with certain priority.
- Fixed predefined query for recently added strings.
- Fixed searching returning duplicate matches.
- Fixed notifications rendering in Gmail.
- Fixed reverting changes from the history.
- Added links to events in digest notifications.
- Fixed email for account removal confirmation.

- Added support for Slack authentication in Docker container.
- Avoid sending notifications for not subscribed languages.
- Include Celery queues in performance overview.
- Fixed documentation links for addons.
- Reduced false negatives for unchanged translation check.
- Raised bleach dependency to address CVE-2020-6802.
- Fixed listing project level changes in history.
- Fixed stats invalidation in some corner cases.
- Fixed searching for certain string states.
- Improved format string checks behavior on missing percent.
- Fixed authentication using some third party providers.

### 4.9.2 Weblate 3.11.2

Released on February 22nd 2020.

- Fixed rendering of suggestions.
- Fixed some strings wrongly reported as having no words.

### 4.9.3 Weblate 3.11.1

Випущено 20 лютого 2020 року.

- Documented Celery setup changes.
- Improved filename validation on component creation.
- Fixed minimal versions of some dependencies.
- Fixed adding groups with certain Django versions.
- Fixed manual pushing to upstream repository.
- Improved glossary matching.

### 4.9.4 Weblate 3.11

Випущено 17 лютого 2012 року.

- Allow using VCS push URL during component creation via API.
- Rendered width check now shows image with the render.
- Fixed links in notifications e-mails.
- Improved look of plaintext e-mails.
- Display ignored checks and allow to make them active again.
- Display nearby keys on monolingual translations.
- Додано підтримку групування форм рядків.
- Recommend upgrade to new Weblate versions in the system checks.
- Provide more detailed analysis for duplicate language alert.
- Include more detailed license info on the project pages.

- Automatically unshallow local copies if needed.
- Fixed download of strings needing action.
- New alert to warn about using the same filemask twice.
- Improve XML placeables extraction.
- The `SINGLE_PROJECT` can now enforce redirection to chosen project.
- Added option to resolve comments.
- Added bulk editing of flags.
- Added support for *Мітки рядків*.
- Added bulk edit addon.
- Додано параметр для *Примусові перевірки*.
- Increased default validity of confirmation links.
- Improved Matomo integration.
- Fixed *Вже перекладено* to correctly handle source string change.
- Extended automatic updates configuration by `AUTO_UPDATE`.
- LINGUAS addons now do full sync of translations in Weblate.

### 4.9.5 Weblate 3.10.3

Released on January 18th 2020.

- Support for translate-toolkit 2.5.0.

### 4.9.6 Weblate 3.10.2

Released on January 18th 2020.

- Add lock indication to projects.
- Fixed CSS bug causing flickering in some web browsers.
- Fixed searching on systems with non-English locales.
- Improved repository matching for GitHub and Bitbucket hooks.
- Fixed data migration on some Python 2.7 installations.
- Allow configuration of Git shallow cloning.
- Improved background notification processing.
- Fixed broken form submission when navigating back in web browser.
- New addon to configure YAML formatting.
- Fixed same plurals check to not fire on single plural form languages.
- Fixed regex search on some fields.

### 4.9.7 Weblate 3.10.1

Released on January 9th 2020.

- Extended API with translation creation.
- Fixed several corner cases in data migrations.
- Compatibility with Django 3.0.
- Improved data cleanup performance.
- Added support for customizable security.txt.
- Improved breadcrumbs in changelog.
- Improved translations listing on dashboard.
- Improved HTTP responses for webhooks.
- Added support for GitLab merge requests in Docker container.

### 4.9.8 Weblate 3.10

Released on December 20th 2019.

- Improved application user interface.
- Added doublespace check.
- Fixed creating new languages.
- Avoid sending auditlog notifications to deleted e-mails.
- Added support for read only strings.
- Added support for Markdown in comments.
- Allow placing translation instruction text in project info.
- Add copy to clipboard for secondary languages.
- Improved support for Mercurial.
- Improved Git repository fetching performance.
- Add search lookup for age of string.
- Show source language for all translations.
- Show context for nearby strings.
- Added support for notifications on repository operations.
- Improved translation listings.
- Extended search capabilities.
- Added support for automatic translation strings marked for editing.
- Avoid sending duplicate notifications for linked component alerts.
- Improve default merge request message.
- Better indicate string state in Zen mode.
- Added support for more languages in Yandex Translate.
- Improved look of notification e-mails.
- Provide choice for translation license.

### 4.9.9 Weblate 3.9.1

Released on October 28th 2019.

- Remove some unneeded files from backups.
- Fixed potential crash in reports.
- Fixed cross database migration failure.
- Added support for force pushing Git repositories.
- Reduced risk of registration token invalidation.
- Fixed account removal hitting rate limiter.
- Added search based on priority.
- Fixed possible crash on adding strings to JSON file.
- Safe HTML check and fixup now honor source string markup.
- Avoid sending notifications to invited and deleted users.
- Fix SSL connection to redis in Celery in Docker container.

### 4.9.10 Weblate 3.9

Released on October 15th 2019.

- Include Weblate metadata in downloaded files.
- Improved UI for failing checks.
- Indicate missing strings in format checks.
- Separate check for French punctuation spacing.
- Add support for fixing some of quality checks errors.
- Add separate permission to create new projects.
- Extend stats for char counts.
- Improve support for Java style language codes.
- Added new generic check for placeholders.
- Added support for WebExtension JSON placeholders.
- Added support for flat XML format.
- Extended API with project, component and translation removal and creation.
- Added support for Gitea and Gitee webhooks.
- Added new custom regex based check.
- Allow to configure contributing to shared translation memory.
- Added ZIP download for more translation files.
- Make XLIFF standard compliant parsing of maxwidth and font.
- Added new check and fixer for safe HTML markup for translating web applications.
- Add component alert on unsupported configuration.
- Added automatic translation addon to bootstrap translations.
- Extend automatic translation to add suggestions.
- Display addon parameters on overview.



- Sentry is now supported through modern Sentry SDK instead of Raven.
- Changed example settings to be better fit for production environment.
- Added automated backups using BorgBackup.
- Split cleanup addon for RESX to avoid unwanted file updates.
- Added advanced search capabilities.
- Allow users to download their own reports.
- Added localization guide to help configuring components.
- Added support for GitLab merge requests.
- Improved display of repository status.
- Perform automated translation in the background.

### 4.9.11 Weblate 3.8

Released on August 15th 2019.

- Added support for simplified creating of similar components.
- Added support for parsing translation flags from the XML based file formats.
- Log exceptions into Celery log.
- Improve performance of repository scoped addons.
- Improved look of notification e-mails.
- Fixed password reset behavior.
- Improved performance on most of translation pages.
- Fixed listing of languages not known to Weblate.
- Add support for cloning addons to discovered components.
- Add support for replacing file content with uploaded.
- Add support for translating non VCS based content.
- Added OpenGraph widget image to use on social networks.
- Added support for animated screenshots.
- Improved handling of monolingual XLIFF files.
- Avoid sending multiple notifications for single event.
- Add support for filtering changes.
- Extended predefined periods for reporting.
- Added webhook support for Azure Repos.
- New opt-in notifications on pending suggestions or untranslated strings.
- Add one click unsubscribe link to notification e-mails.
- Fixed false positives with Has been translated check.
- New management interface for admins.
- String priority can now be specified using flags.
- Added language management views.
- Add checks for Qt library and Ruby format strings.
- Added configuration to better fit single project installations.

- Notify about new string on source string change on monolingual translations.
- Added separate view for translation memory with search capability.

#### 4.9.12 Weblate 3.7.1

Released on June 28th 2019.

- Documentation updates.
- Fixed some requirements constraints.
- Updated language database.
- Localization updates.
- Various user interface tweaks.
- Improved handling of unsupported but discovered translation files.
- More verbosely report missing file format requirements.

#### 4.9.13 Weblate 3.7

Released on June 21st 2019.

- Added separate Celery queue for notifications.
- Use consistent look with application for API browsing.
- Include approved stats in the reports.
- Report progress when updating translation component.
- Allow to abort running background component update.
- Extend template language for filename manipulations.
- Use templates for editor link and repository browser URL.
- Indicate max length and current characters count when editing translation.
- Improved handling of abbreviations in unchanged translation check.
- Refreshed landing page for new contributors.
- Add support for configuring msgmerge addon.
- Delay opening SMTP connection when sending notifications.
- Improved error logging.
- Allow custom location in MO generating addon.
- Added addons to cleanup old suggestions or comments.
- Added option to enable horizontal mode in the Zen editor.
- Improved import performance with many linked components.
- Fixed examples installation in some cases.
- Improved rendering of alerts in changes.
- Added new horizontal stats widget.
- Improved format strings check on plurals.
- Added font management tool.
- New check for rendered text dimensions.

- Added support for subtitle formats.
- Include overall completion stats for languages.
- Added reporting at project and global scope.
- Improved user interface when showing translation status.
- New Weblate logo and color scheme.
- New look of bitmap badges.

### 4.9.14 Weblate 3.6.1

Released on April 26th 2019.

- Improved handling of monolingual XLIFF files.
- Fixed digest notifications in some corner cases.
- Fixed addon script error alert.
- Fixed generating MO file for monolingual PO files.
- Fixed display of uninstalled checks.
- Indicate administered projects on project listing.
- Allow update to recover from missing VCS repository.

### 4.9.15 Weblate 3.6

Released on April 20th 2019.

- Add support for downloading user data.
- Addons are now automatically triggered upon installation.
- Improved instructions for resolving merge conflicts.
- Cleanup addon is now compatible with app store metadata translations.
- Configurable language code syntax when adding new translations.
- Warn about using Python 2 with planned termination of support in April 2020.
- Extract special characters from the source string for visual keyboard.
- Extended contributor stats to reflect both source and target counts.
- Admins and consistency addons can now add translations even if disabled for users.
- Fixed description of toggle disabling Language-Team header manipulation.
- Notify users mentioned in comments.
- Removed file format autodetection from component setup.
- Fixed generating MO file for monolingual PO files.
- Added digest notifications.
- Added support for muting component notifications.
- Added notifications for new alerts, whiteboard messages or components.
- Notifications for administered projects can now be configured.
- Improved handling of three letter language codes.

