The Weblate Manual

Wydanie 4.13

Michal Čihař

15 cze 2022

Dokumenty użytkownika

1 Dokumenty użytkownika **1**
1.1 Podstawy Weblate . . . 1
1.2 Rejestracja i profil użytkownika . . . 1
1.3 Tłumaczenie za pomocą Weblate . . . 10
1.4 Pobieranie i przesyłanie tłumaczeń . . . 21
1.5 Słownik . . . 25
1.6 Kontrole i korekty . . . 27
1.7 Wyszukiwanie . . . 56
1.8 Przepływy pracy tłumaczenia . . . 63
1.9 Najczęściej zadawane pytania . . . 66
1.10 Obsługiwane formaty plików . . . 74
1.11 Integracja kontroli wersji . . . 95
1.12 REST API Weblate . . . 103
1.13 Klient Weblate . . . 151
1.14 Weblate Python API . . . 156

2 Dokumenty administratora **158**
2.1 Instrukcje konfiguracji . . . 158
2.2 Wdrożenia Weblate . . . 223
2.3 Aktualizacja Weblate . . . 224
2.4 Tworzenie kopii zapasowych i przenoszenie weblate . . . 231
2.5 Uwierzytelnienie . . . 237
2.6 Kontrola dostępu . . . 248
2.7 Projekty tłumaczeń . . . 257
2.8 Definicje języków . . . 275
2.9 Ciągła lokalizacja . . . 278
2.10 Licencjonowanie tłumaczeń . . . 287
2.11 Proces tłumaczenia . . . 289
2.12 Kontrole i korekty . . . 295
2.13 Konfigurowanie automatycznych sugestii . . . 306
2.14 Dodatki . . . 317
2.15 Pamięć tłumaczeniowa . . . 335
2.16 Konfiguracja . . . 337
2.17 Przykładowa konfiguracja . . . 362
2.18 Polecenia zarządzania . . . 378
2.19 Ogłoszenia . . . 389
2.20 Listy komponentów . . . 392
2.21 Opcjonalne moduły Weblate . . . 393
2.22 Dostosowywanie Weblate . . . 398
2.23 Interfejs zarządzania . . . 401
2.24 Uzyskaj wsparcie dla Weblate . . . 409

2.25 Dokumenty prawne . . . 412

3 Dokumenty współautora **414**
3.1 Przyczyń się do rozwoju Weblate . . . 414
3.2 Rozpoczęcie wspierania rozwoju kodu Weblate . . . 416
3.3 Kod źródłowy Weblate . . . 420
3.4 Debugowanie Weblate . . . 421
3.5 Wewnętrzna struktura Weblate . . . 423
3.6 Tworzenie dodatków . . . 424
3.7 Frontend Weblate . . . 426
3.8 Zgłaszanie problemów w Weblate . . . 427
3.9 Pakiet testowy Weblate i ciągła integracja . . . 428
3.10 Schematy danych . . . 429
3.11 Wydanie Weblate . . . 433
3.12 Bezpieczeństwo i prywatność . . . 434
3.13 Współtworzenie modułów Weblate . . . 434
3.14 O Weblate . . . 435
3.15 Licencja . . . 436

4 Historia zmian **437**
4.1 Weblate 4.13 . . . 437
4.2 Weblate 4.12.2 . . . 438
4.3 Weblate 4.12.1 . . . 438
4.4 Weblate 4.12 . . . 438
4.5 Weblate 4.11.2 . . . 439
4.6 Weblate 4.11.1 . . . 439
4.7 Weblate 4.11 . . . 439
4.8 Weblate 4.10.1 . . . 440
4.9 Weblate 4.10 . . . 440
4.10 Weblate 4.9.1 . . . 441
4.11 Weblate 4.9 . . . 441
4.12 Weblate 4.8.1 . . . 442
4.13 Weblate 4.8 . . . 442
4.14 Weblate 4.7.2 . . . 443
4.15 Weblate 4.7.1 . . . 443
4.16 Weblate 4.7 . . . 443
4.17 Weblate 4.6.2 . . . 444
4.18 Weblate 4.6.1 . . . 444
4.19 Weblate 4.6 . . . 444
4.20 Weblate 4.5.3 . . . 445
4.21 Weblate 4.5.2 . . . 445
4.22 Weblate 4.5.1 . . . 446
4.23 Weblate 4.5 . . . 446
4.24 Weblate 4.4.2 . . . 447
4.25 Weblate 4.4.1 . . . 447
4.26 Weblate 4.4 . . . 447
4.27 Weblate 4.3.2 . . . 448
4.28 Weblate 4.3.1 . . . 449
4.29 Weblate 4.3 . . . 449
4.30 Weblate 4.2.2 . . . 450
4.31 Weblate 4.2.1 . . . 450
4.32 Weblate 4.2 . . . 450
4.33 Weblate 4.1.1 . . . 451
4.34 Weblate 4.1 . . . 451
4.35 Weblate 4.0.4 . . . 453
4.36 Weblate 4.0.3 . . . 453
4.37 Weblate 4.0.2 . . . 453
4.38 Weblate 4.0.1 . . . 454

4.39 Weblate 4.0 454
4.40 Seria Weblate 3.x 455
4.41 Seria Weblate 2.x 466
4.42 Seria Weblate 1.x 477
4.43 Seria Weblate 0.x 481

Indeks modułów Pythona **485**

HTTP Routing Table **486**

Indeks **489**

ROZDZIAŁ 1

Dokumenty użytkownika

1.1 Podstawy Weblate

1.1.1 Struktura projektów i komponentów

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 *Android string resources*). 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. The translation propagation can be disabled per *Konfiguracja komponentu* using *Zezwól na propagację tłumaczenia* in case the translations should diverge.

Zobacz także:

../devel/integration

1.2 Rejestracja i profil użytkownika

1.2.1 Rejestracja

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.

You can register by following a few simple steps:

1. Fill out the registration form with your credentials.
2. Activate registration by following the link in the e-mail you receive.
3. Optionally adjust your profile to choose which languages you know.

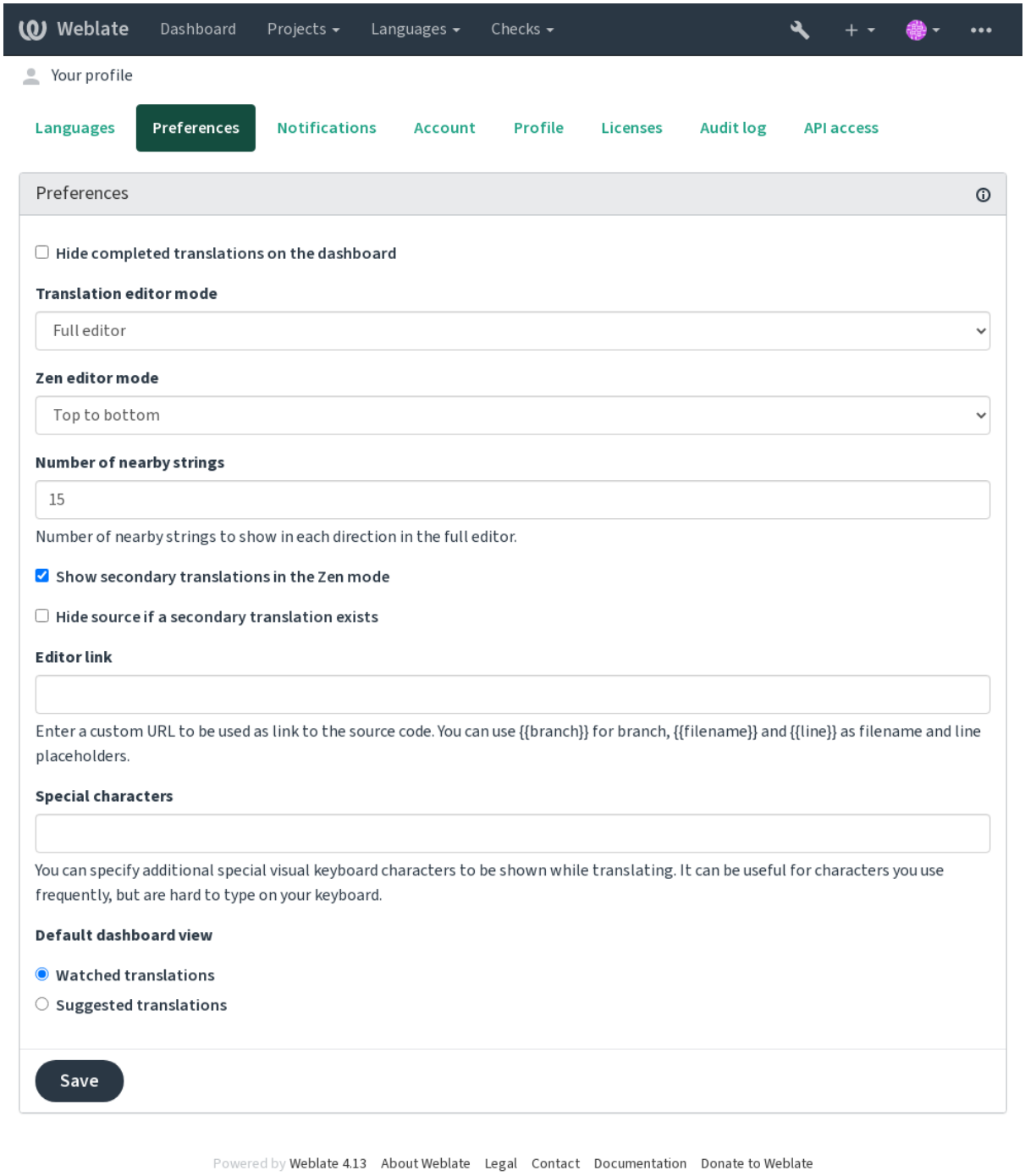
1.2.2 Kokpit

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

Nowe w wersji 2.5.

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

Podpowiedź: You can switch to different views using the navigation tabs.



The menu has these options:

- *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 *Listy komponentów*.

You can configure your personal default dashboard view in the *Preferences* section of your user profile settings.

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

1.2.3 Profil użytkownika

The user profile is accessible by clicking your user icon in the top-right of the top menu, then the *Settings* menu.

The user profile contains your preferences. Name and e-mail address is used in VCS commits, so keep this info accurate.

Informacja: All language selections only offer currently translated languages.

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

Języki

1.2.4 Język interfejsu

Wybierz język, w którym chcesz wyświetlać interfejs użytkownika.

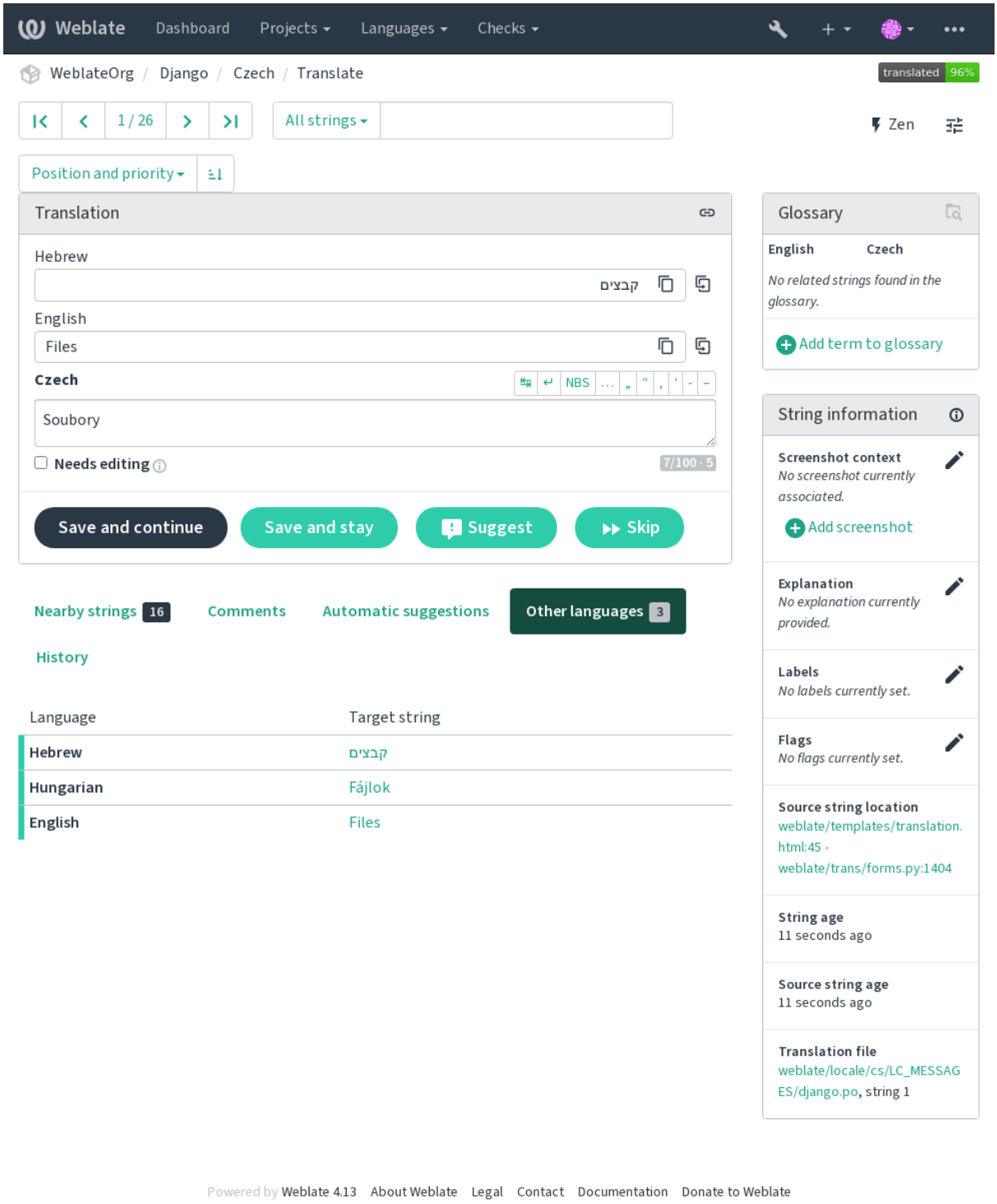
Tłumaczone języki

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.

Weblate
Dashboard
Projects
Languages
Checks
Dashboard
Watched translations 13
Suggested translations 5
Insights
Search
Translation
Translated
Unfinished
Unfinished words
Checks
Suggestions
WeblateOrg/Android — Czech
MIT
76%
3
3
WeblateOrg/Django — Hungarian
GPL-3.0
69%
8
109
1
WeblateOrg/Django — Czech
GPL-3.0
96%
1
12
4
WeblateOrg/Django — Hebrew
GPL-3.0
92%
2
15
WeblateOrg/Djangojs — Hungarian
GPL-3.0
96%
2
6
WeblateOrg/Djangojs — Hebrew
GPL-3.0
WeblateOrg/Djangojs — Czech
GPL-3.0
WeblateOrg/Language names — Czech
GPL-3.0
WeblateOrg/Language names — Hungarian
GPL-3.0
81%
4
5
WeblateOrg/Language names — Hebrew
GPL-3.0
WeblateOrg/WeblateOrg — Hungarian
GPL-3.0
100%
WeblateOrg/WeblateOrg — Czech
GPL-3.0
100%
WeblateOrg/WeblateOrg — Hebrew
GPL-3.0
100%
Powered by Weblate 4.13
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Języki dodatkowe

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:



1.2.5 Preferencje

Domyślny widok kokpitu

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.

Zobacz także:

Listy komponentów

Odnośnik edytora

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

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

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

Zobacz także:

You can find more info on registering custom URL protocols for the editor in the Nette documentation.

Znaki specjalne

Additional special characters to include in the *Klawiatura wizualna*.

1.2.6 Powiadomienia

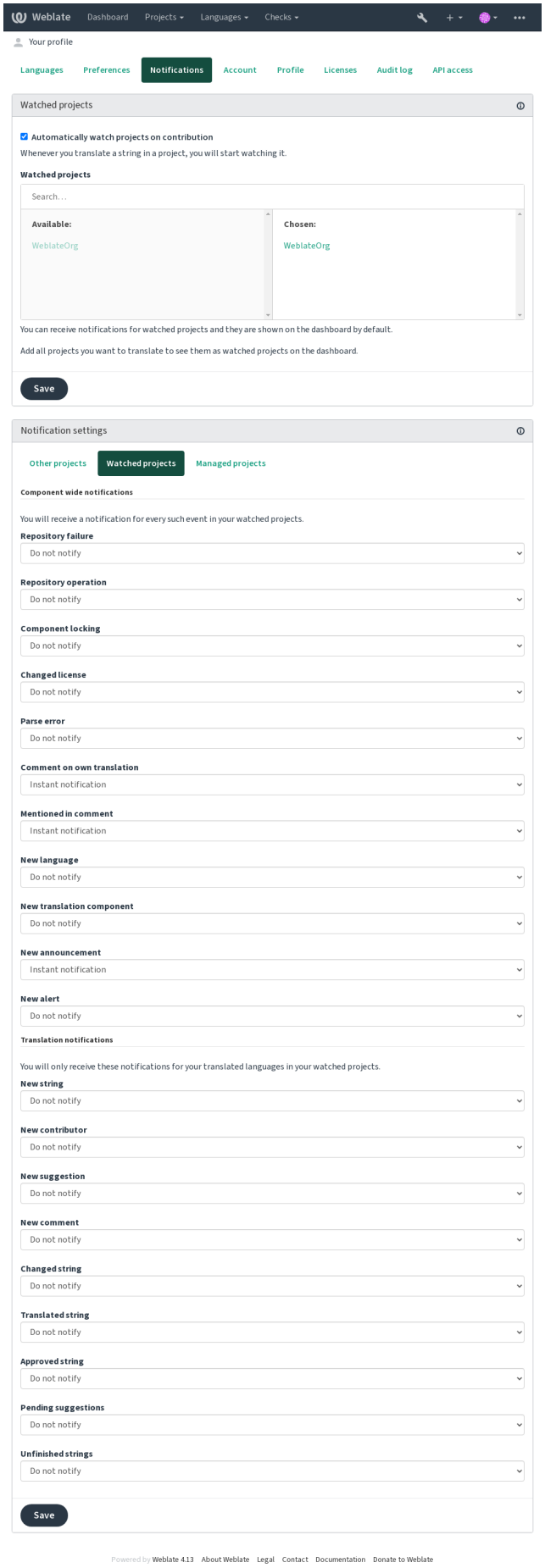
Subscribe to various notifications from the *Notifications* tab. Notifications for selected events on watched or administered projects will be sent to you per e-mail.

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 overview page and select appropriate choice from the *Watching* menu.

In case *Automatically watch projects on contribution* is enabled you will automatically start watching projects upon translating a string. The default value depends on `DEFAULT_AUTO_WATCH`.

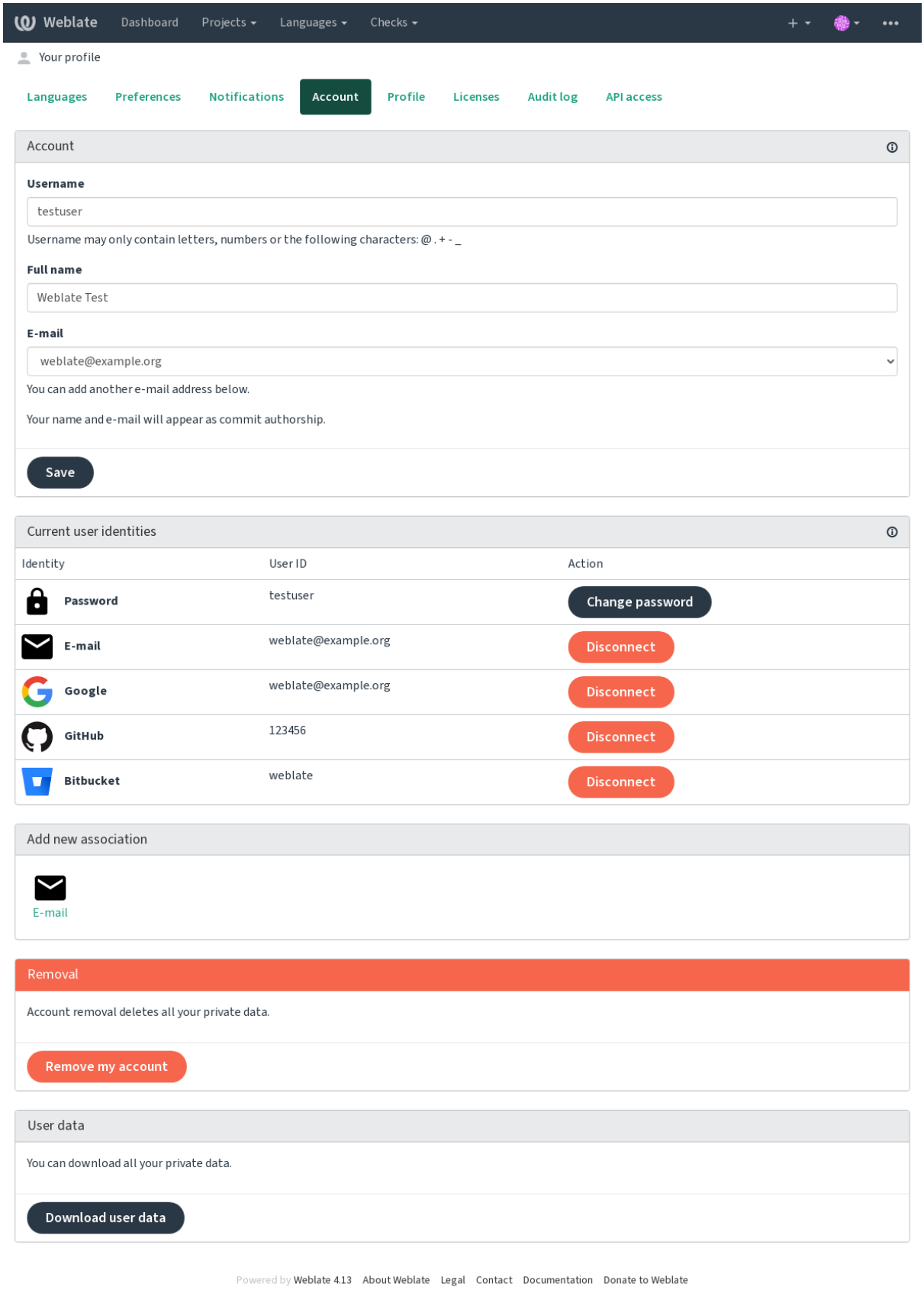
Informacja: You will not receive notifications for your own actions.


Weblate
Dashboard
Projects
Languages
Checks
Your profile
Languages
Preferences
Notifications
Account
Profile
Licenses
Audit log
API access
Watched projects
Automatically watch projects on contribution
Whenever you translate a string in a project, you will start watching it.
Watched projects
Search...
Available:
WeblateOrg
Chosen:
WeblateOrg
You can receive notifications for watched projects and they are shown on the dashboard by default.
Add all projects you want to translate to see them as watched projects on the dashboard.
Save
Notification settings
Other projects
Watched projects
Managed projects
Component wide notifications
You will receive a notification for every such event in your watched projects.
Repository failure
Do not notify
Repository operation
Do not notify
Component locking
Do not notify
Changed license
Do not notify
Parse error
Do not notify
Comment on own translation
Instant notification
Mentioned in comment
Instant notification
New language
Do not notify
New translation component
Do not notify
New announcement
Instant notification
New alert
Do not notify
Translation notifications
You will only receive these notifications for your translated languages in your watched projects.
New string
Do not notify
New contributor
Do not notify
New suggestion
Do not notify
New comment
Do not notify
Changed string
Do not notify
Translated string
Do not notify
Approved string
Do not notify
Pending suggestions
Do not notify
Unfinished strings
Do not notify
Save
Powered by Weblate 4.13
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.2.7 Konto

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 *Eksport danych użytkowników Weblate*).

Informacja: 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
Your profile
Languages
Preferences
Notifications
Account
Profile
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
Change password
E-mail
weblate@example.org
Disconnect
Google
weblate@example.org
Disconnect
GitHub
123456
Disconnect
Bitbucket
weblate
Disconnect
Add new association
E-mail
Removal
Account removal deletes all your private data.
Remove my account
User data
You can download all your private data.
Download user data
Powered by Weblate 4.13
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.2.8 Profil

Wszystkie pola na tej stronie są opcjonalne i można je usunąć w dowolnym momencie, a wypełniając je, wyrażasz zgodę na udostępnianie tych danych wszędzie tam, gdzie pojawia się Twój profil użytkownika.

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

1.2.9 Licencje

1.2.10 Dostęp do API

You can get or reset your API access token here.

1.2.11 Dziennik aktywności

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.

Zobacz także:

Działanie za odwrotnym proxy

1.3 Tłumaczenie za pomocą Weblate

Thank you for interest in translating using Weblate. Projects can either be set up for direct translation, or by way of accepting suggestions made by users without accounts.

Overall, there are two modes of translation:

- The project accepts direct translations
- The project only accepts suggestions, which are automatically validated once a defined number of votes is reached

Please see *Przepływy pracy tłumaczenia* for more info on translation workflow.

Options for translation project visibility:

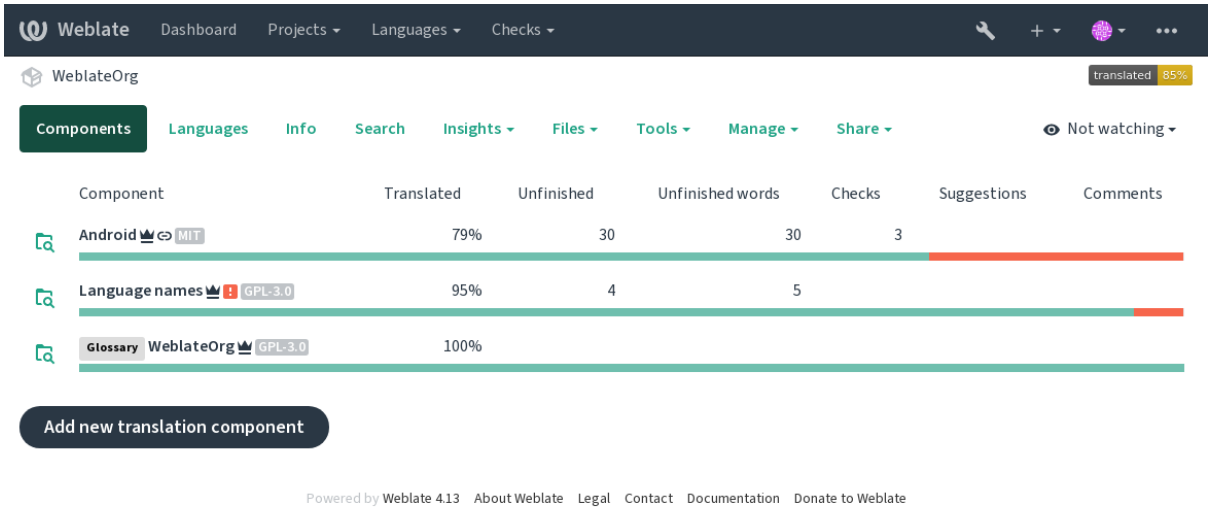
- Widoczne publicznie
- Visible only to a certain group of translators

Zobacz także:

Kontrola dostępu, *Przepływy pracy tłumaczenia*

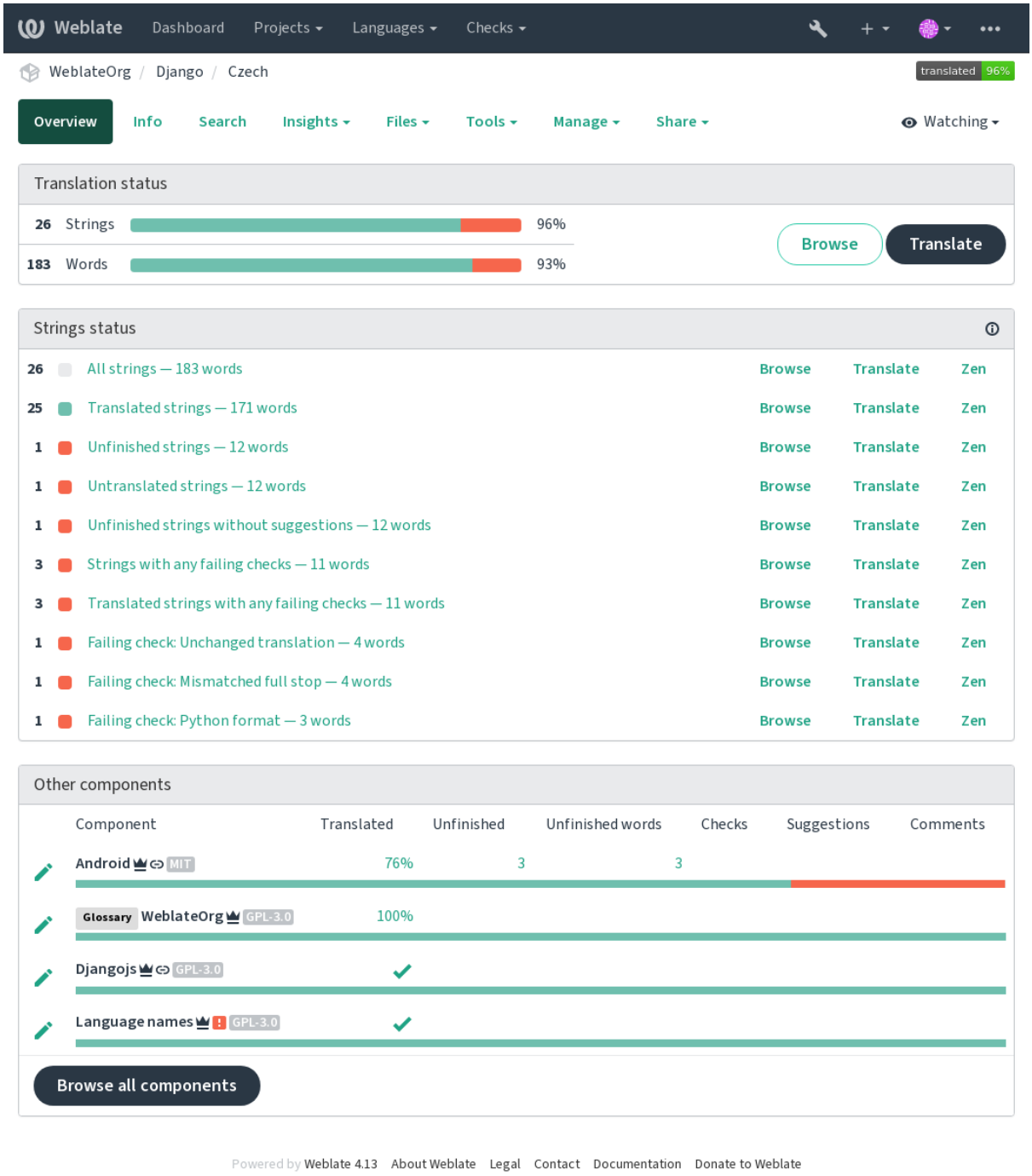
1.3.1 Projekty tłumaczeń

Translation projects hold related components; resources for the same software, book, or project.