### 4.9.16 Weblate 3.5.1

Released on March 10th 2019.

- Fixed Celery systemd unit example.
- Fixed notifications from HTTP repositories with login.
- Fixed race condition in editing source string for monolingual translations.
- Include output of failed addon execution in the logs.
- Improved validation of choices for adding new language.
- Allow to edit file format in component settings.
- Update installation instructions to prefer Python 3.
- Performance and consistency improvements for loading translations.
- Make Microsoft Terminology service compatible with current Zeeb releases.
- Localization updates.

### 4.9.17 Weblate 3.5

Released on March 3rd 2019.

- Improved performance of built-in translation memory.
- Added interface to manage global translation memory.
- Improved alerting on bad component state.
- Added user interface to manage whiteboard messages.
- Addon commit message now can be configured.
- Reduce number of commits when updating upstream repository.
- Fixed possible metadata loss when moving component between projects.
- Improved navigation in the Zen mode.
- Added several new quality checks (Markdown related and URL).
- Added support for app store metadata files.
- Added support for toggling GitHub or Gerrit integration.
- Added check for Kashida letters.
- Added option to squash commits based on authors.
- Improved support for XLSX file format.
- Compatibility with Tesseract 4.0.
- Billing addon now removes projects for unpaid billings after 45 days.

### 4.9.18 Weblate 3.4

Released on January 22nd 2019.

- Added support for XLIFF placeholders.
- Celery can now utilize multiple task queues.
- Added support for renaming and moving projects and components.
- Include characters counts in reports.
- Added guided adding of translation components with automatic detection of translation files.
- Customizable merge commit messages for Git.
- Added visual indication of component alerts in navigation.
- Improved performance of loading translation files.
- New addon to squash commits prior to push.
- Improved displaying of translation changes.
- Changed default merge style to rebase and made that configurable.
- Better handle private use subtags in language code.
- Improved performance of fulltext index updates.
- Extended file upload API to support more parameters.

### 4.9.19 Weblate 3.3

Released on November 30th 2018.

- Added support for component and project removal.
- Improved performance for some monolingual translations.
- Added translation component alerts to highlight problems with a translation.
- Expose XLIFF string rename as context when available.
- Added support for XLIFF states.
- Added check for non writable files in DATA\_DIR.
- Improved CSV export for changes.

### 4.9.20 Weblate 3.2.2

Released on October 20th 2018.

- Remove no longer needed Babel dependency.
- Updated language definitions.
- Improve documentation for addons, LDAP and Celery.
- Fixed enabling new dos-eol and auto-java-messageformat flags.
- Fixed running setup.py test from PyPI package.
- Improved plurals handling.
- Fixed translation upload API failure in some corner cases.
- Fixed updating Git configuration in case it was changed manually.

### 4.9.21 Weblate 3.2.1

Released on October 10th 2018.

- Document dependency on backports.csv on Python 2.7.
- Fix running tests under root.
- Improved error handling in gitexport module.
- Fixed progress reporting for newly added languages.
- Correctly report Celery worker errors to Sentry.
- Fixed creating new translations with Qt Linguist.
- Fixed occasional fulltext index update failures.
- Improved validation when creating new components.
- Added support for cleanup of old suggestions.

### 4.9.22 Weblate 3.2

Released on October 6th 2018.

- Add `install_addon` management command for automated addon installation.
- Allow more fine grained `ratelimit` settings.
- Added support for export and import of Excel files.
- Improve component cleanup in case of multiple component discovery addons.
- Rewritten Microsoft Terminology machine translation backend.
- Weblate now uses Celery to offload some processing.
- Improved search capabilities and added regular expression search.
- Added support for Youdao Zhiyun API machine translation.
- Added support for Baidu API machine translation.
- Integrated maintenance and cleanup tasks using Celery.
- Improved performance of loading translations by almost 25%.
- Removed support for merging headers on upload.
- Removed support for custom commit messages.
- Configurable editing mode (`zen/full`).
- Added support for error reporting to Sentry.
- Added support for automated daily update of repositories.
- Added support for creating projects and components by users.
- Built in translation memory now automatically stores translations done.
- Users and projects can import their existing translation memories.
- Better management of related strings for screenshots.
- Added support for checking Java MessageFormat.

See [3.2 milestone on GitHub](#) for detailed list of addressed issues.

### 4.9.23 Weblate 3.1.1

Released on July 27th 2018.

- Fix testsuite failure on some setups.

### 4.9.24 Weblate 3.1

Released on July 27th 2018.

- Upgrades from older version than 3.0.1 are not supported.
- Allow to override default commit messages from settings.
- Improve webhooks compatibility with self hosted environments.
- Added support for Amazon Translate.
- Compatibility with Django 2.1.
- Django system checks are now used to diagnose problems with installation.
- Removed support for soon shutdown libavatar service.
- New addon to mark unchanged translations as needing edit.
- Add support for jumping to specific location while translating.
- Downloaded translations can now be customized.
- Improved calculation of string similarity in translation memory matches.
- Added support by signing Git commits by GnuPG.

### 4.9.25 Weblate 3.0.1

Released on June 10th 2018.

- Fixed possible migration issue from 2.20.
- Localization updates.
- Removed obsolete hook examples.
- Improved caching documentation.
- Fixed displaying of admin documentation.
- Improved handling of long language names.

### 4.9.26 Weblate 3.0

Released on June 1st 2018.

- Rewritten access control.
- Several code cleanups that lead to moved and renamed modules.
- New addon for automatic component discovery.
- The import\_project management command has now slightly different parameters.
- Added basic support for Windows RC files.
- New addon to store contributor names in PO file headers.
- The per component hook scripts are removed, use addons instead.
- Add support for collecting contributor agreements.

- Access control changes are now tracked in history.
- New addon to ensure all components in a project have same translations.
- Support for more variables in commit message templates.
- Add support for providing additional textual context.

## 4.10 Weblate 2.x series

### 4.10.1 Weblate 2.20

Released on April 4th 2018.

- Improved speed of cloning subversion repositories.
- Changed repository locking to use third party library.
- Added support for downloading only strings needing action.
- Added support for searching in several languages at once.
- New addon to configure gettext output wrapping.
- New addon to configure JSON formatting.
- Added support for authentication in API using RFC 6750 compatible Bearer authentication.
- Added support for automatic translation using machine translation services.
- Added support for HTML markup in whiteboard messages.
- Added support for mass changing state of strings.
- Translate-toolkit at least 2.3.0 is now required, older versions are no longer supported.
- Added built in translation memory.
- Added componentlists overview to dashboard and per component list overview pages.
- Added support for DeepL machine translation service.
- Machine translation results are now cached inside Weblate.
- Додано підтримку переупорядкування надісланих змін.

### 4.10.2 Weblate 2.19.1

Випущено 20 лютого 2018 року.

- Fixed migration issue on upgrade from 2.18.
- Improved file upload API validation.

### 4.10.3 Weblate 2.19

Випущено 15 лютого 2018 року.

- Fixed imports across some file formats.
- Display human friendly browser information in audit log.
- Added TMX exporter for files.
- Various performance improvements for loading translation files.
- Added option to disable access management in Weblate in favor of Django one.



- Improved glossary lookup speed for large strings.
- Compatibility with django\_auth\_ldap 1.3.0.
- Configuration errors are now stored and reported persistently.
- Honor ignore flags in whitespace autofixer.
- Improved compatibility with some Subversion setups.
- Improved built in machine translation service.
- 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.

### 4.10.4 Weblate 2.18

Випущено 15 грудня 2017 року.

- Extended contributor stats.
- Improved configuration of special characters virtual keyboard.
- Added support for DTD file format.
- Changed keyboard shortcuts to less likely collide with browser/system ones.
- Improved support for approved flag in XLIFF files.
- Added support for not wrapping long strings in gettext PO files.
- Added button to copy permalink for current translation.
- Dropped support for Django 1.10 and added support for Django 2.0.
- Removed locking of translations while translating.
- Added support for adding new strings to monolingual translations.
- Added support for translation workflows with dedicated reviewers.

### 4.10.5 Weblate 2.17.1

Released on October 13th 2017.

- Fixed running testsuite in some specific situations.
- Locales updates.

### 4.10.6 Weblate 2.17

Released on October 13th 2017.

- Weblate by default does shallow Git clones now.
- Improved performance when updating large translation files.
- Added support for blocking certain e-mails from registration.
- Users can now delete their own comments.
- Added preview step to search and replace feature.
- Client side persistence of settings in search and upload forms.
- Extended search capabilities.
- More fine grained per project ACL configuration.
- Default value of BASE\_DIR has been changed.
- Added two step account removal to prevent accidental removal.
- Project access control settings is now editable.
- Added optional spam protection for suggestions using Akismet.

### 4.10.7 Weblate 2.16

Released on August 11th 2017.

- Various performance improvements.
- Added support for nested JSON format.
- Added support for WebExtension JSON format.
- Fixed git exporter authentication.
- Improved CSV import in certain situations.
- Improved look of Other translations widget.
- The max-length checks is now enforcing length of text in form.
- Make the commit\_pending age configurable per component.
- Various user interface cleanups.
- Fixed component/project/site wide search for translations.

### 4.10.8 Weblate 2.15

Released on June 30th 2017.

- Show more related translations in other translations.
- Add option to see translations of current string to other languages.
- Use 4 plural forms for Lithuanian by default.
- Fixed upload for monolingual files of different format.
- Improved error messages on failed authentication.
- Keep page state when removing word from glossary.
- Added direct link to edit secondary language translation.
- Added Perl format quality check.

- Added support for rejecting reused passwords.
- Extended toolbar for editing RTL languages.

### 4.10.9 Weblate 2.14.1

Released on May 24th 2017.

- Fixed possible error when paginating search results.
- Fixed migrations from older versions in some corner cases.
- Fixed possible CSRF on project watch and unwatch.
- The password reset no longer authenticates user.
- Fixed possible CAPTCHA bypass on forgotten password.

### 4.10.10 Weblate 2.14

Released on May 17th 2017.