1.3.2 Linki do tłumaczeń

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



1.3.3 Sugestie

Informacja: Actual permissions might vary depending on your Weblate configuration.

Anonymous users can only (by default) forward suggestions. Doing so is still available to signed-in users, in cases where uncertainty about the translation arises, prompting other translators to review it.

The suggestions are scanned on a daily basis to remove duplicates and suggestions matching the current translation.

1.3.4 Komentarze

Three types of comments can be posted: for translations, source strings, or to report source string bugs when this functionality is turned on using *Włącz przeglądy źródeł*. Choose the one suitable to topic you want to discuss. Source string comments are in any event good for providing feedback on the original string, for example that it should be rephrased or to ask questions about it.

Możesz używać składni Markdown we wszystkich komentarzach i wspominać innych użytkowników za pomocą `@wspomnienie`.

Zobacz także:

report-source, *Przeglądy ciągów źródłowych*, *Włącz przeglądy źródeł*

1.3.5 Warianty

Variants are used to group different length variants of the string. The frontend of your project can then use different strings depending on the screen or window size.

Zobacz także:

variants, *Warianty*

1.3.6 Etykiety

Labels are used to categorize strings within a project to further customize the localization workflow (for example to define categories of strings).

Następujące etykiety są używane przez Weblate:

Przetłumaczono automatycznie
Ciąg został przetłumaczony przy użyciu *Tłumaczenie automatyczne*.

Źródło wymaga przeglądu
Ciąg został oznaczony do przeglądu przy użyciu *Przeglądy ciągów źródłowych*.

Zobacz także:

labels

1.3.7 Tłumaczenie

On the translation page, the source string and an editing area for its translation are shown. Should the translation be plural, multiple source strings and editing areas are shown, each described and labeled in the amount of plural forms the translated language has.

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 info 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). Translation fields for any secondary languages translators select in the preferences will be shown (see *Języki dodatkowe*) above the source string.

Pod tłumaczeniem tłumacze znajdą sugestie innych osób do zaakceptowania (✓), zaakceptowania ze zmianami (✏) lub usunięcia (⍰).

Liczba mnoga

Words changing form to account of their numeric designation are called plurals. Each language has its own definition of plurals. English, for example, supports one. 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 in turn used in the translated application or project depends on the configured plural formula. Weblate shows the basic info, and the Language Plural Rules by the Unicode Consortium is a more detailed description.

Zobacz także:

Formuła liczby mnogiej

Alternative translations

Nowe w wersji 4.13.

Informacja: This is currently only supported with *Multivalue CSV file*.

With some formats, it is possible to have more translations for a single string. You can add more alternative translations using the *Tools* menu. Any blank alternative translations will be automatically removed upon saving.

Weblate Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate translated 96%

1 / 1 Custom search '%(count)s word' Zen

Position and priority

Translation

English

Singular
%(count)s word

Plural
%(count)s words

Czech, One NBS ... „ “ , ‘ - –
%(count)s slovo
15/140 · 14

Czech, Few NBS ... „ “ , ‘ - –
%(count)s slova
15/140 · 15

Czech, Other NBS ... „ “ , ‘ - –
%(count)s slov
14/140 · 15

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

Needs editing

Save and continue | Save and stay | Suggest | Skip

Nearby strings 20 | Comments | Automatic suggestions | Other languages 3

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

Glossary

English Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context
No screenshot currently associated.
Add screenshot

Explanation
No explanation currently provided.

Labels
No labels currently set.

Flags
python-format

Source string location
weblate/templates/translation.html:149

String age
6 seconds ago

Source string age
6 seconds ago

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

Powered by Weblate 4.13 About Weblate Legal Contact Documentation Donate to Weblate

Skróty klawiaturowe

Zmienione w wersji 2.18: The keyboard shortcuts have been revamped in 2.18 to less likely collide with browser or system defaults.

The following keyboard shortcuts can be utilized during translation:

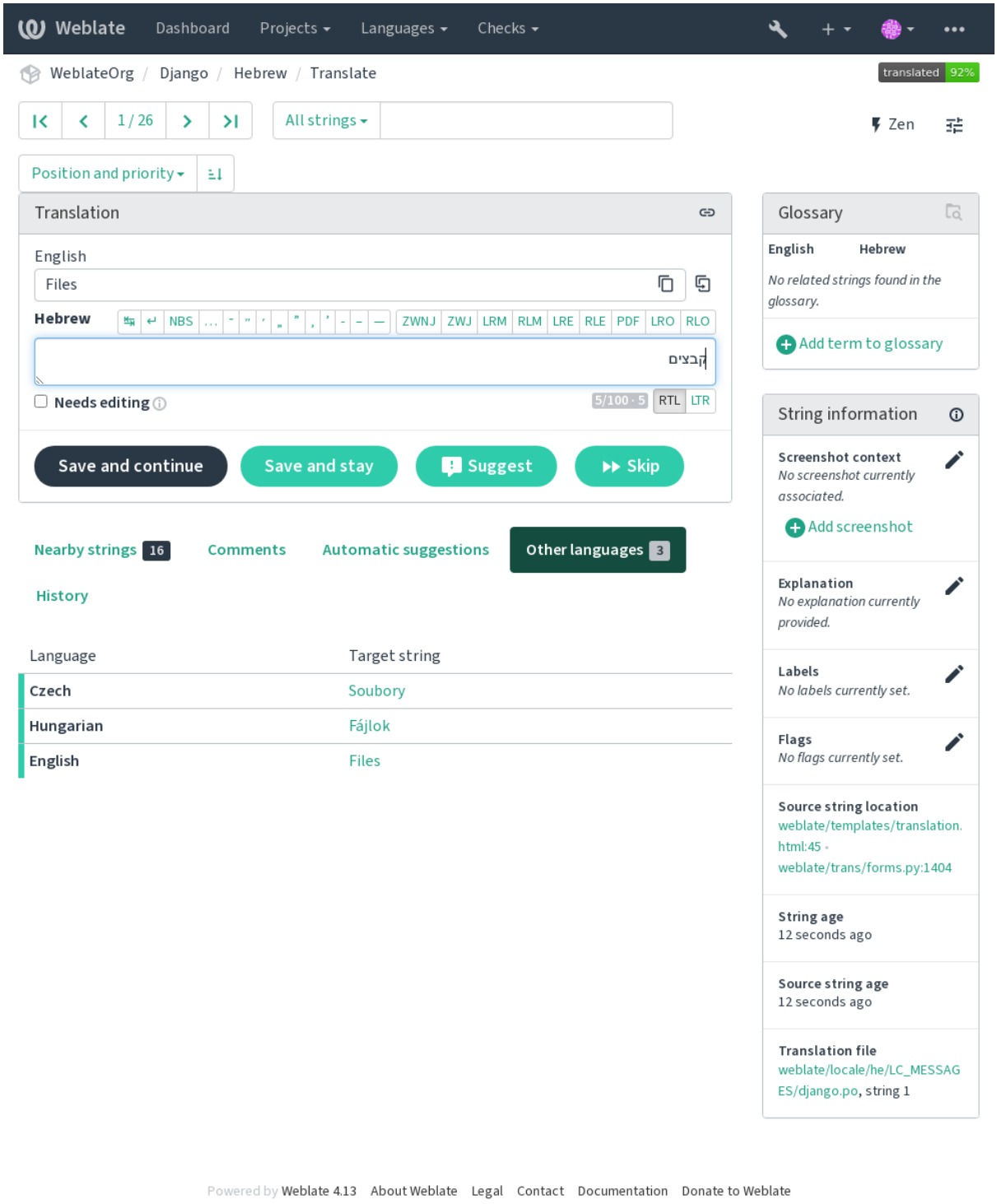
Skróty klawiszowe	Opis
`Alt+Home`	Przejdź do pierwszego tłumaczenia w bieżącym wyszukiwaniu.
`Alt+End`	Przejdź do ostatniego tłumaczenia w bieżącym wyszukiwaniu.
`Alt+PageUp` lub `Ctrl+↑` lub `Alt+↑` lub `Cmd+↑`	Przejdź do poprzedniego tłumaczenia w bieżącym wyszukiwaniu.
`Alt+PageDown` lub `Ctrl+↓` lub `Alt+↓` lub `Cmd+↓`	Przejdź do następnego tłumaczenia w bieżącym wyszukiwaniu.
`Alt+Enter` lub `Ctrl+Enter` lub `Cmd+Enter`	Submit current form; this is same as pressing *Save and continue* while editing translation.
`Ctrl+Shift+Enter` lub `Cmd+Shift+Enter`	Usuń oznaczenie tłumaczenia jako wymagającego edycji i prześlij je.
`Ctrl+E` lub `Cmd+E`	Focus translation editor.
`Ctrl+U` lub `Cmd+U`	Focus comment editor.
`Ctrl+M` lub `Cmd+M`	Shows *Automatic suggestions* tab, see *Automatyczne sugestie*.
`Ctrl+1` do `Ctrl+9` lub `Cmd+1` to `Cmd+9`	Copies placeable of given number from source string.
`Ctrl+M+1` do `9` lub `Cmd+M+1` do `9`	Copy the machine translation of given number to current translation.
`Ctrl+I+1` do `9` lub `Cmd+I+1` do `9`	Zignoruj jedną pozycję na liście testów zakończonych niepowodzeniem.
`Ctrl+J` lub `Cmd+J`	Pokazuje kartę *Ciągi sąsiadujące*.
`Ctrl+S` lub `Cmd+S`	Focus search field.
`Ctrl+O` lub `Cmd+O`	Skopiuj ciąg źródłowy.
`Ctrl+Y` lub `Cmd+Y`	Przełącz pole wyboru *Wymaga edycji*.

Klawiatura wizualna

A small visual keyboard row is shown just above the translation field. This can be useful to keep local punctuation in mind (as the row is local to each language), or have characters otherwise hard to type handy.

The shown symbols factor into three categories:

- User configured *Znaki specjalne* defined in the *Profil użytkownika*
- Per-language characters provided by Weblate (e.g. quotes or RTL specific characters)
- Characters configured using `SPECIAL_CHARS`


Weblate
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Hebrew / Translate
translated 92%
1 / 26
All strings
Zen
Position and priority
Translation
English
Files
Hebrew
NBS ... ZWNJ ZWJ LRM RLM LRE RLE PDF LRO RLO
קבצים
Needs editing
5/100 · 5
RTL LTR
Save and continue
Save and stay
Suggest
Skip
Nearby strings 16
Comments
Automatic suggestions
Other languages 3
History
Language
Target string
Czech
Soubory
Hungarian
Fájlok
English
Files
Glossary
English
Hebrew
No related strings found in the glossary.
Add term to glossary
String information
Screenshot context
No screenshot currently associated.
Add screenshot
Explanation
No explanation currently provided.
Labels
No labels currently set.
Flags
No flags currently set.
Source string location
weblate/templates/translation.html:45 · weblate/trans/forms.py:1404
String age
12 seconds ago
Source string age
12 seconds ago
Translation file
weblate/locale/he/LC_MESSAGES/django.po, string 1
Powered by Weblate 4.13
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Kontekst tłumaczenia

This contextual description provides related info about the current string.

Atrybuty ciągu
Things like message ID, context (`msgctxt`) or location in source code.

Zrzuty ekranu
Screenshots can be uploaded to Weblate to better inform translators of where and how the string is used, see *Kontekst wizualny dla ciągów*.

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

Inne wystąpienia
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 *Niespójność*). You can choose which one to use.

Pamięć tłumaczeniowa
Look at similar strings translated in past, see *Pamięć tłumaczeniowa*.

Słownik
Displays terms from the project glossary used in the current message.

Ostatnie zmiany
List of people whom have changed this message recently using Weblate.

Projekt
Project info like instructions for translators, or a directory or link to the string in the version control system repository the project uses.

If you want direct links, the translation format has to support it.

Historia tłumaczeń

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.

Translated string length

Weblate can limit the length of a translation in several ways to ensure the translated string is not too long:

- The default limitation for translation is ten times longer than the source string. This can be turned off by `LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH`. In case you are hitting this, it might be also caused by a monolingual translation erroneously set up as bilingual one, making Weblate mistaking the translation key for the actual source string. See *Formaty dwujęzyczne i jednojęzyczne* for more info.
- Maximal length in characters defined by translation file or flag, see *Maksymalna długość tłumaczenia*.
- Maximal rendered size in pixels defined by flags, see *Maksymalny rozmiar tłumaczenia*.

1.3.8 Automatyczne sugestie

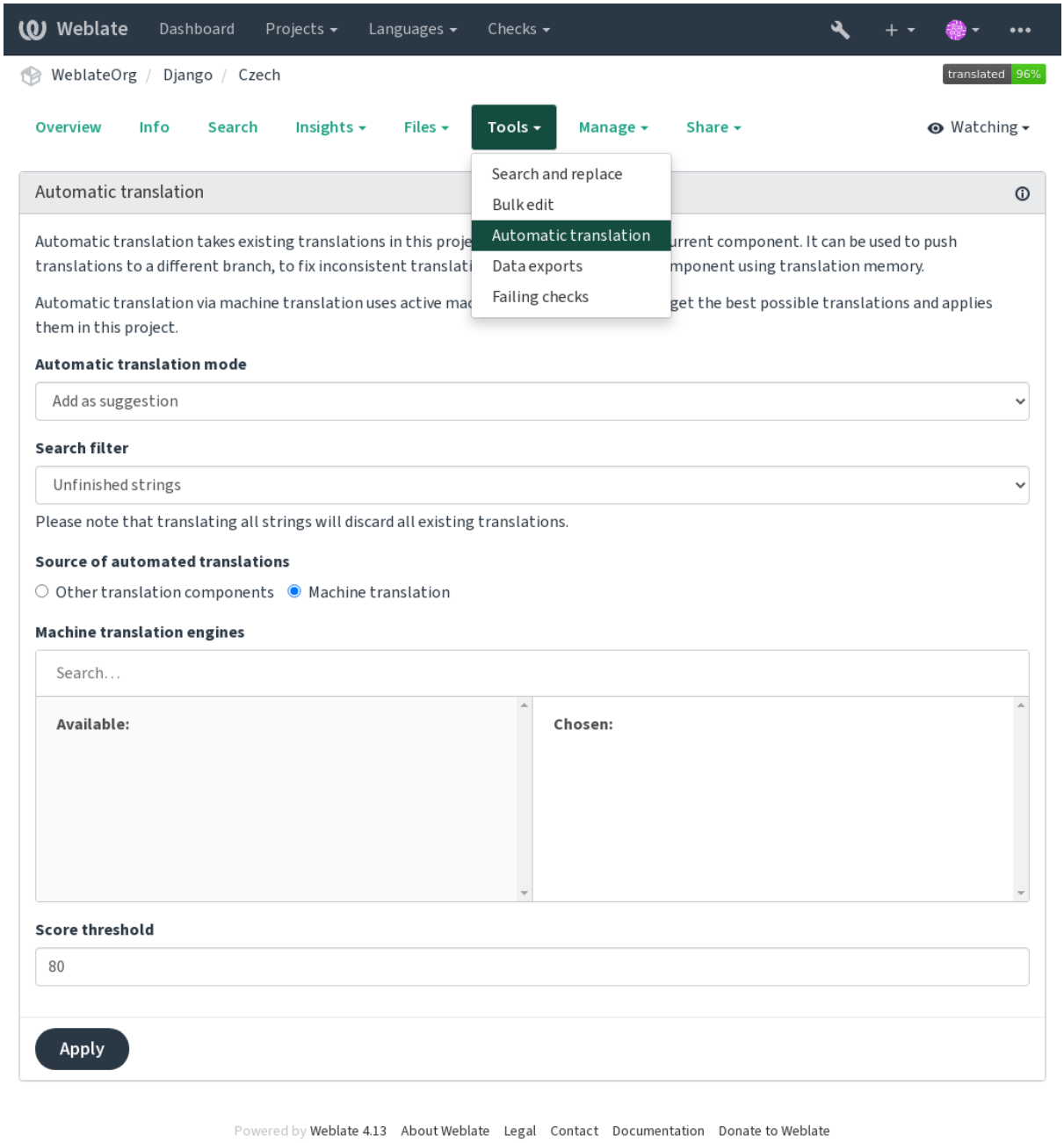
Based on configuration and your translated language, Weblate provides suggestions from several machine translation tools and *Pamięć tłumaczeniowa*. All machine translations are available in a single tab of each translation page.

Zobacz także:

You can find the list of supported tools in *Konfigurowanie automatycznych sugestii*.

1.3.9 Tłumaczenie automatyczne

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:



Two modes of operation are possible:

- Using other Weblate components as a source for translations.

- Using selected machine translation services with translations above a certain quality threshold.

You can also choose which strings are to be auto-translated.

Ostrzeżenie: 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 the application and its website) or when bootstrapping a translation for a new component using existing translations (translation memory).

Automatycznie przetłumaczone ciągi są oznaczone jako *Przetłumaczono automatycznie*.

Zobacz także:

Zachowanie takich samych tłumaczeń między komponentami

1.3.10 Ograniczenie szybkości

To avoid abuse of the interface, rate limiting is applied to several operations like searching, sending contact forms or translating. If affected by it, you are blocked for a certain period until you can perform the operation again.

Default limits and fine-tuning is described in the administrative manual, see *Ograniczenie szybkości*.

1.3.11 Wyszukaj i zamień

Change terminology effectively or perform bulk fixing of the strings using *Search and replace* in the *Tools* menu.

Podpowiedź: Nie martw się, że popsujesz ciągi znaków. Jest to dwuetapowy proces przedstawiający podgląd edytowanych ciągów przed potwierdzeniem rzeczywistej zmiany.

1.3.12 Edycja zbiorcza

Bulk editing allows performing one operation on number of strings. You define strings by searching for them and set up something to be done for matching ones. The following operations are supported:

- Changing string state (for example to approve all unreviewed strings).
- Adjust translation flags (see *Dostosowywanie zachowania za pomocą flag*)
- Adjust string labels (see labels)

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

Zobacz także:

Bulk edit add-on

1.3.13 Widok matrycy

To compare different languages efficiently you can use the matrix view. It is available on every component page under the *Tools* menu. First select all languages you want to compare and confirm your selection, after that you can click on any translation to open and edit it quickly.

The matrix view is also a very good starting point to find missing translations in different languages and quickly add them from one view.

1.3.14 Tryb zen

The Zen editor can be enabled by clicking the *Zen* button on the top right while translating a component. It simplifies the layout and removes additional UI elements such as *Nearby strings* or the *Glossary*.

You can select the Zen editor as your default editor using the *Preferencje* tab on your *Profil użytkownika*. Here you can also choose between having translations listed *Top to bottom* or *Side by side* depending on your personal preference.

1.4 Pobieranie i przesyłanie tłumaczeń

You can export files from a translation, make changes, and import them again. This allows working offline, and then merging changes back into the existing translation. This works even if it has been changed in the meantime.

Informacja: Available options might be limited by *access control* settings.

1.4.1 Pobieranie tłumaczeń

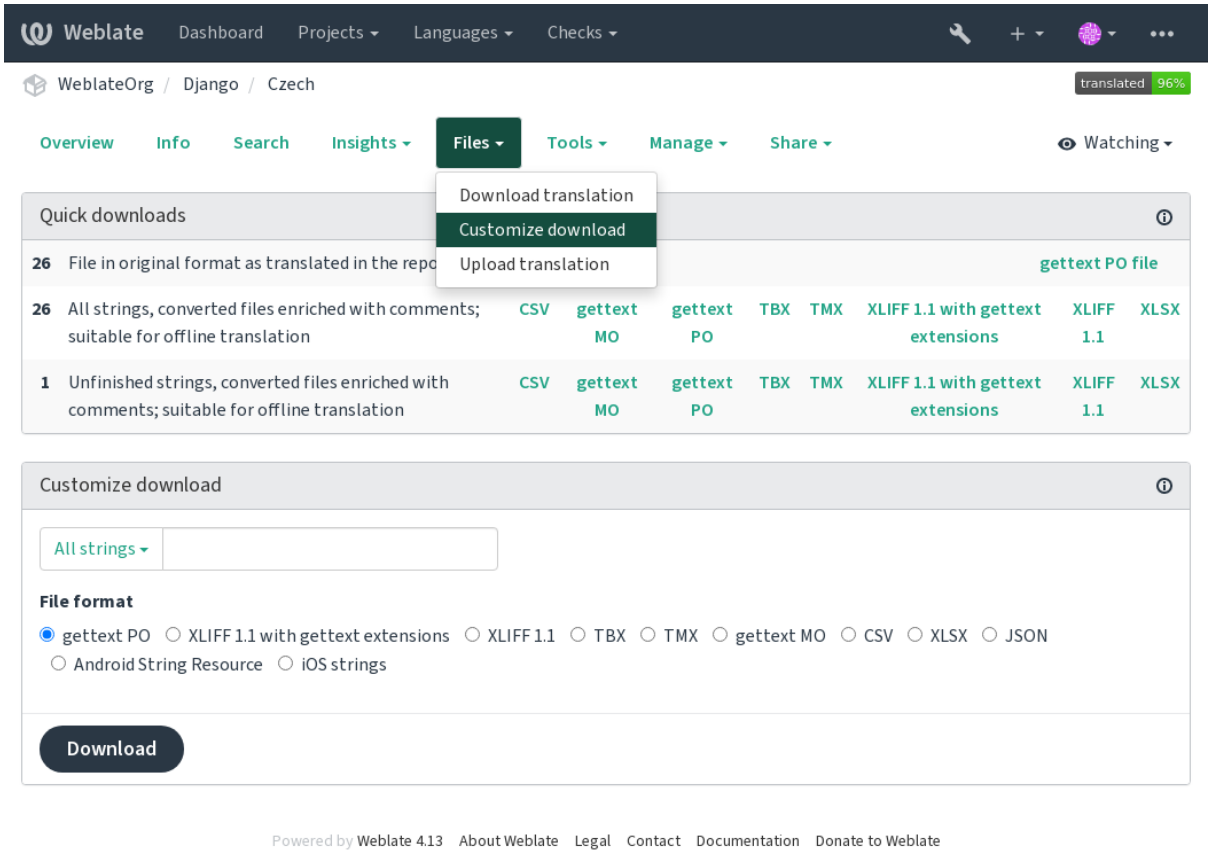
From the project or component dashboard, translatable files can be downloaded in the *Files* menu.

The first option is to download the file in the original format as it is stored in the repository. In this case, any pending changes in the translation are getting committed and the up-to-date file is yield without any conversions.

You can also download the translation converted into one of the widely used localization formats. The converted files will be enriched with data provided in Weblate; such as additional context, comments or flags. Several file formats are available via the *Files* ↓ *Customize download* menu:

- gettext PO
- XLIFF z rozszerzeniami gettext
- XLIFF 1.1
- TermBase eXchange
- Pamięć tłumaczeniowa eXchange
- gettext MO (only available when translation is using gettext PO)
- CSV
- Excel Open XML
- JSON (dostępny tylko dla tłumaczeń jednojęzycznych)
- Android String Resource (dostępne tylko dla tłumaczeń jednojęzycznych)
- Ciągi iOS (dostępne tylko dla tłumaczeń jednojęzycznych)

Podpowiedź: The content available in the converted files differs based on file format features, you can find overview in *Możliwe typy tłumaczeń*.

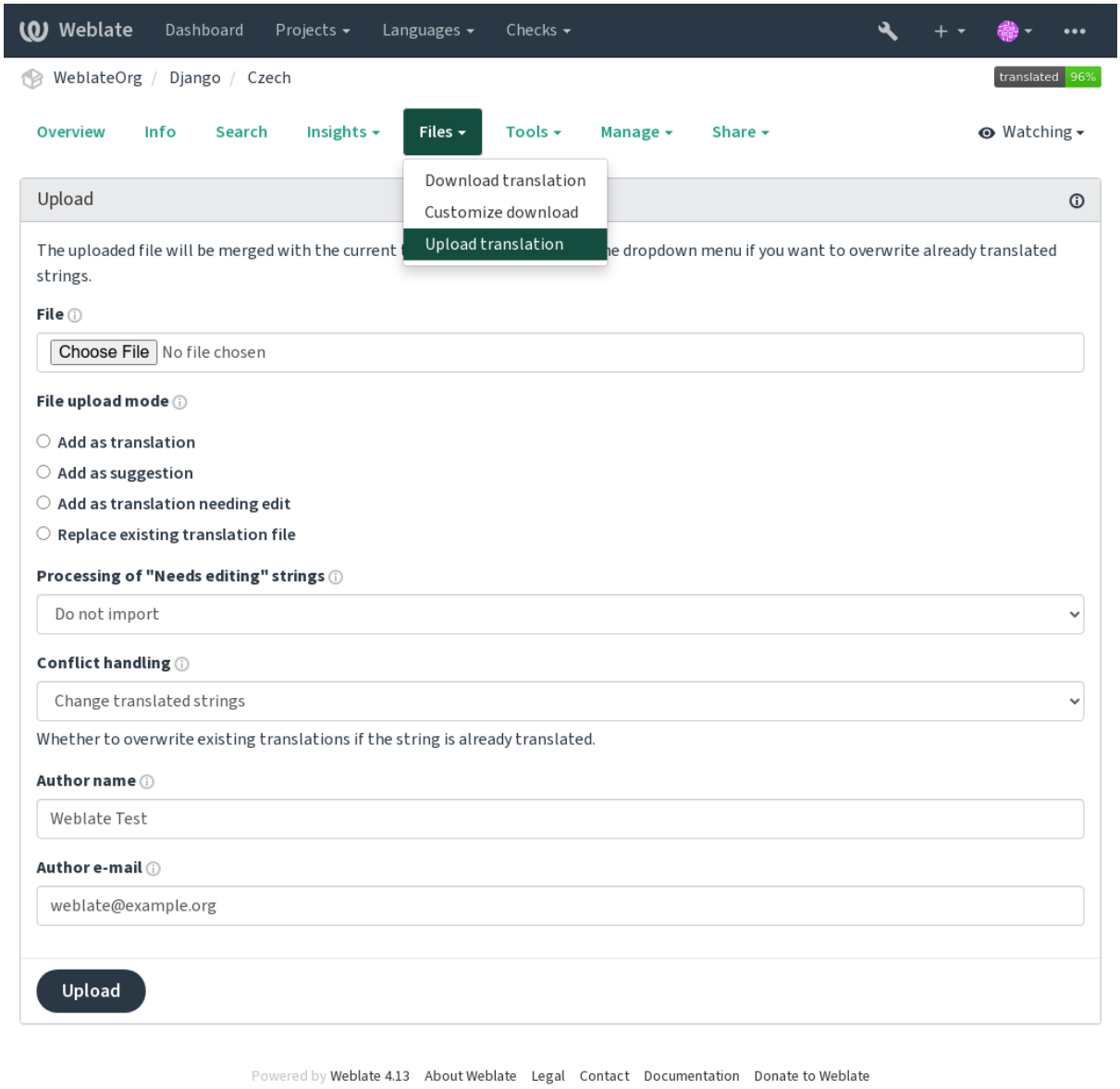


Zobacz także:

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

1.4.2 Przesyłanie tłumaczeń

When you have made your changes, use *Upload translation* in the *Files* menu.