- Add glossary entries using AJAX.
- The logout now uses POST to avoid CSRF.
- The API key token reset now uses POST to avoid CSRF.
- Weblate sets Content-Security-Policy by default.
- The local editor URL is validated to avoid self-XSS.
- The password is now validated against common flaws by default.
- Notify users about important activity with their account such as password change.
- The CSV exports now escape potential formulas.
- Various minor improvements in security.
- The authentication attempts are now rate limited.
- Suggestion content is stored in the history.
- Store important account activity in audit log.
- Ask for password confirmation when removing account or adding new associations.
- Show time when suggestion has been made.
- There is new quality check for trailing semicolon.
- Ensure that search links can be shared.
- Included source string information and screenshots in the API.
- Allow to overwrite translations through API upload.

### 4.10.11 Weblate 2.13.1

Released on Apr 12th 2017.

- Fixed listing of managed projects in profile.
- Fixed migration issue where some permissions were missing.
- Fixed listing of current file format in translation download.
- Return HTTP 404 when trying to access project where user lacks privileges.

### 4.10.12 Weblate 2.13

Released on Apr 12th 2017.

- Fixed quality checks on translation templates.
- Added quality check to trigger on losing translation.
- Add option to view pending suggestions from user.
- Add option to automatically build component lists.
- Default dashboard for unauthenticated users can be configured.
- Add option to browse 25 random strings for review.
- History now indicates string change.
- Better error reporting when adding new translation.
- Added per language search within project.
- Group ACLs can now be limited to certain permissions.
- The per project ACLs are now implemented using Group ACL.
- Added more fine grained privileges control.
- Various minor UI improvements.

### 4.10.13 Weblate 2.12

Released on Mar 3rd 2017.

- Improved admin interface for groups.
- Added support for Yandex Translate API.
- Improved speed of site wide search.
- Added project and component wide search.
- Added project and component wide search and replace.
- Improved rendering of inconsistent translations.
- Added support for opening source files in local editor.
- Added support for configuring visual keyboard with special characters.
- Improved screenshot management with OCR support for matching source strings.
- Default commit message now includes translation information and URL.
- Added support for Joomla translation format.
- Improved reliability of import across file formats.

#### 4.10.14 Weblate 2.11

Released on Jan 31st 2017.

- Include language detailed information on language page.
- Mercurial backend improvements.
- Added option to specify translation component priority.
- More consistent usage of Group ACL even with less used permissions.
- Added WL\_BRANCH variable to hook scripts.
- Improved developer documentation.
- Better compatibility with various Git versions in Git exporter addon.
- Included per project and component stats.
- Added language code mapping for better support of Microsoft Translate API.
- Moved fulltext cleanup to background job to make translation removal faster.
- Fixed displaying of plural source for languages with single plural form.
- Improved error handling in import\_project.
- Various performance improvements.

#### 4.10.15 Weblate 2.10.1

Released on Jan 20th 2017.

- Do not leak account existence on password reset form (CVE-2017-5537).

#### 4.10.16 Weblate 2.10

Released on Dec 15th 2016.

- Added quality check to check whether plurals are translated differently.
- Fixed GitHub hooks for repositories with authentication.
- Added optional Git exporter module.
- Support for Microsoft Cognitive Services Translator API.
- Simplified project and component user interface.
- Added automatic fix to remove control characters.
- Added per language overview to project.
- Added support for CSV export.
- Added CSV download for stats.
- Added matrix view for quick overview of all translations
- Added basic API for changes and strings.
- Added support for Apertium APy server for machine translations.

### 4.10.17 Weblate 2.9

Released on Nov 4th 2016.

- Extended parameters for createadmin management command.
- Extended import\_json to be able to handle with existing components.
- Added support for YAML files.
- Project owners can now configure translation component and project details.
- Use «Watched» instead of «Subscribed» projects.
- Projects can be watched directly from project page.
- Added multi language status widget.
- Highlight secondary language if not showing source.
- Record suggestion deletion in history.
- Improved UX of languages selection in profile.
- Fixed showing whiteboard messages for component.
- Keep preferences tab selected after saving.
- Show source string comment more prominently.
- Automatically install Gettext PO merge driver for Git repositories.
- Added search and replace feature.
- Added support for uploading visual context (screenshots) for translations.

### 4.10.18 Weblate 2.8

Released on Aug 31st 2016.

- Documentation improvements.
- Переклади.
- Updated bundled javascript libraries.
- Added list\_translators management command.
- Django 1.8 is no longer supported.
- Fixed compatibility with Django 1.10.
- Added Subversion support.
- Separated XML validity check from XML mismatched tags.
- Fixed API to honor HIDE\_REPO\_CREDENTIALS settings.
- Show source change in Zen mode.
- Alt+PageUp/PageDown/Home/End now works in Zen mode as well.
- Add tooltip showing exact time of changes.
- Add option to select filters and search from translation page.
- Added UI for translation removal.
- Improved behavior when inserting placeables.
- Fixed auto locking issues in Zen mode.

### 4.10.19 Weblate 2.7

Released on Jul 10th 2016.

- Removed Google web translate machine translation.
- Improved commit message when adding translation.
- Fixed Google Translate API for Hebrew language.
- Compatibility with Mercurial 3.8.
- Added import\_json management command.
- Correct ordering of listed translations.
- Show full suggestion text, not only a diff.
- Extend API (detailed repository status, statistics, ...).
- Testsuite no longer requires network access to test repositories.

### 4.10.20 Weblate 2.6

Released on Apr 28th 2016.

- Fixed validation of components with language filter.
- Improved support for XLIFF files.
- Fixed machine translation for non English sources.
- Added REST API.
- Django 1.10 compatibility.
- Added categories to whiteboard messages.

### 4.10.21 Weblate 2.5

Released on Mar 10th 2016.

- Fixed automatic translation for project owners.
- Improved performance of commit and push operations.
- New management command to add suggestions from command line.
- Added support for merging comments on file upload.
- Added support for some GNU extensions to C printf format.
- Documentation improvements.
- Added support for generating translator credits.
- Added support for generating contributor stats.
- Site wide search can search only in one language.
- Improve quality checks for Armenian.
- Support for starting translation components without existing translations.
- Support for adding new translations in Qt TS.
- Improved support for translating PHP files.
- Performance improvements for quality checks.
- Виправлено загальний пошук за сайтом для рядків, які не пройшли перевірок.

- Added option to specify source language.
- Improved support for XLIFF files.
- Extended list of options for `import_project`.
- Improved targeting for whiteboard messages.
- Support for automatic translation across projects.
- Optimized fulltext search index.
- Added management command for auto translation.
- Added placeables highlighting.
- Added keyboard shortcuts for placeables, checks and machine translations.
- Improved translation locking.
- Added quality check for AngularJS interpolation.
- Added extensive group based ACLs.
- Clarified terminology on strings needing review (formerly fuzzy).
- Clarified terminology on strings needing action and not translated strings.
- Support for Python 3.
- Dropped support for Django 1.7.
- Dropped dependency on msginit for creating new gettext PO files.
- Added configurable dashboard views.
- Improved notifications on parse errors.
- Added option to import components with duplicate name to `import_project`.
- Improved support for translating PHP files
- Added XLIFF export for dictionary.
- Added XLIFF and gettext PO export for all translations.
- Documentation improvements.
- Added support for configurable automatic group assignments.
- Improved adding of new translations.

#### 4.10.22 Weblate 2.4

Released on Sep 20th 2015.

- Improved support for PHP files.
- Ability to add ACL to anonymous user.
- Improved configurability of `import_project` command.
- Added CSV dump of history.
- Avoid copy/paste errors with whitespace characters.
- Added support for Bitbucket webhooks.
- Tighter control on fuzzy strings on translation upload.
- Several URLs have changed, you might have to update your bookmarks.
- Hook scripts are executed with VCS root as current directory.
- Hook scripts are executed with environment variables describing current component.



- Add management command to optimize fulltext index.
- Added support for error reporting to Rollbar.
- Projects now can have multiple owners.
- Project owners can manage themselves.
- Added support for `javascript-format` used in gettext PO.
- Support for adding new translations in XLIFF.
- Improved file format autodetection.
- Extended keyboard shortcuts.
- Improved dictionary matching for several languages.
- Improved layout of most of pages.
- Support for adding words to dictionary while translating.
- Added support for filtering languages to be managed by Weblate.
- Added support for translating and importing CSV files.
- Rewritten handling of static files.
- Direct login/registration links to third-party service if that's the only one.
- Commit pending changes on account removal.
- Add management command to change site name.
- Add option to configure default committer.
- Add hook after adding new translation.
- Add option to specify multiple files to add to commit.

### 4.10.23 Weblate 2.3

Released on May 22nd 2015.

- Dropped support for Django 1.6 and South migrations.
- Support for adding new translations when using Java Property files
- Allow to accept suggestion without editing.
- Improved support for Google OAuth 2.0
- Added support for Microsoft .resx files.
- Tuned default robots.txt to disallow big crawling of translations.
- Simplified workflow for accepting suggestions.
- Added project owners who always receive important notifications.
- Allow to disable editing of monolingual template.
- More detailed repository status view.
- Direct link for editing template when changing translation.
- Allow to add more permissions to project owners.
- Allow to show secondary language in Zen mode.
- Support for hiding source string in favor of secondary language.

### 4.10.24 Weblate 2.2

Released on Feb 19th 2015.

- Поліпшення у швидкодії.
- Fulltext search on location and comments fields.
- New SVG/javascript based activity charts.
- Support for Django 1.8.
- Support for deleting comments.
- Added own SVG badge.
- Added support for Google Analytics.
- Improved handling of translation filenames.
- Added support for monolingual JSON translations.
- Record component locking in a history.
- Support for editing source (template) language for monolingual translations.
- Added basic support for Gerrit.

### 4.10.25 Weblate 2.1

Released on Dec 5th 2014.

- Added support for Mercurial repositories.
- Replaced Glyphicon font by Awesome.
- Added icons for social authentication services.
- Better consistency of button colors and icons.
- Documentation improvements.
- Різні виправлення помилок.
- Automatic hiding of columns in translation listing for small screens.
- Changed configuration of filesystem paths.
- Improved SSH keys handling and storage.
- Improved repository locking.
- Customizable quality checks per source string.
- Allow to hide completed translations from dashboard.