Obsługiwane formaty plików

Any file in a supported file format can be uploaded, but it is still recommended to use the same file format as the one used for translation, otherwise some features might not be translated properly.

Zobacz także:

Obsługiwane formaty plików, *Pobieranie i przesyłanie tłumaczeń*

Metody importu

These are the choices presented when uploading translation files:

Dodaj jako tłumaczenie (`przetłumacz`)
Imported strings are added as translations to existing strings. This is the most common usecase, and the default behavior.

Only translations are used from the uploaded file and no additional content.

Dodaj jako sugestię (`suggest`)
Imported strings are added as suggestions, do this when you want to have your uploaded strings reviewed.

Only translations are used from the uploaded file and no additional content.

Dodaj jako tłumaczenie wymagające edycji (`rozmyte`)
Imported strings are added as translations needing edit. This can be useful when you want translations to be used, but also reviewed.

Only translations are used from the uploaded file and no additional content.

Zastąp istniejący plik tłumaczenia (`zamień`)
Existing file is replaced with new content. This can lead to loss of existing translations, use with caution.

Zaktualizuj ciągi źródłowe (`źródło`)
Updates source strings in bilingual translation file. This is similar to what *Zaktualizuj pliki PO, aby dopasować POT (msgmerge)* does.

Ta opcja jest obsługiwana tylko w przypadku niektórych formatów plików.

Dodaj nowe ciągi (`dodaj`)
Adds new strings to the translation. It skips the one which already exist.

In case you want to both add new strings and update existing translations, upload the file second time with *Add as translation*.

This option is available only with *Zarządzaj ciągami* turned on.

Z przesłanego pliku używane są tylko źródło, tłumaczenie i klucz (kontekst).

Zobacz także:

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

Obsługa konfliktów

Defines how to deal with uploaded strings which are already translated.

Ciągi wymagające edycji

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”.

Zastępowanie autorstwa

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 Słownik

Each project can include one or more glossaries as a shorthand for storing terminology. Glossary easify maintaining consistency of the translation.

A glossary for each language can be managed on its own, but they are stored together as a single component which helps project admins and multilingual translators to maintain some cross-language consistency as well. Terms from the glossary containing words from the currently translated string are displayed in the sidebar of the translation editor.

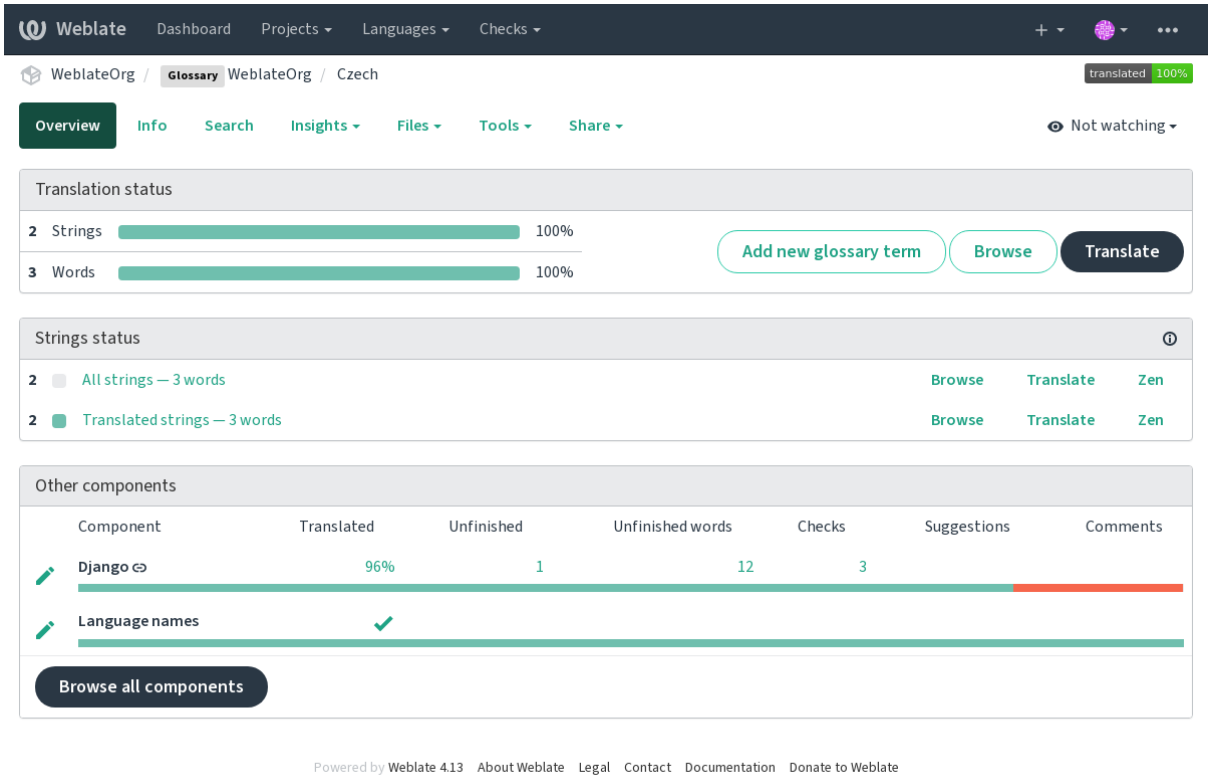
1.5.1 Zarządzanie słownikami

Zmienione w wersji 4.5: Glossaries are now regular translation components and you can use all Weblate features on them — commenting, storing in a remote repository, or adding explanations.

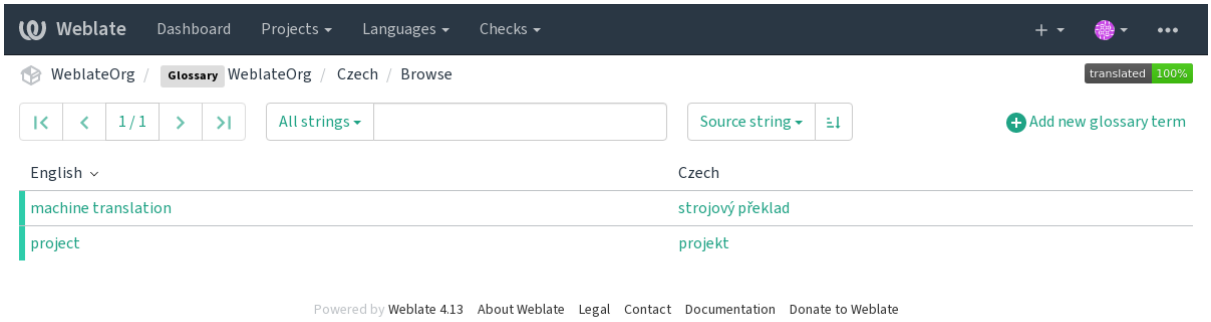
Użyj dowolnego komponentu jako słownika, włączając *Użyj jako słownika*. Możesz utworzyć wiele słowników dla jednego projektu.

An empty glossary for a given project is automatically created with the project. Glossaries are shared among all components of the same project, and optionally with other projects using *Udostępniony w projektach* from the respective glossary component.

The glossary component looks like any other component in Weblate with added colored label:



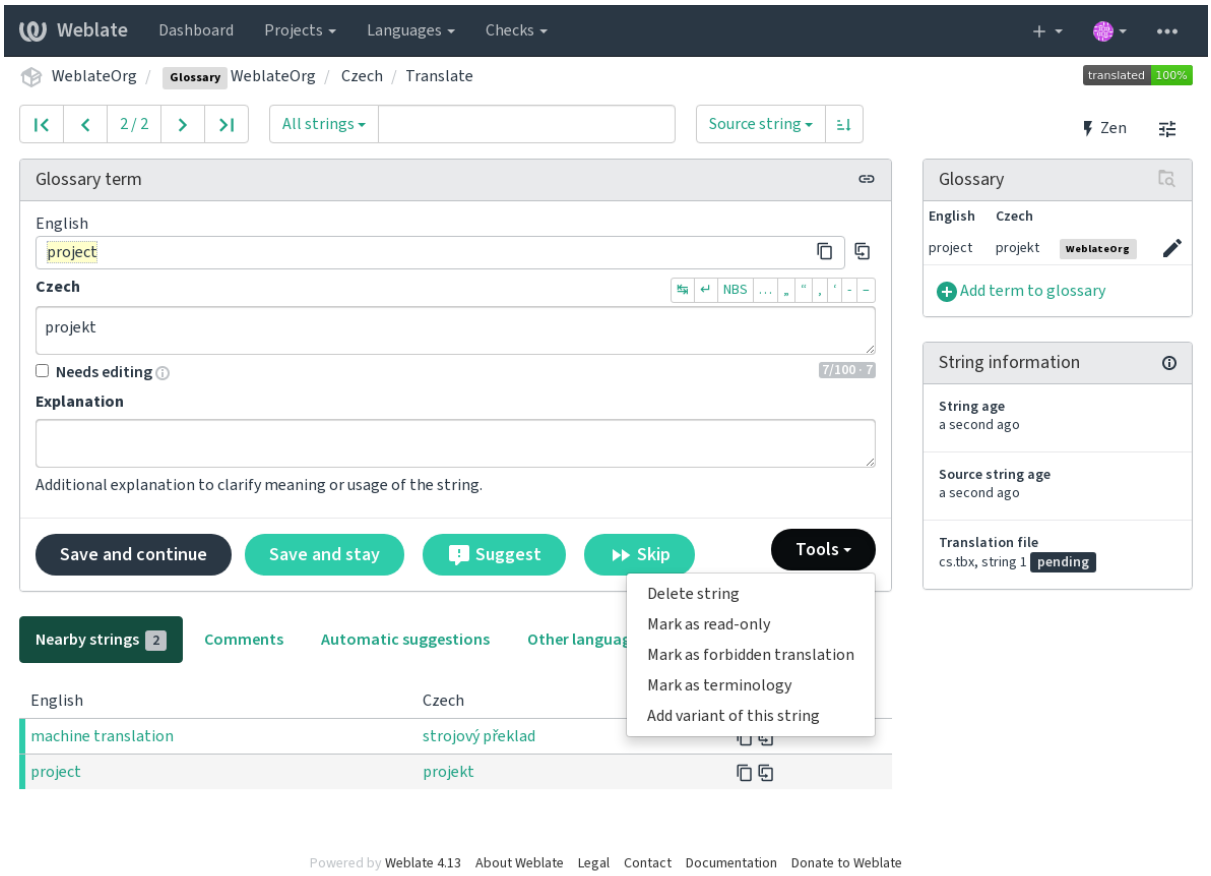
Możesz przeglądać wszystkie terminy w słowniku:



lub edytować je jako dowolne tłumaczenia.

1.5.2 Słownik terminów

Glossary terms are translated the same way regular strings are. You can toggle additional features using the *Tools* menu for each term.



Terminy nieprzetłumaczalne

Nowe w wersji 4.5.

Flagging certain glossary term translations `read-only` by bulk-editing, typing in the flag, or by using *Tools* ↓ *Mark as read-only* means they can not be translated. Use this for brand names or other terms that should not be changed in other languages. Such terms are visually highlighted in the glossary sidebar.

Zobacz także:

Dostosowywanie zachowania za pomocą flag

Zabronione tłumaczenia

Nowe w wersji 4.5.

Flagging certain glossary term translations as `forbidden`, by bulk-editing, typing in the flag, or by using *Tools* ↓ *Mark as forbidden translation* means they are **not** to be used. Use this to clarify translation when some words are ambiguous or could have unexpected meanings.

Zobacz także:

Dostosowywanie zachowania za pomocą flag

Terminologia

Nowe w wersji 4.5.

Flagging certain glossary terms as `terminology` by bulk-editing, typing in the flag, or by using *Tools* ↓ *Mark as terminology* adds entries for them to all languages in the glossary. Use this for important terms that should be well thought out, and retain a consistent meaning across all languages.

Zobacz także:

Dostosowywanie zachowania za pomocą flag

Warianty

Variants are a generic way to group strings together. All term variants are listed in the glossary sidebar when translating.

Podpowiedź: You can use this to add abbreviations or shorter expressions for a term.

Zobacz także:

variants

1.6 Kontrole i korekty

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:

Weblate Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate translated 96%

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

1 / 1 Custom search '%(count)s word' Position Zen

Translation

English

Singular
%(count)s word

Plural
%(count)s words

Czech, One

0/140 · 14

Czech, Few
několik slov

12/140 · 15

Czech, Other
%(count)s slov

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

14/140 · 15

Needs editing

Save and continue Save and stay Suggest Skip

Nearby strings 20 Comments Automatic suggestions Other languages 3 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
For all languages

Missing plurals 2
Some plural forms are untranslated
Dismiss
For all languages

Glossary

English Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context
No screenshot currently associated.
Add screenshot

Explanation
No explanation currently provided.

Labels
No labels currently set.

Flags
python-format

Source string location
weblate/templates/translation.html:149

String age
10 seconds ago

Source string age
10 seconds ago

Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5
pending

Powered by Weblate 4.13 About Weblate Legal Contact Documentation Donate to Weblate

1.6.1 Automatyczne korekty

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

Zobacz także:

`AUTOFIX_LIST`

1.6.2 Kontrole jakości

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.

Zobacz także:

`CHECK_LIST`, *Dostosowywanie zachowania za pomocą flag*

1.6.3 Sprawdzanie tłumaczeń

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

Znaczniki BBCode

Podsumowanie
BBCode w tłumaczeniu nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.markup.BBCodeCheck`

Flaga do zignorowania
`ignore-bbcode`

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.

Informacja: The method for detecting BBCode is currently quite simple so this check might produce false positives.

Kolejne zduplikowane wyrazy

Nowe w wersji 4.1.

Podsumowanie
Tekst zawiera to samo słowo dwa razy z rzędu

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.duplicate.DuplicateCheck`

Flaga do zignorowania
`ignore-duplicate`

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

Podpowiedź: This check includes language specific rules to avoid false positives. In case it triggers falsely in your case, let us know. See *Zgłaszanie problemów w Weblate*.

Nie jest zgodny z słownikiem

Nowe w wersji 4.5.

Podsumowanie
Tłumaczenie nie jest zgodne z terminami zdefiniowanymi w słowniku.

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.glossary.GlossaryCheck`

Dostępna flaga
`check-glossary`

Flaga do zignorowania
`ignore-check-glossary`

This check has to be turned on using `check-glossary` flag (see *Dostosowywanie zachowania za pomocą flag*). Please consider following prior to enabling it:

- It does exact string matching, the glossary is expected to contain terms in all variants.
- Checking each string against glossary is expensive, it will slow down any operation in Weblate which involves running checks like importing strings or translating.

Zobacz także:

Słownik, *Dostosowywanie zachowania za pomocą flag*, *Flagi tłumaczeń*

Podwójna spacja

Podsumowanie
Tłumaczenie zawiera podwójną spację

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.DoubleSpaceCheck`

Flaga do zignorowania
`ignore-double-space`

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.

Formatowanie ciągów

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 *Dodatkowe informacje o ciągach źródłowych*) or in *Konfiguracja komponentu*. 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.

Podpowiedź: In case specific format check is not available in Weblate, you can use generic *Symbole zastępcze*.

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

Weblate Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate translated 96%

1 / 1 Custom search '%(count)s word' Zen

Position and priority

Translation

English

Singular
%(count)s word

Plural
%(count)s words

Czech, One
%(count)s slovo
15/140 · 14

Czech, Few
%(count)s slova
15/140 · 15

Czech, Other
%(count)s slov
14/140 · 15

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

Needs editing

Save and continue Save and stay Suggest Skip

Nearby strings 20 Comments Automatic suggestions Other languages 3

History

None
String updated in the repository
WeblateOrg / Django — Czech

English
Singular
%(count)s word
Plural
%(count)s words
Czech Translated
One
%(count)s slovo
Few
%(count)s slova
Other
%(count)s slov
6 seconds ago

Glossary
English Czech
No related strings found in the glossary.
Add term to glossary

String information
Screenshot context
No screenshot currently associated.
Add screenshot
Explanation
No explanation currently provided.
Labels
No labels currently set.
Flags
python-format
Source string location
weblate/templates/translation.html:149
String age
6 seconds ago
Source string age
6 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5

Browse all component changes

Powered by Weblate 4.13 About Weblate Legal Contact Documentation Donate to Weblate

Ciąg znaków interpolacji AngularJS

Podsumowanie
Ciągi interpolacji AngularJS nie pasują do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.angularjs.AngularJSInterpolationCheck`

Dostępna flaga
`angularjs-format`

Flaga do zignorowania
`ignore-angularjs-format`

Przykład ciągu formatu z nazwami
`Your balance is {{amount}} {{ currency }}`

Zobacz także:

Formatowanie ciągów, Interpolacja tekstu w AngularJS

Format C

Podsumowanie
Ciąg w formacie C nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.CFormatCheck`

Dostępna flaga
`c-format`

Flaga do zignorowania
`ignore-c-format`

Przykład prostego ciągu formatu
`There are %d apples`

Przykład ciągu formatu pozycji
`Your balance is %1$d %2$s`

Zobacz także:

***Formatowanie ciągów*,**
C format strings, C printf format

Format C#

Podsumowanie
Ciąg w formacie C# nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.CSharpFormatCheck`

Dostępna flaga
`c-sharp-format`

Flaga do zignorowania
`ignore-c-sharp-format`

Przykład ciągu formatu pozycji
`There are {0} apples`

Zobacz także:

Formatowanie ciągów, Format ciągów w C#

Literały szablonu ECMAScript

Podsumowanie
Literały szablonu ECMAScript nie są zgodne ze źródłem

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.ESTemplateLiteralsCheck`

Dostępna flaga
`es-format`

Flaga do zignorowania
`ignore-es-format`

Przykład interpolacji
`There are ${number} apples`

Zobacz także:

Formatowanie ciągów, Template literals

interpolacja i18next

Nowe w wersji 4.0.

Podsumowanie
Interpolacja i18next nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.I18NextInterpolationCheck`

Dostępna flaga
`i18next-interpolation`

Flaga do zignorowania
`ignore-i18next-interpolation`

Przykład interpolacji
`There are {{number}} apples`

Przykład zagnieżdżania
`There are $t(number) apples`

Zobacz także:

Formatowanie ciągów, Interpolacja i18next

ICU MessageFormat

Nowe w wersji 4.9.

Podsumowanie
Błędy składniowe i/lub niezgodności symboli zastępczych w ciągach ICU MessageFormat.

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.icu.ICUMessageFormatCheck`

Dostępna flaga
`icu-message-format`

Flaga do zignorowania
`ignore-icu-message-format`

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

This check has support for both pure ICU MessageFormat messages as well as ICU with simple XML tags. You can configure the behavior of this check by using `icu-flags:*`, either by opting into XML support or by disabling certain sub-checks. For example, the following flag enables XML support while disabling validation of plural sub-messages:

`xml`	Enable support for simple XML tags. By default, XML tags are parsed loosely. Stray < characters are ignored if they are not reasonably part of a tag.
`strict-xml`	Enable support for strict XML tags. All < characters must be escaped if they are not part of a tag.
`-highlight`	Wyłącz wyróżnianie symboli zastępczych w edytorze.
`-require_other`	Disable requiring sub-messages to have an `other` selector.
`-submessage_selector`	Skip checking that sub-message selectors match the source.
`-types`	Pomiń sprawdzanie, czy typy symboli zastępczych są zgodne ze źródłem.
`-extra`	Skip checking that no placeholders are present that were not present in the source string.
`-missing`	Skip checking that no placeholders are missing that were present in the source string.

Additionally, when `strict-xml` is not enabled but `xml` is enabled, you can use the `icu-tag-prefix:PREFIX` flag to require that all XML tags start with a specific string. For example, the following flag will only allow XML tags to be matched if they start with `<x:`:

This would match `<x:link>click here</x:link>` but not `<strong>this</strong>`.

Zobacz także:

Składnia ICU MessageFormat, *Formatowanie ciągów*, ICU: Formatting Messages, Format.JS: Message Syntax

Format Java

Podsumowanie
Ciąg w formacie Java nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.JavaFormatCheck`

Dostępna flaga
`java-format`

Flaga do zignorowania
`ignore-java-format`

Przykład prostego ciągu formatu
`There are %d apples`

Przykład ciągu formatu pozycji
`Your balance is %1$d %2$s`

Zobacz także:

Formatowanie ciągów, Ciągi w formacie Java

Format Java Message

Podsumowanie
Ciąg Java MessageFormat nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.JavaMessageFormatCheck`

Flaga do włączania bezwarunkowego
`java-messageformat`

Flaga włączająca automatyczne wykrywanie
`auto-java-messageformat` włącza sprawdzanie tylko wtedy, gdy w źródle znajduje się ciąg formatujący

Flaga do zignorowania
`ignore-java-messageformat`

Przykład ciągu formatu pozycji
`There are {0} apples`

Zobacz także:

Formatowanie ciągów, Java MessageFormat

Format JavaScript

Podsumowanie
Ciąg w formacie JavaScript nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.JavaScriptFormatCheck`

Dostępna flaga
`javascript-format`

Flaga do zignorowania
`ignore-javascript-format`

Przykład prostego ciągu formatu
`There are %d apples`

Zobacz także:

Formatowanie ciągów, JavaScript formatting strings

Format Lua

Podsumowanie
Ciąg w formacie Lua nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.LuaFormatCheck`

Dostępna flaga
`lua-format`

Flaga do zignorowania
`ignore-lua-format`

Przykład prostego ciągu formatu
`There are %d apples`

Zobacz także:

Formatowanie ciągów, Lua formatting strings

Format Object Pascal

Podsumowanie
Ciąg formatu Object Pascal nie jest zgodny ze źródłem

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.ObjectPascalFormatCheck`

Dostępna flaga
`object-pascal-format`

Flaga do zignorowania
`ignore-object-pascal-format`

Przykład prostego ciągu formatu
`There are %d apples`

Zobacz także:

Formatowanie ciągów, Object Pascal formatting strings, Free Pascal formatting strings Delphi formatting strings

Procent symboli zastępczych

Nowe w wersji 4.0.

Podsumowanie
Procent zapychaczy nie jest zgodny ze źródłem

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.PercentPlaceholdersCheck`

Dostępna flaga
`percent-placeholders`

Flaga do zignorowania
`ignore-percent-placeholders`

Przykład prostego ciągu formatu
`There are %number% apples`

Zobacz także:

Formatowanie ciągów,

Format języka Perl

Podsumowanie
Ciąg w formacie Perl nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.PerlFormatCheck`

Dostępna flaga
`perl-format`

Flaga do zignorowania
`ignore-perl-format`

Przykład prostego ciągu formatu
`There are %d apples`

Przykład ciągu formatu pozycji
`Your balance is %1$d %2$s`

Zobacz także:

Formatowanie ciągów, Perl sprintf, Perl Format Strings

Format PHP

Podsumowanie
Ciąg w formacie PHP nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.PHPFormatCheck`

Dostępna flaga
`php-format`

Flaga do zignorowania
`ignore-php-format`

Przykład prostego ciągu formatu
`There are %d apples`

Przykład ciągu formatu pozycji
`Your balance is %1$d %2$s`

Zobacz także:

Formatowanie ciągów, PHP sprintf documentation, PHP Format Strings

Format klamrowy Python

Podsumowanie
Ciąg w formacie Python brace nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.PythonBraceFormatCheck`

Dostępna flaga
`python-brace-format`

Flaga do zignorowania
`ignore-python-brace-format`

Prosty format ciągów
`There are {} apples`

Przykład ciągu formatu z nazwami
`Your balance is {amount} {currency}`

Zobacz także:

Formatowanie ciągów, Python brace format, Python Format Strings

Format Python

Podsumowanie
Ciąg w formacie Python nie pasuje do ciągu źródłowego

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.PythonFormatCheck`

Dostępna flaga
`python-format`

Flaga do zignorowania
`ignore-python-format`

Prosty format ciągów
`There are %d apples`

Przykład ciągu formatu z nazwami
`Your balance is %(amount)d %(currency)s`

Zobacz także:

Formatowanie ciągów, Python string formatting, Python Format Strings

Format Qt

Podsumowanie
Ciąg w formacie Qt nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.qt.QtFormatCheck`

Dostępna flaga
`qt-format`

Flaga do zignorowania
`ignore-qt-format`

Przykład ciągu formatu pozycji
`There are %1 apples`

Zobacz także:

Formatowanie ciągów, Qt QString::arg()

Forma liczby mnogiej Qt

Podsumowanie
Ciąg w formacie liczby mnogiej Qt nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.qt.QtPluralCheck`

Dostępna flaga
`qt-plural-format`

Flaga do zignorowania
`ignore-qt-plural-format`

Przykład ciągu w formacie liczby mnogiej
`There are %Ln apple(s)`

Zobacz także:

Formatowanie ciągów, Przewodnik po Qt i18n

Format Ruby

Podsumowanie
Ciąg w formacie Ruby nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.ruby.RubyFormatCheck`

Dostępna flaga
`ruby-format`

Flaga do zignorowania
`ignore-ruby-format`

Przykład prostego ciągu formatu
`There are %d apples`

Przykład ciągu formatu pozycji
`Your balance is %1$f %2$s`

Przykład ciągu formatu z nazwami
`Your balance is %+.2<amount>f %<currency>s`

Named template string
`Your balance is %{amount} %{currency}`

Zobacz także:

Formatowanie ciągów, Ruby Kernel#sprintf

Format Scheme

Podsumowanie
Ciąg w formacie Scheme nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.SchemeFormatCheck`

Dostępna flaga
`scheme-format`

Flaga do zignorowania
`ignore-scheme-format`

Przykład prostego ciągu formatu
`There are ~d apples`

Zobacz także:

Formatowanie ciągów, Srfi 28, Chicken Scheme format, Guile Scheme formatted output

Formatowanie Vue I18n

Podsumowanie
Formatowanie Vue I18n nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.format.VueFormattingCheck`

Dostępna flaga
`vue-format`

Flaga do zignorowania
`ignoruj-vue-format`

Nazwane formatowanie
`There are {count} apples`

Formatowanie i18n w Rails
`There are %{count} apples`

Połączone wiadomości lokalne
`@:message.dio @:message.the_world!`

Zobacz także:

Formatowanie ciągów, Vue I18n Formatting, Vue I18n Linked locale messages

Zostało przetłumaczone

Podsumowanie
Ten ciąg został już przetłumaczony w przeszłości

Zakres
wszystkie ciągi

Sprawdź klasę
`weblate.checks.consistency.TranslatedCheck`

Flaga do zignorowania
`ignore-translated`

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

Niespójność

Podsumowanie
Ten ciąg ma więcej niż jedno tłumaczenie w tym projekcie lub jest nieprzetłumaczony w niektórych komponentach.

Zakres
wszystkie ciągi

Sprawdź klasę
`weblate.checks.consistency.ConsistencyCheck`

Flaga do zignorowania
`ignore-inconsistent`

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.

Ta kontrola dotyczy wszystkich komponentów w projekcie, które mają włączoną opcję *Zezwól na propagację tłumaczenia*.

Podpowiedź: For performance reasons, the check might not find all inconsistencies, it limits number of matches.

Informacja: 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 untranslated in some components just by clicking on the *Use this translation* button displayed on each line in the *Other occurrences* tab.

You can use *Tłumaczenie automatyczne* add-on to automate translating of newly added strings which are already translated in another component.

Zobacz także:

Zachowanie takich samych tłumaczeń między komponentami

Użyto litery Kashida

Nowe w wersji 3.5.

Podsumowanie
Ozdobne litery kashida nie powinny być używane

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.KashidaCheck`

Flaga do zignorowania
`ignore-kashida`

The decorative Kashida letters should not be used in translation. These are also known as Tatweel.

Zobacz także:

Kashida w Wikipedii

Łącza Markdown

Nowe w wersji 3.5.

Podsumowanie
Łącza Markdown nie pasują do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.markup.MarkdownLinkCheck`

Dostępna flaga
`md-text`

Flaga do zignorowania
`ignore-md-link`

Markdown links do not match source.

Zobacz także:

Łącza Markdown

Odniesienia do Markdown

Nowe w wersji 3.5.

Podsumowanie
Odnośniki do łączy Markdown nie pasują do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.markup.MarkdownRefLinkCheck`

Dostępna flaga
`md-text`

Flaga do zignorowania
`ignore-md-reflink`

Markdown link references do not match source.

Zobacz także:

Markdown links

Składnia Markdown

Nowe w wersji 3.5.

Podsumowanie
Składnia Markdown nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.markup.MarkdownSyntaxCheck`

Dostępna flaga
`md-text`

Flaga do zignorowania
`ignore-md-syntax`

Składnia Markdown nie pasuje do źródła

Zobacz także:

Markdown span elements

Maksymalna długość tłumaczenia

Podsumowanie
Tłumaczenie nie powinno przekraczać określonej długości

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.MaxLengthCheck`

Dostępna flaga
`max-length`

Flaga do zignorowania
`ignore-max-length`

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`.