### 4.10.26 Weblate 2.0

Released on Nov 6th 2014.

- New responsive UI using Bootstrap.
- Rewritten VCS backend.
- Documentation improvements.
- Added whiteboard for site wide messages.
- Configurable strings priority.
- Added support for JSON file format.

- Fixed generating mo files in certain cases.
- Added support for GitLab notifications.
- Added support for disabling translation suggestions.
- Django 1.7 support.
- ACL projects now have user management.
- Extended search possibilities.
- Give more hints to translators about plurals.
- Fixed Git repository locking.
- Compatibility with older Git versions.
- Improved ACL support.
- Added buttons for per language quotes and other special characters.
- Support for exporting stats as JSONP.

## 4.11 Weblate 1.x series

### 4.11.1 Weblate 1.9

Released on May 6th 2014.

- Django 1.6 compatibility.
- No longer maintained compatibility with Django 1.4.
- Management commands for locking/unlocking translations.
- Improved support for Qt TS files.
- Users can now delete their account.
- Avatars can be disabled.
- Merged first and last name attributes.
- Avatars are now fetched and cached server side.
- Added support for shields.io badge.

### 4.11.2 Weblate 1.8

Released on November 7th 2013.

- Please check manual for upgrade instructions.
- Nicer listing of project summary.
- Better visible options for sharing.
- More control over anonymous users privileges.
- Supports login using third party services, check manual for more details.
- Users can login by e-mail instead of username.
- Documentation improvements.
- Improved source strings review.
- Searching across all strings.

- Better tracking of source strings.
- Captcha protection for registration.

### 4.11.3 Weblate 1.7

Released on October 7th 2013.

- Please check manual for upgrade instructions.
- Support for checking Python brace format string.
- Per component customization of quality checks.
- Detailed per translation stats.
- Changed way of linking suggestions, checks and comments to strings.
- Users can now add text to commit message.
- Support for subscribing on new language requests.
- Support for adding new translations.
- Widgets and charts are now rendered using Pillow instead of Pango + Cairo.
- Add status badge widget.
- Dropped invalid text direction check.
- Changes in dictionary are now logged in history.
- Performance improvements for translating view.

### 4.11.4 Weblate 1.6

Released on July 25th 2013.

- Nicer error handling on registration.
- Browsing of changes.
- Fixed sorting of machine translation suggestions.
- Improved support for MyMemory machine translation.
- Added support for Amagama machine translation.
- Various optimizations on frequently used pages.
- Highlights searched phrase in search results.
- Support for automatic fixups while saving the message.
- Tracking of translation history and option to revert it.
- Added support for Google Translate API.
- Added support for managing SSH host keys.
- Various form validation improvements.
- Various quality checks improvements.
- Performance improvements for import.
- Added support for voting on suggestions.
- Cleanup of admin interface.

### 4.11.5 Weblate 1.5

Released on April 16th 2013.

- Please check manual for upgrade instructions.
- Added public user pages.
- Better naming of plural forms.
- Added support for TBX export of glossary.
- Added support for Bitbucket notifications.
- Activity charts are now available for each translation, language or user.
- Extended options of import\_project admin command.
- Compatible with Django 1.5.
- Avatars are now shown using libavatar.
- Added possibility to pretty print JSON export.
- Various performance improvements.
- Indicate failing checks or fuzzy strings in progress bars for projects or languages as well.
- Added support for custom pre-commit hooks and committing additional files.
- Rewritten search for better performance and user experience.
- New interface for machine translations.
- Added support for monolingual po files.
- Extend amount of cached metadata to improve speed of various searches.
- Now shows word counts as well.

### 4.11.6 Weblate 1.4

Released on January 23rd 2013.

- Fixed deleting of checks/comments on string deletion.
- Added option to disable automatic propagation of translations.
- Added option to subscribe for merge failures.
- Correctly import on projects which needs custom ttkit loader.
- Added sitemaps to allow easier access by crawlers.
- Provide direct links to string in notification e-mails or feeds.
- Various improvements to admin interface.
- Provide hints for production setup in admin interface.
- Added per language widgets and engage page.
- Improved translation locking handling.
- Show code snippets for widgets in more variants.
- Indicate failing checks or fuzzy strings in progress bars.
- More options for formatting commit message.
- Fixed error handling with machine translation services.
- Improved automatic translation locking behaviour.

- Support for showing changes from previous source string.
- Added support for substring search.
- Various quality checks improvements.
- Support for per project ACL.
- Basic string tests coverage.

### 4.11.7 Weblate 1.3

Released on November 16th 2012.

- Compatibility with PostgreSQL database backend.
- Removes languages removed in upstream git repository.
- Improved quality checks processing.
- Added new checks (BB code, XML markup and newlines).
- Support for optional rebasing instead of merge.
- Possibility to relocate Weblate (for example to run it under /weblate path).
- Support for manually choosing file type in case autodetection fails.
- Better support for Android resources.
- Support for generating SSH key from web interface.
- More visible data exports.
- New buttons to enter some special characters.
- Support for exporting dictionary.
- Support for locking down whole Weblate installation.
- Checks for source strings and support for source strings review.
- Support for user comments for both translations and source strings.
- Better changes log tracking.
- Changes can now be monitored using RSS.
- Improved support for RTL languages.

### 4.11.8 Weblate 1.2

Released on August 14th 2012.

- Weblate now uses South for database migration, please check upgrade instructions if you are upgrading.
- Fixed minor issues with linked git repos.
- New introduction page for engaging people with translating using Weblate.
- Added widgets which can be used for promoting translation projects.
- Added option to reset repository to origin (for privileged users).
- Project or component can now be locked for translations.
- Possibility to disable some translations.
- Configurable options for adding new translations.
- Configuration of git commits per project.

- Simple antispam protection.
- Better layout of main page.
- Support for automatically pushing changes on every commit.
- Support for e-mail notifications of translators.
- List only used languages in preferences.
- Improved handling of not known languages when importing project.
- Support for locking translation by translator.
- Optionally maintain Language-Team header in po file.
- Include some statistics in about page.
- Supports (and requires) django-registration 0.8.
- Caching of counted strings with failing checks.
- Checking of requirements during setup.
- Documentation improvements.

### 4.11.9 Weblate 1.1

Released on July 4th 2012.

- Improved several translations.
- Better validation while creating component.
- Added support for shared git repositories across components.
- Do not necessary commit on every attempt to pull remote repo.
- Added support for offloading indexing.

### 4.11.10 Weblate 1.0

Released on May 10th 2012.

- Improved validation while adding/saving component.
- Experimental support for Android component files (needs patched ttkit).
- Updates from hooks are run in background.
- Improved installation instructions.
- Improved navigation in dictionary.

## 4.12 Weblate 0.x series

### 4.12.1 Weblate 0.9

Released on April 18th 2012.

- Fixed import of unknown languages.
- Improved listing of nearby messages.
- Improved several checks.
- Documentation updates.

- Added definition for several more languages.
- Various code cleanups.
- Documentation improvements.
- Changed file layout.
- Update helper scripts to Django 1.4.
- Improved navigation while translating.
- Better handling of po file renames.
- Better validation while creating component.
- Integrated full setup into syncdb.
- Added list of recent changes to all translation pages.
- Check for not translated strings ignores format string only messages.

### 4.12.2 Weblate 0.8

Released on April 3rd 2012.

- Replaced own full text search with Whoosh.
- Various fixes and improvements to checks.
- New command updatechecks.
- Lot of translation updates.
- Added dictionary for storing most frequently used terms.
- Added /admin/report/ for overview of repositories status.
- Machine translation services no longer block page loading.
- Management interface now contains also useful actions to update data.
- Records log of changes made by users.
- Ability to postpone commit to Git to generate less commits from single user.
- Possibility to browse failing checks.
- Automatic translation using already translated strings.
- New about page showing used versions.
- Django 1.4 compatibility.
- Ability to push changes to remote repo from web interface.
- Added review of translations done by others.

### 4.12.3 Weblate 0.7

Випущено 16 лютого 2012 року.

- Direct support for GitHub notifications.
- Added support for cleaning up orphaned checks and translations.
- Displays nearby strings while translating.
- Displays similar strings while translating.
- Improved searching for string.



#### 4.12.4 Weblate 0.6

Випущено 14 лютого 2012 року.

- Added various checks for translated messages.
- Tunable access control.
- Improved handling of translations with new lines.
- Added client side sorting of tables.
- Please check upgrading instructions in case you are upgrading.

#### 4.12.5 Weblate 0.5

Випущено 12 лютого 2012 року.

- **Support for machine translation using following online services:**
  - Apertium
  - Microsoft Translator
  - MyMemory
- Several new translations.
- Improved merging of upstream changes.
- Better handle concurrent git pull and translation.
- Propagating works for fuzzy changes as well.
- Propagating works also for file upload.
- Fixed file downloads while using FastCGI (and possibly others).

#### 4.12.6 Weblate 0.4

Випущено 8 лютого 2012 року.

- Added usage guide to documentation.
- Fixed API hooks not to require CSRF protection.

#### 4.12.7 Weblate 0.3

Випущено 8 лютого 2012 року.

- Better display of source for plural translations.
- New documentation in Sphinx format.
- Displays secondary languages while translating.
- Improved error page to give list of existing projects.
- New per language stats.

### 4.12.8 Weblate 0.2

Випущено 7 лютого 2012 року.

- Удосконалено перевірку коректності декількох форм.
- Warn users on profile upgrade.
- Remember URL for login.
- Іменування текстових областей при введенні форм множини.
- Автоматичне розширення області перекладу.

### 4.12.9 Weblate 0.1

Випущено 6 лютого 2012 року.

- Початковий випуск.

### W

wlc, [142](#)  
wlc.config, [143](#)  
wlc.main, [143](#)

---

## HTTP Routing Table

---

/	GET /api/projects/(string:project)/languages/, 114
ANY /, 98	GET /api/projects/(string:project)/repository/, 111
/api	GET /api/projects/(string:project)/statistics/, 114
GET /api/, 101	GET /api/roles/, 107
GET /api/changes/, 130	GET /api/roles/(int:id)/, 107
GET /api/changes/(int:id)/, 130	GET /api/screenshots/, 130
GET /api/component-lists/, 133	GET /api/screenshots/(int:id)/, 130
GET /api/component-lists/(str:slug)/, 133	GET /api/screenshots/(int:id)/file/, 131
GET /api/components/, 115	GET /api/translations/, 123
GET /api/components/(string:project)/(string:component)/, 115	GET /api/translations/(string:project)/(string:component)/changes/, 123
GET /api/components/(string:project)/(string:component)/lock/, 119	GET /api/translations/(string:project)/(string:component)/monolingual_base/, 125
GET /api/components/(string:project)/(string:component)/monolingual_base/, 119	GET /api/translations/(string:project)/(string:component)/new_template/, 126
GET /api/components/(string:project)/(string:component)/new_template/, 121	GET /api/translations/(string:project)/(string:component)/repository/, 127
GET /api/components/(string:project)/(string:component)/repository/, 121	GET /api/translations/(string:project)/(string:component)/screenshots/, 128
GET /api/components/(string:project)/(string:component)/screenshots/, 120	GET /api/units/, 129
GET /api/components/(string:project)/(string:component)/statistics/, 119	GET /api/units/(int:id)/, 129
GET /api/components/(string:project)/(string:component)/translations/, 123	GET /api/users/, 101
GET /api/components/(string:project)/(string:component)/translations/, 121	GET /api/users/(str:username)/, 101
GET /api/groups/, 104	GET /api/users/(str:username)/notifications/, 103
GET /api/groups/(int:id)/, 104	GET /api/users/(str:username)/notifications/(int:id)/, 103
GET /api/languages/, 108	POST /api/component-lists/(str:slug)/components/, 108
GET /api/languages/(string:language)/, 108	POST /api/components/(string:project)/(string:component)/, 119
GET /api/languages/(string:language)/statistics/, 109	POST /api/components/(string:project)/(string:component)/changes/, 120
GET /api/projects/, 110	POST /api/components/(string:project)/(string:component)/groups/, 104
GET /api/projects/(string:project)/, 110	POST /api/groups/(int:id)/componentlists/, 112
GET /api/projects/(string:project)/changes/, 111	
GET /api/projects/(string:project)/components/, 112	

106	106
POST /api/groups/(int:id)/components/,	DELETE /api/groups/(int:id)/projects/(int:project_id)/, 106
106	106
POST /api/groups/(int:id)/languages/,	DELETE /api/languages/(string:language)/, 109
106	109
POST /api/groups/(int:id)/projects/,	DELETE /api/projects/(string:project)/, 111
106	111
POST /api/groups/(int:id)/roles/, 105	DELETE /api/roles/(int:id)/, 108
POST /api/languages/, 108	DELETE /api/screenshots/(int:id)/, 132
POST /api/projects/, 110	DELETE /api/screenshots/(int:id)/units/(int:unit_id)/, 131
POST /api/projects/(string:project)/components/, 112	DELETE /api/translations/(string:project)/(string:component)/(string:language)/autotranslate/, 125
POST /api/projects/(string:project)/repository/, 111	DELETE /api/users/(str:username)/, 102
POST /api/roles/, 107	DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 104
POST /api/screenshots/, 131	104
POST /api/screenshots/(int:id)/file/,	PATCH /api/component-lists/(str:slug)/, 133
131	133
POST /api/screenshots/(int:id)/units/,	PATCH /api/components/(string:project)/(string:component)/(string:language)/autotranslate/, 117
131	117
POST /api/translations/(string:project)/autotranslate/, 126	PATCH /api/languages/(string:language)/, 105
POST /api/translations/(string:project)/(string:component)/(string:language)/file/, 127	109
POST /api/translations/(string:project)/(string:component)/(string:language)/repository/, 128	PATCH /api/roles/(int:id)/, 107
POST /api/translations/(string:project)/(string:component)/(string:language)/repository/, 128	PATCH /api/screenshots/(string:language)/repository/, 132
POST /api/translations/(string:project)/(string:component)/(string:language)/translations/, 126	PATCH /api/users/(str:username)/, 102
POST /api/users/, 101	PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 103
POST /api/users/(str:username)/groups/, 103	103
POST /api/users/(str:username)/notifications/(int:subscription_id)/, 103	<b>/exports</b>
PUT /api/component-lists/(str:slug)/, 133	GET /exports/rss/, 137
133	137
PUT /api/components/(string:project)/(string:component)/, 118	GET /exports/rss/(string:project)/, 137
118	137
PUT /api/groups/(int:id)/, 105	GET /exports/rss/(string:project)/(string:component)/(string:language)/, 137
PUT /api/languages/(string:language)/, 109	137
109	136
PUT /api/roles/(int:id)/, 107	<b>/hooks</b>
PUT /api/screenshots/(int:id)/, 132	GET /hooks/update/(string:project)/, 134
PUT /api/users/(str:username)/, 102	134
PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 103	GET /hooks/update/(string:project)/(string:component)/(string:language)/, 134
103	134
DELETE /api/component-lists/(str:slug)/, 133	POST /hooks/azure/, 135
133	135
DELETE /api/component-lists/(str:slug)/components/(string:project)/(string:component)/, 134	POST /hooks/bitbucket/, 135
134	135
DELETE /api/components/(string:project)/(string:component)/, 118	POST /hooks/bitbucket/, 135
118	135
DELETE /api/groups/(int:id)/, 105	POST /hooks/bitbucket/, 135
DELETE /api/groups/(int:id)/componentlists/(int:component_list_id)/, 107	POST /hooks/gitee/, 135
107	135
DELETE /api/groups/(int:id)/components/(int:component_id)/, 106	POST /hooks/gitee/, 135
106	134
DELETE /api/groups/(int:id)/languages/(string:language_code)/, 106	POST /hooks/gitlab/, 134
106	134
	POST /hooks/pagure/, 135
	135

## Symbols

```
.XML resource file
  file format, 85
--add
  auto_translate command line option, 327
--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, 326
--base-file-template TEMPLATE
  import_project command line option, 331
--check
  importusers command line option, 332
--config PATH
  wlc command line option, 139
--config-section SECTION
  wlc command line option, 139
--configuration CONFIG
  install_addon command line option, 333
--convert
  wlc command line option, 140
--email USER@EXAMPLE.COM
  createadmin command line option, 328
--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, 138
--ignore
  import_json command line option, 329
--inconsistent
  auto_translate command line option, 327
--input
  wlc command line option, 140
--key KEY
  wlc command line option, 138
--lang LANGUAGE
  loadpo command line option, 334
--language-code
  list_translators command line option, 333
--language-map LANGMAP
  import_memory command line option, 330
--language-regex REGEX
  import_project command line option, 331
--license NAME
  import_project command line option, 331
--license-url URL
  import_project command line option, 331
--main-component
  import_project command line option, 331
--main-component COMPONENT
  import_json command line option, 329
--mt MT
  auto_translate command line option, 327
--name
  createadmin command line option, 328
--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, 328
--no-privs-update
  setupgroups command line option, 335
--no-projects-update
```

- setupgroups command line option, 335
- no-update
  - setuplang command line option, 335
- output
  - wlc command line option, 140
- overwrite
  - auto\_translate command line option, 327
  - wlc command line option, 140
- password PASSWORD
  - createadmin command line option, 328
- project PROJECT
  - import\_json command line option, 329
- source PROJECT/COMPONENT
  - auto\_translate command line option, 327
- threshold THRESHOLD
  - auto\_translate command line option, 327
- update
  - createadmin command line option, 328
  - import\_json command line option, 329
  - install\_addon command line option, 333
- url URL
  - wlc command line option, 138
- user USERNAME
  - auto\_translate command line option, 327
- username USERNAME
  - createadmin command line option, 328
- vcs NAME
  - import\_project command line option, 331

## A

- add\_suggestions
  - weblate admin command, 326
- add\_suggestions command line option
  - author USER@EXAMPLE.COM, 326
- ADMINS
  - setting, 181
- AKISMET\_API\_KEY
  - setting, 286
- ALLOWED\_HOSTS
  - setting, 181
- Android
  - file format, 80
- ANONYMOUS\_USER\_NAME
  - setting, 286
- API, 98, 137, 142
- Apple strings
  - file format, 81
- ARB
  - file format, 84
- AUDITLOG\_EXPIRY
  - setting, 287
- AUTH\_LOCK\_ATTEMPTS

- setting, 287
- AUTH\_TOKEN\_VALID
  - setting, 288
- auto\_translate
  - weblate admin command, 327
- auto\_translate command line option
  - add, 327
  - inconsistent, 327
  - mt MT, 327
  - overwrite, 327
  - source PROJECT/COMPONENT, 327
  - threshold THRESHOLD, 327
  - user USERNAME, 327
- AUTO\_UPDATE
  - setting, 287
- AUTOFIX\_LIST
  - setting, 288
- AVATAR\_URL\_PREFIX
  - setting, 288

## B

- BASE\_DIR
  - setting, 289
- bilingual
  - translation, 72

## C

- celery\_queues
  - weblate admin command, 327
- changes
  - wlc command line option, 140
- CHECK\_LIST
  - setting, 289
- checkgit
  - weblate admin command, 327
- cleanup
  - wlc command line option, 139
- cleanuptrans
  - weblate admin command, 328
- Comma separated values
  - file format, 85
- Command (*клас в wlc.main*), 143
- COMMENT\_CLEANUP\_DAYS
  - setting, 290
- commit
  - wlc command line option, 139
- commit\_pending
  - weblate admin command, 328
- commit\_pending command line option
  - age HOURS, 328
- COMMIT\_PENDING\_HOURS
  - setting, 290
- commitgit
  - weblate admin command, 328
- createadmin
  - weblate admin command, 328
- createadmin command line option
  - email USER@EXAMPLE.COM, 328

- name, 328
- no-password, 328
- password PASSWORD, 328
- update, 328
- username USERNAME, 328
- CSP\_CONNECT\_SRC
  - setting, 289
- CSP\_FONT\_SRC
  - setting, 289
- CSP\_IMG\_SRC
  - setting, 289
- CSP\_SCRIPT\_SRC
  - setting, 289
- CSP\_STYLE\_SRC
  - setting, 289
- CSV
  - file format, 85

## D

- DATA\_DIR
  - setting, 290
- DATABASE\_BACKUP
  - setting, 291
- DATABASES
  - setting, 182
- DEBUG
  - setting, 182
- DEFAULT\_ACCESS\_CONTROL
  - setting, 291
- DEFAULT\_ADD\_MESSAGE
  - setting, 291
- DEFAULT\_ADDON\_MESSAGE
  - setting, 291
- DEFAULT\_ADDONS
  - setting, 291
- DEFAULT\_COMMIT\_MESSAGE
  - setting, 291
- DEFAULT\_COMMITTER\_EMAIL
  - setting, 292
- DEFAULT\_COMMITTER\_NAME
  - setting, 292
- DEFAULT\_DELETE\_MESSAGE
  - setting, 291
- DEFAULT\_FROM\_EMAIL
  - setting, 182
- DEFAULT\_MERGE\_MESSAGE
  - setting, 291
- DEFAULT\_MERGE\_STYLE
  - setting, 292
- DEFAULT\_PULL\_MESSAGE
  - setting, 293
- DEFAULT\_RESTRICTED\_COMPONENT
  - setting, 291
- DEFAULT\_TRANSLATION\_PROPAGATION
  - setting, 292
- download
  - wlc command line option, 140
- DTD

- file format, 87
- dump\_memory
  - weblate admin command, 329
- dumpuserdata
  - weblate admin command, 329

## E

- ENABLE\_AVATARS
  - setting, 293
- ENABLE\_HOOKS
  - setting, 293
- ENABLE\_HTTPS
  - setting, 293
- ENABLE\_SHARING
  - setting, 293

## F

- file format
  - .XML resource file, 85
  - Android, 80
  - Apple strings, 81
  - ARB, 84
  - Comma separated values, 85
  - CSV, 85
  - DTD, 87
  - gettext, 75
  - go-i18n, 83
  - GWT properties, 78
  - i18next, 83
  - INI translations, 78, 79
  - Java properties, 77
  - Joomla translations, 79
  - JSON, 82
  - PHP strings, 81
  - PO, 75
  - Qt, 79
  - RC, 87
  - RESX, 85
  - Ruby YAML, 86
  - Ruby YAML Ain't Markup Language, 86
  - string resources, 80
  - TS, 79
  - XLIFF, 76
  - XML, 87
  - YAML, 86
  - YAML Ain't Markup Language, 86

## G

- get() (*wlc. Weblate.memod*), 142
- gettext
  - file format, 75
- GITHUB\_USERNAME
  - setting, 294
- GITLAB\_USERNAME
  - setting, 293
- go-i18n
  - file format, 83
- GOOGLE\_ANALYTICS\_ID



setting, 294  
GWT properties  
file format, 78

## H

HIDE\_REPO\_CREDENTIALS  
setting, 294

## I

i18next  
file format, 83  
import\_demo  
weblate admin command, 329  
import\_json  
weblate admin command, 329  
import\_json command line option  
--ignore, 329  
--main-component COMPONENT, 329  
--project PROJECT, 329  
--update, 329  
import\_memory  
weblate admin command, 330  
import\_memory command line option  
--language-map LANGMAP, 330  
import\_project  
weblate admin command, 330  
import\_project command line option  
--base-file-template TEMPLATE, 331  
--file-format FORMAT, 331  
--language-regex REGEX, 331  
--license NAME, 331  
--license-url URL, 331  
--main-component, 331  
--name-template TEMPLATE, 331  
--new-base-template TEMPLATE, 331  
--vcs NAME, 331  
importuserdata  
weblate admin command, 332  
importusers  
weblate admin command, 332  
importusers command line option  
--check, 332  
INI translations  
file format, 78, 79  
install\_addon  
weblate admin command, 333  
install\_addon command line option  
--addon ADDON, 333  
--configuration CONFIG, 333  
--update, 333  
IP\_BEHIND\_REVERSE\_PROXY  
setting, 294  
IP\_PROXY\_HEADER  
setting, 294  
IP\_PROXY\_OFFSET  
setting, 295  
iPad  
translation, 81

iPhone  
translation, 81

## J

Java properties  
file format, 77  
Joomla translations  
file format, 79  
JSON  
file format, 82

## L

LEGAL\_URL  
setting, 295  
LICENSE\_EXTRA  
setting, 295  
LICENSE\_FILTER  
setting, 296  
LICENSE\_REQUIRED  
setting, 296  
LIMIT\_TRANSLATION\_LENGTH\_BY\_SOURCE\_LENGTH  
setting, 296  
list\_languages  
weblate admin command, 333  
list\_translators  
weblate admin command, 333  
list\_translators command line option  
--language-code, 333  
list\_versions  
weblate admin command, 334  
list-components  
wlc command line option, 139  
list-languages  
wlc command line option, 139  
list-projects  
wlc command line option, 139  
list-translations  
wlc command line option, 139  
load() (*wlc.config.WeblateConfig* *memod*), 143  
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, 139  
lock\_translation  
weblate admin command, 334  
lock-status  
wlc command line option, 139  
LOGIN\_REQUIRED\_URLS  
setting, 296  
LOGIN\_REQUIRED\_URLS\_EXCEPTIONS  
setting, 297

ls  
wlc command line option, 139

## M

MACHINE\_TRANSLATION\_SERVICES  
setting, 298  
main() (*в модулі wlc.main*), 143  
MATOMO\_SITE\_ID  
setting, 297  
MATOMO\_URL  
setting, 297  
monolingual  
translation, 72  
move\_language  
weblate admin command, 334  
MT\_APERTIUM\_APY  
setting, 298  
MT\_AWS\_ACCESS\_KEY\_ID  
setting, 299  
MT\_AWS\_REGION  
setting, 299  
MT\_AWS\_SECRET\_ACCESS\_KEY  
setting, 299  
MT\_BAIDU\_ID  
setting, 299  
MT\_BAIDU\_SECRET  
setting, 299  
MT\_DEEPL\_API\_VERSION  
setting, 299  
MT\_DEEPL\_KEY  
setting, 300  
MT\_GOOGLE\_CREDENTIALS  
setting, 300  
MT\_GOOGLE\_KEY  
setting, 300  
MT\_GOOGLE\_LOCATION  
setting, 300  
MT\_GOOGLE\_PROJECT  
setting, 300  
MT\_MICROSOFT\_BASE\_URL  
setting, 301  
MT\_MICROSOFT\_COGNITIVE\_KEY  
setting, 301  
MT\_MICROSOFT\_ENDPOINT\_URL  
setting, 301  
MT\_MICROSOFT\_REGION  
setting, 301  
MT\_MODERNMT\_KEY  
setting, 301  
MT\_MODERNMT\_URL  
setting, 301  
MT\_MYMEMORY\_EMAIL  
setting, 302  
MT\_MYMEMORY\_KEY  
setting, 302  
MT\_MYMEMORY\_USER  
setting, 302  
MT\_NETEASE\_KEY

setting, 302  
MT\_NETEASE\_SECRET  
setting, 302  
MT\_SAP\_BASE\_URL  
setting, 303  
MT\_SAP\_PASSWORD  
setting, 304  
MT\_SAP\_SANDBOX\_APIKEY  
setting, 303  
MT\_SAP\_USE\_MT  
setting, 304  
MT\_SAP\_USERNAME  
setting, 303  
MT\_SERVICES  
setting, 298  
MT\_TMSERVER  
setting, 302  
MT\_YANDEX\_KEY  
setting, 303  
MT\_YOUDAO\_ID  
setting, 303  
MT\_YOUDAO\_SECRET  
setting, 303

## N

NEARBY\_MESSAGES  
setting, 304

## P

PHP strings  
file format, 81  
PIWIK\_SITE\_ID  
setting, 297  
PIWIK\_URL  
setting, 297  
PO  
file format, 75  
post() (*wlc.Weblate.memod*), 142  
pull  
wlc command line option, 139  
push  
wlc command line option, 139  
pushgit  
weblate admin command, 335  
pushgit command line option  
--force-commit, 335  
Python, 142

## Q

Qt  
file format, 79

## R

RATELIMIT\_ATTEMPTS  
setting, 304  
RATELIMIT\_LOCKOUT  
setting, 304  
RATELIMIT\_WINDOW

- setting, 304
- RC
  - file format, 87
- register\_command() (в модулі wlc.main), 143
- REGISTRATION\_ALLOW\_BACKENDS
  - setting, 305
- REGISTRATION\_CAPTCHA
  - setting, 305
- REGISTRATION\_EMAIL\_MATCH
  - setting, 305
- REGISTRATION\_OPEN
  - setting, 305
- repo
  - wlc command line option, 139
- REPOSITORY\_ALERT\_THRESHOLD
  - setting, 306
- reset
  - wlc command line option, 139
- REST, 98
- RESX
  - file format, 85
- RFC
  - RFC 4646, 72
- Ruby YAML
  - file format, 86
- Ruby YAML Ain't Markup Language
  - file format, 86

## S

- SECRET\_KEY
  - setting, 182
- SENTRY\_DSN
  - setting, 306
- SERVER\_EMAIL
  - setting, 182
- SESSION\_ENGINE
  - setting, 182
- setting
  - ADMINS, 181
  - AKISMET\_API\_KEY, 286
  - ALLOWED\_HOSTS, 181
  - ANONYMOUS\_USER\_NAME, 286
  - AUDITLOG\_EXPIRY, 287
  - AUTH\_LOCK\_ATTEMPTS, 287
  - AUTH\_TOKEN\_VALID, 288
  - AUTO\_UPDATE, 287
  - AUTOFIX\_LIST, 288
  - AVATAR\_URL\_PREFIX, 288
  - BASE\_DIR, 289
  - CHECK\_LIST, 289
  - COMMENT\_CLEANUP\_DAYS, 290
  - COMMIT\_PENDING\_HOURS, 290
  - CSP\_CONNECT\_SRC, 289
  - CSP\_FONT\_SRC, 289
  - CSP\_IMG\_SRC, 289
  - CSP\_SCRIPT\_SRC, 289
  - CSP\_STYLE\_SRC, 289
  - DATA\_DIR, 290

- DATABASE\_BACKUP, 291
- DATABASES, 182
- DEBUG, 182
- DEFAULT\_ACCESS\_CONTROL, 291
- DEFAULT\_ADD\_MESSAGE, 291
- DEFAULT\_ADDON\_MESSAGE, 291
- DEFAULT\_ADDONS, 291
- DEFAULT\_COMMIT\_MESSAGE, 291
- DEFAULT\_COMMITTER\_EMAIL, 292
- DEFAULT\_COMMITTER\_NAME, 292
- DEFAULT\_DELETE\_MESSAGE, 291
- DEFAULT\_FROM\_EMAIL, 182
- DEFAULT\_MERGE\_MESSAGE, 291
- DEFAULT\_MERGE\_STYLE, 292
- DEFAULT\_PULL\_MESSAGE, 293
- DEFAULT\_RESTRICTED\_COMPONENT, 291
- DEFAULT\_TRANSLATION\_PROPAGATION, 292
- ENABLE\_AVATARS, 293
- ENABLE\_HOOKS, 293
- ENABLE\_HTTPS, 293
- ENABLE\_SHARING, 293
- GITHUB\_USERNAME, 294
- GITLAB\_USERNAME, 293
- GOOGLE\_ANALYTICS\_ID, 294
- HIDE\_REPO\_CREDENTIALS, 294
- IP\_BEHIND\_REVERSE\_PROXY, 294
- IP\_PROXY\_HEADER, 294
- IP\_PROXY\_OFFSET, 295
- LEGAL\_URL, 295
- LICENSE\_EXTRA, 295
- LICENSE\_FILTER, 296
- LICENSE\_REQUIRED, 296
- LIMIT\_TRANSLATION\_LENGTH\_BY\_SOURCE\_LENGTH, 296
- LOCALIZE\_CDN\_PATH, 296
- LOCALIZE\_CDN\_URL, 296
- LOGIN\_REQUIRED\_URLS, 296
- LOGIN\_REQUIRED\_URLS\_EXCEPTIONS, 297
- MACHINE\_TRANSLATION\_SERVICES, 298
- MATOMO\_SITE\_ID, 297
- MATOMO\_URL, 297
- MT\_APERTIUM\_API, 298
- MT\_AWS\_ACCESS\_KEY\_ID, 299
- MT\_AWS\_REGION, 299
- MT\_AWS\_SECRET\_ACCESS\_KEY, 299
- MT\_BAIDU\_ID, 299
- MT\_BAIDU\_SECRET, 299
- MT\_DEEPL\_API\_VERSION, 299
- MT\_DEEPL\_KEY, 300
- MT\_GOOGLE\_CREDENTIALS, 300
- MT\_GOOGLE\_KEY, 300
- MT\_GOOGLE\_LOCATION, 300
- MT\_GOOGLE\_PROJECT, 300
- MT\_MICROSOFT\_BASE\_URL, 301
- MT\_MICROSOFT\_COGNITIVE\_KEY, 301
- MT\_MICROSOFT\_ENDPOINT\_URL, 301
- MT\_MICROSOFT\_REGION, 301

MT\_MODERNMT\_KEY, 301  
 MT\_MODERNMT\_URL, 301  
 MT\_MYMEMORY\_EMAIL, 302  
 MT\_MYMEMORY\_KEY, 302  
 MT\_MYMEMORY\_USER, 302  
 MT\_NETEASE\_KEY, 302  
 MT\_NETEASE\_SECRET, 302  
 MT\_SAP\_BASE\_URL, 303  
 MT\_SAP\_PASSWORD, 304  
 MT\_SAP\_SANDBOX\_APIKEY, 303  
 MT\_SAP\_USE\_MT, 304  
 MT\_SAP\_USERNAME, 303  
 MT\_SERVICES, 298  
 MT\_TMSERVER, 302  
 MT\_YANDEX\_KEY, 303  
 MT\_YOUDAO\_ID, 303  
 MT\_YOUDAO\_SECRET, 303  
 NEARBY\_MESSAGES, 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  
 SECRET\_KEY, 182  
 SENTRY\_DSN, 306  
 SERVER\_EMAIL, 182  
 SESSION\_ENGINE, 182  
 SIMPLIFY\_LANGUAGES, 306  
 SINGLE\_PROJECT, 307  
 SITE\_DOMAIN, 306  
 SITE\_TITLE, 307  
 SPECIAL\_CHARS, 307  
 STATUS\_URL, 307  
 SUGGESTION\_CLEANUP\_DAYS, 308  
 URL\_PREFIX, 308  
 VCS\_BACKENDS, 308  
 VCS\_CLONE\_DEPTH, 308  
 WEBLATE\_ADDONS, 309  
 WEBLATE\_EXPORTERS, 309  
 WEBLATE\_FORMATS, 310  
 WEBLATE\_GPG\_IDENTITY, 310  
 setupgroups  
     weblate admin command, 335  
 setupgroups command line option  
     --no-privs-update, 335  
     --no-projects-update, 335  
 setuplang  
     weblate admin command, 335  
 setuplang command line option  
     --no-update, 335  
 show  
     wlc command line option, 139  
 SIMPLIFY\_LANGUAGES

    setting, 306  
 SINGLE\_PROJECT  
     setting, 307  
 SITE\_DOMAIN  
     setting, 306  
 SITE\_TITLE  
     setting, 307  
 SPECIAL\_CHARS  
     setting, 307  
 statistics  
     wlc command line option, 139  
 STATUS\_URL  
     setting, 307  
 string resources  
     file format, 80  
 SUGGESTION\_CLEANUP\_DAYS  
     setting, 308

## T

translation  
     bilingual, 72  
     iPad, 81  
     iPhone, 81  
     monolingual, 72  
 TS  
     file format, 79

## U

unlock  
     wlc command line option, 140  
 unlock\_translation  
     weblate admin command, 335  
 updatechecks  
     weblate admin command, 336  
 updategit  
     weblate admin command, 336  
 upload  
     wlc command line option, 140  
 URL\_PREFIX  
     setting, 308

## V

VCS\_BACKENDS  
     setting, 308  
 VCS\_CLONE\_DEPTH  
     setting, 308  
 version  
     wlc command line option, 139

## W

Weblate (*клас в wlc*), 142  
 weblate admin command  
     add\_suggestions, 326  
     auto\_translate, 327  
     celery\_queues, 327  
     checkgit, 327  
     cleanuptrans, 328  
     commit\_pending, 328

- commitgit, 328
- createadmin, 328
- dump\_memory, 329
- dumpuserdata, 329
- import\_demo, 329
- import\_json, 329
- import\_memory, 330
- import\_project, 330
- importuserdata, 332
- importusers, 332
- install\_addon, 333
- list\_languages, 333
- list\_translators, 333
- list\_versions, 334
- loadpo, 334
- lock\_translation, 334
- move\_language, 334
- pushgit, 335
- setupgroups, 335
- setuplang, 335
- unlock\_translation, 335
- updatechecks, 336
- updategit, 336
- WEBLATE\_ADDONS
  - setting, 309
- WEBLATE\_ADMIN\_EMAIL, 148, 149, 153
- WEBLATE\_ADMIN\_NAME, 148, 149
- WEBLATE\_ADMIN\_PASSWORD, 145, 148, 149
- WEBLATE\_ALLOWED\_HOSTS, 148, 182, 186, 307
- WEBLATE\_EMAIL\_PORT, 158
- WEBLATE\_EMAIL\_USE\_SSL, 158
- WEBLATE\_EMAIL\_USE\_TLS, 158
- WEBLATE\_EXPORTERS
  - setting, 309
- WEBLATE\_FORMATS
  - setting, 310
- WEBLATE\_GITHUB\_USERNAME, 96
- WEBLATE\_GITLAB\_USERNAME, 97
- WEBLATE\_GPG\_IDENTITY
  - setting, 310
- WEBLATE\_SILENCED\_SYSTEM\_CHECKS, 206
- WEBLATE\_SITE\_DOMAIN, 184, 202
- WeblateConfig (клас в *wlc.config*), 143
- WeblateException, 142
- wlc, 137
  - модуль, 142
- wlc command line option
  - config PATH, 139
  - config-section SECTION, 139
  - convert, 140
  - format {csv,json,text,html}, 138
  - input, 140
  - key KEY, 138
  - output, 140
  - overwrite, 140
  - url URL, 138
- changes, 140
- cleanup, 139

- commit, 139
- download, 140
- list-components, 139
- list-languages, 139
- list-projects, 139
- list-translations, 139
- lock, 139
- lock-status, 139
- ls, 139
- pull, 139
- push, 139
- repo, 139
- reset, 139
- show, 139
- statistics, 139
- unlock, 140
- upload, 140
- version, 139
- wlc.config
  - модуль, 143
- wlc.main
  - модуль, 143

## X

- XLIFF
  - file format, 76
- XML
  - file format, 87

## Y

- YAML
  - file format, 86
- YAML Ain't Markup Language
  - file format, 86



- змінна оточення
  - CELERY\_BACKUP\_OPTIONS, 160
  - CELERY\_BEAT\_OPTIONS, 160
  - CELERY\_MAIN\_OPTIONS, 160
  - CELERY\_MEMORY\_OPTIONS, 160
  - CELERY\_NOTIFY\_OPTIONS, 160
  - CELERY\_TRANSLATE\_OPTIONS, 160
  - POSTGRES\_DATABASE, 156
  - POSTGRES\_HOST, 156
  - POSTGRES\_PASSWORD, 156
  - POSTGRES\_PORT, 156
  - POSTGRES\_SSL\_MODE, 156
  - POSTGRES\_USER, 156
  - REDIS\_DB, 157
  - REDIS\_HOST, 157
  - REDIS\_PASSWORD, 157
  - REDIS\_PORT, 157
  - REDIS\_TLS, 157
  - REDIS\_VERIFY\_SSL, 157
  - ROLLBAR\_ENVIRONMENT, 159
  - ROLLBAR\_KEY, 159
  - SENTRY\_DSN, 159

- SENTRY\_ENVIRONMENT, 159
- SOCIAL\_AUTH\_SLACK\_SECRET, 156
- UWSGI\_WORKERS, 160
- WEBLATE\_ADD\_ADDONS, 159
- WEBLATE\_ADD\_APPS, 159
- WEBLATE\_ADD\_AUTOFIX, 159
- WEBLATE\_ADD\_CHECK, 159
- WEBLATE\_ADD\_LOGIN\_REQUIRED\_URLS\_EXCEPTIONS, 151
- WEBLATE\_ADMIN\_EMAIL, 148, 149, 153
- WEBLATE\_ADMIN\_NAME, 148, 149
- WEBLATE\_ADMIN\_PASSWORD, 145, 148, 149
- WEBLATE\_AKISMET\_API\_KEY, 152
- WEBLATE\_ALLOWED\_HOSTS, 148, 149, 182, 186, 307
- WEBLATE\_AUTH\_LDAP\_BIND\_DN, 153
- WEBLATE\_AUTH\_LDAP\_BIND\_PASSWORD, 153
- WEBLATE\_AUTH\_LDAP\_CONNECTION\_OPTION\_REFERENCE, 153
- WEBLATE\_AUTH\_LDAP\_SERVER\_URI, 153
- WEBLATE\_AUTH\_LDAP\_USER\_ATTR\_MAP, 153
- WEBLATE\_AUTH\_LDAP\_USER\_DN\_TEMPLATE, 153
- WEBLATE\_AUTH\_LDAP\_USER\_SEARCH, 153
- WEBLATE\_AUTH\_LDAP\_USER\_SEARCH\_FILTER, 153
- WEBLATE\_CSP\_CONNECT\_SRC, 152
- WEBLATE\_CSP\_FONT\_SRC, 152
- WEBLATE\_CSP\_IMG\_SRC, 152
- WEBLATE\_CSP\_SCRIPT\_SRC, 152
- WEBLATE\_CSP\_STYLE\_SRC, 152
- WEBLATE\_DATABASE\_BACKUP, 157
- WEBLATE\_DEBUG, 148
- WEBLATE\_DEFAULT\_ACCESS\_CONTROL, 151
- WEBLATE\_DEFAULT\_FROM\_EMAIL, 149
- WEBLATE\_DEFAULT\_RESTRICTED\_COMPONENT, 151
- WEBLATE\_DEFAULT\_TRANSLATION\_PROPAGATION, 151
- WEBLATE\_EMAIL\_BACKEND, 158
- WEBLATE\_EMAIL\_HOST, 158
- WEBLATE\_EMAIL\_HOST\_PASSWORD, 158
- WEBLATE\_EMAIL\_HOST\_USER, 158
- WEBLATE\_EMAIL\_PORT, 158
- WEBLATE\_EMAIL\_USE\_SSL, 158
- WEBLATE\_EMAIL\_USE\_TLS, 158
- WEBLATE\_ENABLE\_HTTPS, 150
- WEBLATE\_GITHUB\_USERNAME, 96, 151
- WEBLATE\_GITLAB\_HOST, 151
- WEBLATE\_GITLAB\_TOKEN, 151
- WEBLATE\_GITLAB\_USERNAME, 97, 151
- WEBLATE\_GOOGLE\_ANALYTICS\_ID, 151
- WEBLATE\_GPG\_IDENTITY, 152
- WEBLATE\_IP\_PROXY\_HEADER, 150
- WEBLATE\_LOGIN\_REQUIRED\_URLS\_EXCEPTIONS, 151
- WEBLATE\_LOGLEVEL, 148
- WEBLATE\_MT\_AWS\_ACCESS\_KEY\_ID, 152
- WEBLATE\_MT\_AWS\_REGION, 152
- WEBLATE\_MT\_AWS\_SECRET\_ACCESS\_KEY, 152
- WEBLATE\_MT\_DEEPL\_API\_VERSION, 152
- WEBLATE\_MT\_DEEPL\_KEY, 152
- WEBLATE\_MT\_GLOSBE\_ENABLED, 153
- WEBLATE\_MT\_GOOGLE\_KEY, 152
- WEBLATE\_MT\_MICROSOFT\_BASE\_URL, 152
- WEBLATE\_MT\_MICROSOFT\_COGNITIVE\_KEY, 152
- WEBLATE\_MT\_MICROSOFT\_ENDPOINT\_URL, 152
- WEBLATE\_MT\_MICROSOFT\_TERMINOLOGY\_ENABLED, 153
- WEBLATE\_MT\_MODERNMT\_KEY, 153
- WEBLATE\_MT\_MYMEMORY\_ENABLED, 153
- WEBLATE\_MT\_SAP\_BASE\_URL, 153
- WEBLATE\_MT\_SAP\_PASSWORD, 153
- WEBLATE\_MT\_SAP\_SANDBOX\_APIKEY, 153
- WEBLATE\_MT\_SAP\_USE\_MT, 153
- WEBLATE\_MT\_SAP\_USERNAME, 153
- WEBLATE\_NO\_EMAIL\_AUTH, 156
- WEBLATE\_REGISTRATION\_ALLOW\_BACKENDS, 149
- WEBLATE\_REGISTRATION\_OPEN, 149
- WEBLATE\_REMOVE\_ADDONS, 159
- WEBLATE\_REMOVE\_APPS, 159
- WEBLATE\_REMOVE\_AUTOFIX, 159
- WEBLATE\_REMOVE\_CHECK, 159
- WEBLATE\_REMOVE\_LOGIN\_REQUIRED\_URLS\_EXCEPTIONS, 151
- WEBLATE\_REQUIRE\_LOGIN, 151
- WEBLATE\_SAML\_IDP\_ENTITY\_ID, 156
- WEBLATE\_SAML\_IDP\_URL, 156
- WEBLATE\_SAML\_IDP\_X509CERT, 156
- WEBLATE\_SECURE\_PROXY\_SSL\_HEADER, 150
- WEBLATE\_SERVER\_EMAIL, 149
- WEBLATE\_SILENCED\_SYSTEM\_CHECKS, 152, 206
- WEBLATE\_SIMPLIFY\_LANGUAGES, 151
- WEBLATE\_SITE\_DOMAIN, 148, 184, 202
- WEBLATE\_SITE\_TITLE, 148
- WEBLATE\_SOCIAL\_AUTH\_AZUREAD\_OAUTH2\_KEY, 155
- WEBLATE\_SOCIAL\_AUTH\_AZUREAD\_OAUTH2\_SECRET, 155
- WEBLATE\_SOCIAL\_AUTH\_AZUREAD\_TENANT\_OAUTH2\_KEY, 155
- WEBLATE\_SOCIAL\_AUTH\_AZUREAD\_TENANT\_OAUTH2\_SECRET, 155
- WEBLATE\_SOCIAL\_AUTH\_AZUREAD\_TENANT\_OAUTH2\_TENANT\_ID, 155
- WEBLATE\_SOCIAL\_AUTH\_BITBUCKET\_KEY, 154
- WEBLATE\_SOCIAL\_AUTH\_BITBUCKET\_SECRET, 154

154  
WEBLATE\_SOCIAL\_AUTH\_FACEBOOK\_KEY,  
154  
WEBLATE\_SOCIAL\_AUTH\_FACEBOOK\_SECRET,  
154  
WEBLATE\_SOCIAL\_AUTH\_FEDORA, 156  
WEBLATE\_SOCIAL\_AUTH\_GITHUB\_KEY, 154  
WEBLATE\_SOCIAL\_AUTH\_GITHUB\_SECRET,  
154  
WEBLATE\_SOCIAL\_AUTH\_GITLAB\_API\_URL,  
155  
WEBLATE\_SOCIAL\_AUTH\_GITLAB\_KEY, 155  
WEBLATE\_SOCIAL\_AUTH\_GITLAB\_SECRET,  
155  
WEBLATE\_SOCIAL\_AUTH\_GOOGLE\_OAUTH2\_KEY,  
155  
WEBLATE\_SOCIAL\_AUTH\_GOOGLE\_OAUTH2\_SECRET,  
155  
WEBLATE\_SOCIAL\_AUTH\_GOOGLE\_OAUTH2\_WHITELISTED\_DOMAINS,  
155  
WEBLATE\_SOCIAL\_AUTH\_GOOGLE\_OAUTH2\_WHITELISTED\_EMAILS,  
155  
WEBLATE\_SOCIAL\_AUTH\_KEYCLOAK\_ACCESS\_TOKEN\_URL,  
155  
WEBLATE\_SOCIAL\_AUTH\_KEYCLOAK\_ALGORITHM,  
155  
WEBLATE\_SOCIAL\_AUTH\_KEYCLOAK\_AUTHORIZATION\_URL,  
155  
WEBLATE\_SOCIAL\_AUTH\_KEYCLOAK\_KEY,  
155  
WEBLATE\_SOCIAL\_AUTH\_KEYCLOAK\_PUBLIC\_KEY,  
155  
WEBLATE\_SOCIAL\_AUTH\_KEYCLOAK\_SECRET,  
155  
WEBLATE\_SOCIAL\_AUTH\_OPENSUSE, 156  
WEBLATE\_SOCIAL\_AUTH\_SLACK\_KEY, 156  
WEBLATE\_SOCIAL\_AUTH\_UBUNTU, 156  
WEBLATE\_TIME\_ZONE, 150  
WEBLATE\_URL\_PREFIX, 152  
WL\_BRANCH, 282  
WL\_COMPONENT\_NAME, 283  
WL\_COMPONENT\_SLUG, 283  
WL\_COMPONENT\_URL, 283  
WL\_ENGAGE\_URL, 283  
WL\_FILE\_FORMAT, 283  
WL\_FILEMASK, 283  
WL\_LANGUAGE, 283  
WL\_NEW\_BASE, 283  
WL\_PATH, 282  
WL\_PREVIOUS\_HEAD, 283  
WL\_PROJECT\_NAME, 283  
WL\_PROJECT\_SLUG, 283  
WL\_REPO, 282  
WL\_TEMPLATE, 283  
WL\_VCS, 282  
модуль  
wlc, 142  
wlc.config, 143  
wlc.main, 143