Podpowiedź: This check looks at number of chars, what might not be the best metric when using proportional fonts to render the text. The *Maksymalny rozmiar tłumaczenia* check does check actual rendering of the text.

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

When `xml-text` flag is also used, the length calculation ignores XML tags.

Maksymalny rozmiar tłumaczenia

Podsumowanie
Tłumaczenie renderowanego tekstu nie powinno przekraczać podanego rozmiaru

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.render.MaxSizeCheck`

Dostępna flaga
`max-size`

Flaga do zignorowania
`ignore-max-size`

Nowe w wersji 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 *Dostosowywanie zachowania za pomocą flag*), 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
```

Podpowiedź: You might want to set `font-*` directives in *Konfiguracja komponentu* 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.

When `xml-text` flag is also used, the length calculation ignores XML tags.

Zobacz także:

Zarządzanie czcionkami, *Dostosowywanie zachowania za pomocą flag*, *Maksymalna długość tłumaczenia*

Niedopasowane \n

Podsumowanie
Liczba \n w tłumaczeniu nie pasuje do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.EscapedNewlineCountingCheck`

Flaga do zignorowania
`ignore-escaped-newline`

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

Niedopasowany dwukropek

Podsumowanie
Źródło i tłumaczenie nie kończą się dwukropkiem

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.EndColonCheck`

Flaga do zignorowania
`ignore-end-colon`

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).

Zobacz także:

Colon w Wikipedii

Niedopasowane wielokropki

Podsumowanie
Źródło i tłumaczenie nie kończą się wielokropkiem

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.EndEllipsisCheck`

Flaga do zignorowania
`ignore-end-ellipsis`

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.

Zobacz także:

Ellipsis w Wikipedii

Niedopasowany wykrzyknik

Podsumowanie
Źródło i tłumaczenie nie kończą się wykrzyknikiem

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.EndExclamationCheck`

Flaga do zignorowania
`ignore-end-exclamation`

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).

Zobacz także:

Exclamation mark on Wikipedia

Niedopasowana kropka

Podsumowanie
Źródło i tłumaczenie nie kończą się kropką

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.EndStopCheck`

Flaga do zignorowania
`ignore-end-stop`

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).

Zobacz także:

Full stop on Wikipedia

Niedopasowany znak zapytania

Podsumowanie
Źródło i tłumaczenie nie kończą się znakiem zapytania

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.EndQuestionCheck`

Flaga do zignorowania
`ignore-end-question`

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).

Zobacz także:

Question mark on Wikipedia

Niedopasowany średnik

Podsumowanie
Źródło i tłumaczenie nie kończą się średnikiem

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.EndSemicolonCheck`

Flaga do zignorowania
`ignore-end-semicolon`

Checks that semicolons at the end of sentences are replicated between both source and translation.

Zobacz także:

Semicolon w Wikipedii

Niedopasowanie podziałów wierszy

Podsumowanie
Liczba nowych wierszy w tłumaczeniu nie jest zgodna ze źródłem

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.NewLineCountCheck`

Flaga do zignorowania
`ignore-newline-count`

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

Brakująca forma liczby mnogiej

Podsumowanie
Niektóre formy liczby mnogiej są nieprzetłumaczone

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.consistency.PluralsCheck`

Flaga do zignorowania
`ignore-plurals`

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.

Symbole zastępcze

Nowe w wersji 3.9.

Podsumowanie
W tłumaczeniu brakuje niektórych symboli zastępczych

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.placeholders.PlaceholderCheck`

Dostępna flaga
`placeholders`

Flaga do zignorowania
`ignore-placeholders`

Zmienione w wersji 4.3: Możesz użyć wyrażenia regularnego jako symbolu zastępczego.

Zmienione w wersji 4.13: With the `case-insensitive` flag, the placeholders are not case-sensitive.

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"
```

In case you have some syntax for placeholders, you can use a regular expression:

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

You can also have case insensitive placeholders:

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

Zobacz także:

Dostosowywanie zachowania za pomocą flag

Odstępy między czcionkami

Nowe w wersji 3.9.

Podsumowanie
Brak niełamliwej spacji przed podwójnym znakiem interpunkcyjnym

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.PunctuationSpacingCheck`

Flaga do zignorowania
`ignore-punctuation-spacing`

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.

Zobacz także:

French and English spacing on Wikipedia

Wyrażenie regularne

Nowe w wersji 3.9.

Podsumowanie
Tłumaczenie nie pasuje do wyrażenia regularnego

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.placeholders.RegexCheck`

Dostępna flaga
`regex`

Flaga do zignorowania
`ignore-regex`

Translation does not match regular expression. The expression is either extracted from the translation file or defined manually using `regex` flag:

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

Te same liczby mnogie

Podsumowanie
Niektóre formy mnogie są tłumaczone w ten sam sposób

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.consistency.SamePluralsCheck`

Flaga do zignorowania
`ignore-same-plurals`

Check that fails if some plural forms are duplicated in the translation. In most languages they have to be different.

Początek od nowej linii

Podsumowanie
Źródło i tłumaczenie nie zaczynają się od nowej linii

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.BeginNewlineCheck`

Flaga do zignorowania
`ignore-begin-newline`

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.

Zobacz także:

Zakończenie nową linią

Spacje początkowe

Podsumowanie
Źródło i tłumaczenie nie zaczynają się od tej samej liczby spacji

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.BeginSpaceCheck`

Flaga do zignorowania
`ignore-begin-space`

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

Zakończenie nową linią

Podsumowanie
Źródło i tłumaczenie nie kończą się nową linią

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.EndNewlineCheck`

Flaga do zignorowania
`ignore-end-newline`

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.

Zobacz także:

Początek od nowej linii

Końcowa spacja

Podsumowanie
Źródło i tłumaczenie nie kończą się spacją

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.EndSpaceCheck`

Flaga do zignorowania
`ignore-end-space`

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.

Niezmienione tłumaczenie

Podsumowanie
Źródło i tłumaczenie są identyczne

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.same.SameCheck`

Flaga do zignorowania
`ignore-same`

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.

This check can help find strings mistakenly untranslated.

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.

Zobacz także:

Konfiguracja komponentu, *Dostosowywanie zachowania za pomocą flag*

Niebezpieczny HTML

Nowe w wersji 3.9.

Podsumowanie
Tłumaczenie używa niebezpiecznych znaczników HTML

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.markup.SafeHTMLCheck`

Dostępna flaga
`safe-html`

Flaga do zignorowania
`ignore-safe-html`

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

Podpowiedź: When `md-text` flag is also used, the Markdown style links are also allowed.

Zobacz także:

The HTML check is performed by the Bleach library developed by Mozilla.

URL

Nowe w wersji 3.5.

Podsumowanie
Tłumaczenie nie zawiera adresu URL

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.markup.URLCheck`

Dostępna flaga
`url`

Flaga do zignorowania
`ignore-url`

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.

Znaczniki XML

Podsumowanie
Znaczniki XML w przetłumaczonym tekście nie pasują do źródła

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.markup.XMLTagsCheck`

Flaga do zignorowania
`ignore-xml-tags`

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.

Informacja: This check is disabled by the `safe-html` flag as the HTML cleanup done by it can produce HTML markup which is not valid XML.

Składnia XML

Nowe w wersji 2.8.

Podsumowanie
Tłumaczenie nie jest w prawidłowym formacie XML

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.markup.XMLValidityCheck`

Flaga do zignorowania
`ignore-xml-invalid`

The XML markup is not valid.

Informacja: This check is disabled by the `safe-html` flag as the HTML cleanup done by it can produce HTML markup which is not valid XML.

Znak spacji o zerowej szerokości

Podsumowanie
Tłumaczenie zawiera dodatkowy znak spacji o zerowej szerokości

Zakres
przetłumaczone ciągi

Sprawdź klasę
`weblate.checks.chars.ZeroWidthSpaceCheck`

Flaga do zignorowania
`ignore-zero-width-space`

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.

Zobacz także:

Zero width space on Wikipedia

1.6.4 Sprawdzanie źródła

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

Wielokropek

Podsumowanie
Ciąg zawiera trzy kropki (…) zamiast znaku wielokropka (…)

Zakres
teksty źródłowe

Sprawdź klasę
`weblate.checks.source.EllipsisCheck`

Flaga do zignorowania
`ignore-ellipsis`

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.

Zobacz także:

Ellipsis w Wikipedii

Składnia ICU MessageFormat

Nowe w wersji 4.9.

Podsumowanie
Błędy składniowe w ciągach ICU MessageFormat.

Zakres
teksty źródłowe

Sprawdź klasę
`weblate.checks.icu.ICUSourceCheck`

Dostępna flaga
`icu-message-format`

Flaga do zignorowania
`ignore-icu-message-format`

Zobacz także:

ICU MessageFormat

Długo nieprzetłumaczone

Nowe w wersji 4.1.

Podsumowanie
Ciąg nie był tłumaczony przez długi czas

Zakres
teksty źródłowe

Sprawdź klasę
`weblate.checks.source.LongUntranslatedCheck`

Flaga do zignorowania
`ignore-long-untranslated`

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

Wiele nieudanych kontroli

Podsumowanie
Tłumaczenia w kilku językach nie przeszły kontroli

Zakres
teksty źródłowe

Sprawdź klasę
`weblate.checks.source.MultipleFailingCheck`

Flaga do zignorowania
`ignore-multiple-failures`

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.

Wiele nienazwanych zmiennych

Nowe w wersji 4.1.

Podsumowanie
W łańcuchu znajduje się wiele nienazwanych zmiennych, co uniemożliwia tłumaczom zmianę ich kolejności

Zakres
teksty źródłowe

Sprawdź klasę
`weblate.checks.format.MultipleUnnamedFormatsCheck`

Flaga do zignorowania
`ignore-unnamed-format`

There are multiple unnamed variables in the string, making it impossible for translators to reorder them.

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

Niespluralizowane

Podsumowanie
Ciąg jest używany jako liczba mnoga, ale nie przy użyciu form mnogich

Zakres
teksty źródłowe

Sprawdź klasę
`weblate.checks.source.OptionalPluralCheck`

Flaga do zignorowania
`ignore-optional-plural`

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.

For example with Gettext in Python it could be:

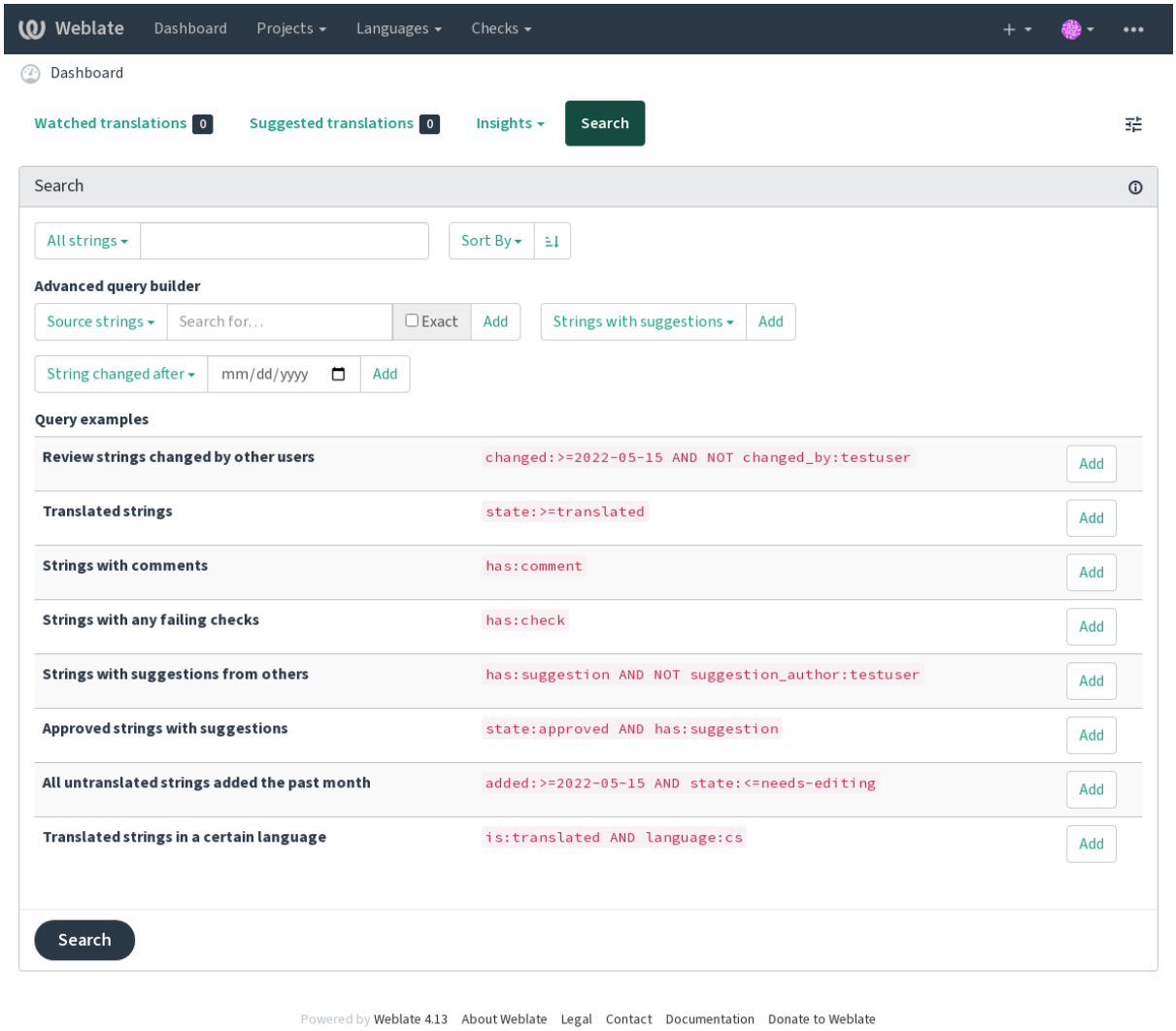
```
from gettext import ngettext

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

1.7 Wyszukiwanie

Nowe w wersji 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 strings.



1.7.1 Proste wyszukiwanie

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'`.

1.7.2 Pola

`source:TEKST`
Wyszukiwanie bez rozróżniania wielkości liter w ciągu źródłowym.

`target:TEKST`
Target string case-insensitive search.

`context:TEKST`
Context string case-insensitive search.

`key:TEKST`
Key string case-insensitive search.

`note:TEKST`
Wyszukiwanie bez rozróżniania wielkości liter w opisie ciągu źródłowego.

location:TEKST
Location string case-insensitive search.

priority:
Priorytet ciągu.

added:DATETIME
Timestamp for when the string was added to Weblate.

state:TEKST
State search (`approved`, `translated`, `needs-editing`, `empty`, `read-only`), supports *Operatory pól*.

pending:BOOLEAN
String pending for flushing to VCS.

has:TEKST
Search for string having attributes - `plural`, `context`, `suggestion`, `comment`, `check`, `dismissed-check`, `translation`, `variant`, `screenshot`, `flags`, `explanation`, `glossary`, `note`, `label`.

is:TEKST
Search for string states (`pending`, `translated`, `untranslated`).

language:TEKST
Docelowy język ciągu.

component:TEXT
Component slug or name case-insensitive search, see *Końcówka adresu URL komponentu* and *Nazwa komponentu*.

project:TEXT
Project slug, see *Końcówka adresu URL*.

changed_by:TEKST
String was changed by author with given username.

changed:DATETIME
String content was changed on date, supports *Operatory pól*.

change_time:DATETIME
String was changed on date, supports *Operatory pól*, unlike `changed` this includes event which don't change content and you can apply custom action filtering using `change_action`.

change_action:TEXT
Filters on change action, useful together with `change_time`. Accepts English name of the change action, either quoted and with spaces or lowercase and spaces replaced by a hyphen. See *Wyszukiwanie zmian* for examples.

check:TEKST
String has failing check.

dismissed_check:TEKST
String has dismissed check.

comment:TEKST
Szukaj w komentarzach użytkowników.

resolved_comment:TEXT
Szukaj w rozwiązanych komentarzach.

comment_author:TEKST
Filtruj według autora komentarza.

suggestion:TEKST
Szukaj w sugestiach.

suggestion_author:TEKST
Filtruj według autora sugestii.

explanation:TEXT
Szukaj w objaśnieniach.

label:TEXT
Szukaj w etykietach.

screenshot:TEXT
Szukaj w zrzutach ekranu.

1.7.3 Operatory logiczne

You can combine lookups using `AND`, `OR`, `NOT` and parentheses to form complex queries. For example: `state:translated AND (source:hello OR source:bar)`

1.7.4 Operatory pól

You can specify operators, ranges or partial lookups for date or numeric searches:

state:>=translated
State is `translated` or better (`approved`).

changed:2019
Zmieniono w roku 2019.

zmieniono:[2019-03-01 na 2019-04-01]
Changed between two given dates.

1.7.5 Dokładne operatory

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`.

1.7.6 Wyszukiwanie zmian

Nowe w wersji 4.4.

Searching for history events can be done using `change_action` and `change_time` operators.

For example, searching for strings marked for edit in 2018 can be entered as `change_time:2018 AND change_action:marked-for-edit` or `change_time:2018 AND change_action:"Marked for edit"`.

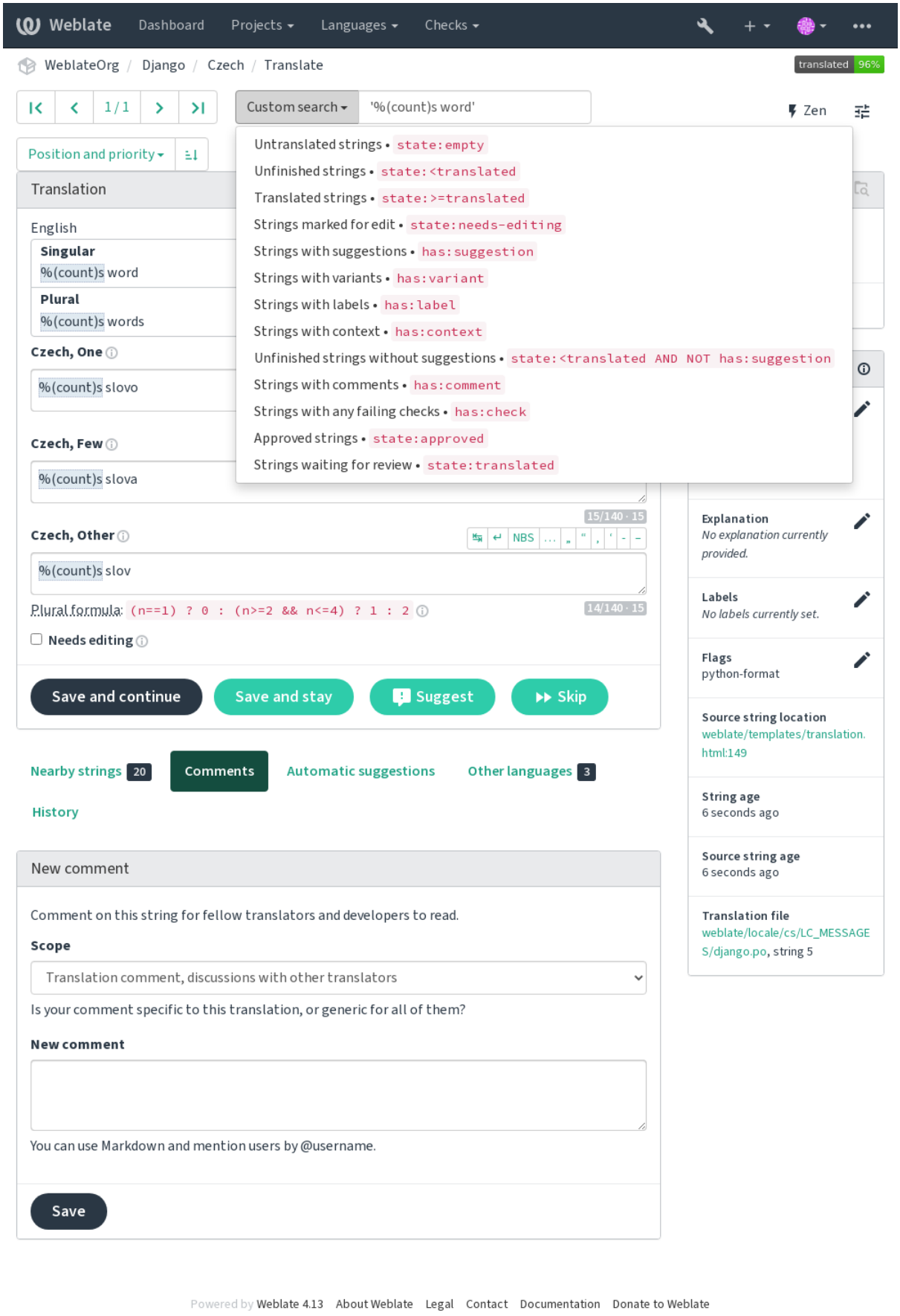
1.7.7 Wyrażenie regularne

Anywhere text is accepted you can also specify a regular expression as `r"regexp"`.

For example, to search for all source strings which contain any digit between 2 and 5, use `source:r"[2-5]"`.

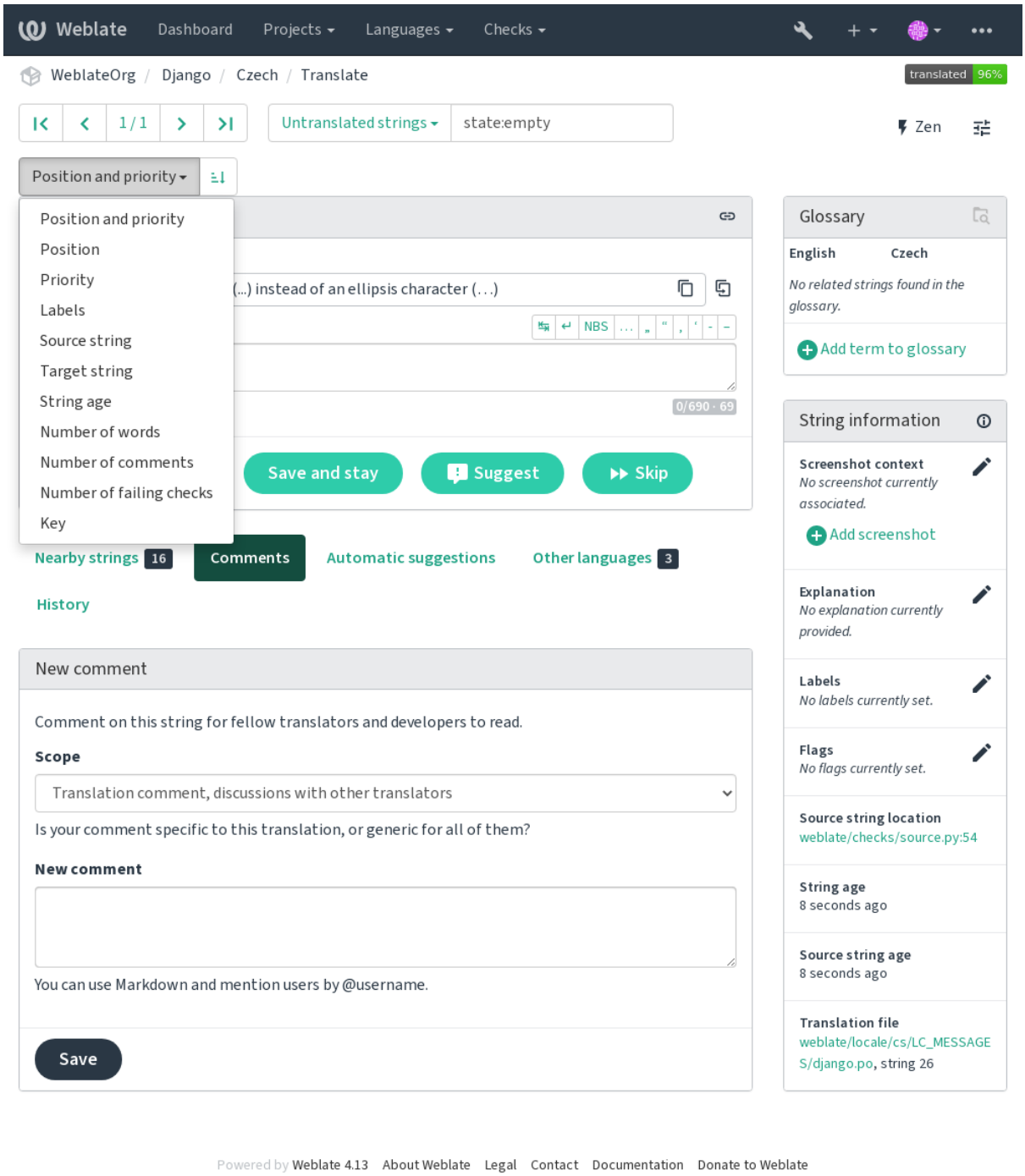
1.7.8 Wstępnie zdefiniowane zapytania

You can select out of predefined queries on the search page, this allows you to quickly access the most frequent searches:


Weblate
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Czech / Translate
translated 96%
1 / 1
Custom search
'%(count)s word'
Zen
Position and priority
Untranslated strings • state:empty
Unfinished strings • state:<translated
Translated strings • state:>=translated
Strings marked for edit • state:needs-editing
Strings with suggestions • has:suggestion
Strings with variants • has:variant
Strings with labels • has:label
Strings with context • has:context
Unfinished strings without suggestions • state:<translated AND NOT has:suggestion
Strings with comments • has:comment
Strings with any failing checks • has:check
Approved strings • state:approved
Strings waiting for review • state:translated
Translation
English
Singular
%(count)s word
Plural
%(count)s words
Czech, One
%(count)s slovo
Czech, Few
%(count)s slova
15/140 · 15
Czech, Other
%(count)s slov
Plural formula: (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2
14/140 · 15
Needs editing
Save and continue
Save and stay
Suggest
Skip
Nearby strings 20
Comments
Automatic suggestions
Other languages 3
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
Explanation
No explanation currently provided.
Labels
No labels currently set.
Flags
python-format
Source string location
weblate/templates/translation.html:149
String age
6 seconds ago
Source string age
6 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5
Powered by Weblate 4.13 About Weblate Legal Contact Documentation Donate to Weblate

1.7.9 Ordering the results

There are many options to order the strings according to your needs:



1.8 Przepływy pracy tłumaczenia

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

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 Dostęp do tłumaczeń

The *access control* is not discussed in detail as a whole in the workflows, as most of its options can be applied to any workflow. Please consult the respective documentation 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 Stan tłumaczenia

Each translated string can be in one of following states:

Nieprzetłumaczone
Translation is empty, it might or not be stored in the file, depending on the file format.

Wymaga edycji
Translation needs editing, this is usually the result of a source string change, fuzzy matching or translator action. The translation is stored in the file, depending on the file format it might be marked as needing edit (for example as it gets a `fuzzy` flag in the Gettext file).

Oczekujące na przegląd
Translation is made, but not reviewed. It is stored in the file as a valid translation.

Zatwierdzone
Translation has been approved in the review. It can no longer be changed by translators, but only by reviewers. Translators can only add suggestions to it.

This state is only available when reviews are enabled.

Sugestie
Suggestions are stored in Weblate only and not in the translation file.

W miarę możliwości stany są reprezentowane w plikach tłumaczeń.

Podpowiedź: In case file format you use does not support storing states, you might want to use *Oznacz niezmienione tłumaczenia jako „Wymaga edycji”* add-on to flag unchanged strings as needing editing.

Zobacz także:

Możliwe typy tłumaczeń, *Przepływy pracy tłumaczenia*

1.8.3 Bezpośrednie tłumaczenie

This is most usual setup for smaller teams, anybody can directly translate. This is also the default setup in Weblate.

- *Każdy urztkownik* może edytować tłumaczenie.
- Suggestions are optional ways to suggest changes, when translators are not sure about the change.

Ustawienia	Wartość	Uwaga
Włącz przeglądy	wyłączona	Skonfigurowane na poziomie projektu.
Włącz sugestie	na	It is useful for users to be able to suggest when they are not sure.
Głosowanie na sugestie	wyłączona	
Automatyczne akceptowanie sugestii	0	
Grupa Tłumaczy	*Użytkownicy*	Lub *Przetłumacz* z *per-project access control*.
Grupa recenzentów	N/A	Nie używane.

1.8.4 Ocena koleżeńska

With this workflow, anybody can add suggestions, and need approval from additional member(s) before it is accepted as a translation.

- *Any user* can add suggestions.
- *Any user* can vote for suggestions.
- Suggestions become translations when given a predetermined number of votes.

Ustawienia	Wartość	Uwaga
Włącz przeglądy	wyłączona	Skonfigurowane na poziomie projektu.
Włącz sugestie	na	
Głosowanie na sugestie	wyłączona	
Automatyczne akceptowanie sugestii	1	You can set higher value to require more peer reviews.
Grupa Tłumaczy	*Użytkownicy*	Lub *Przetłumacz* z *per-project access control*.
Grupa recenzentów	N/A	Not used, all translators review.

1.8.5 Dedykowani recenzenci

Nowe w wersji 2.18: The proper review workflow is supported since Weblate 2.18.

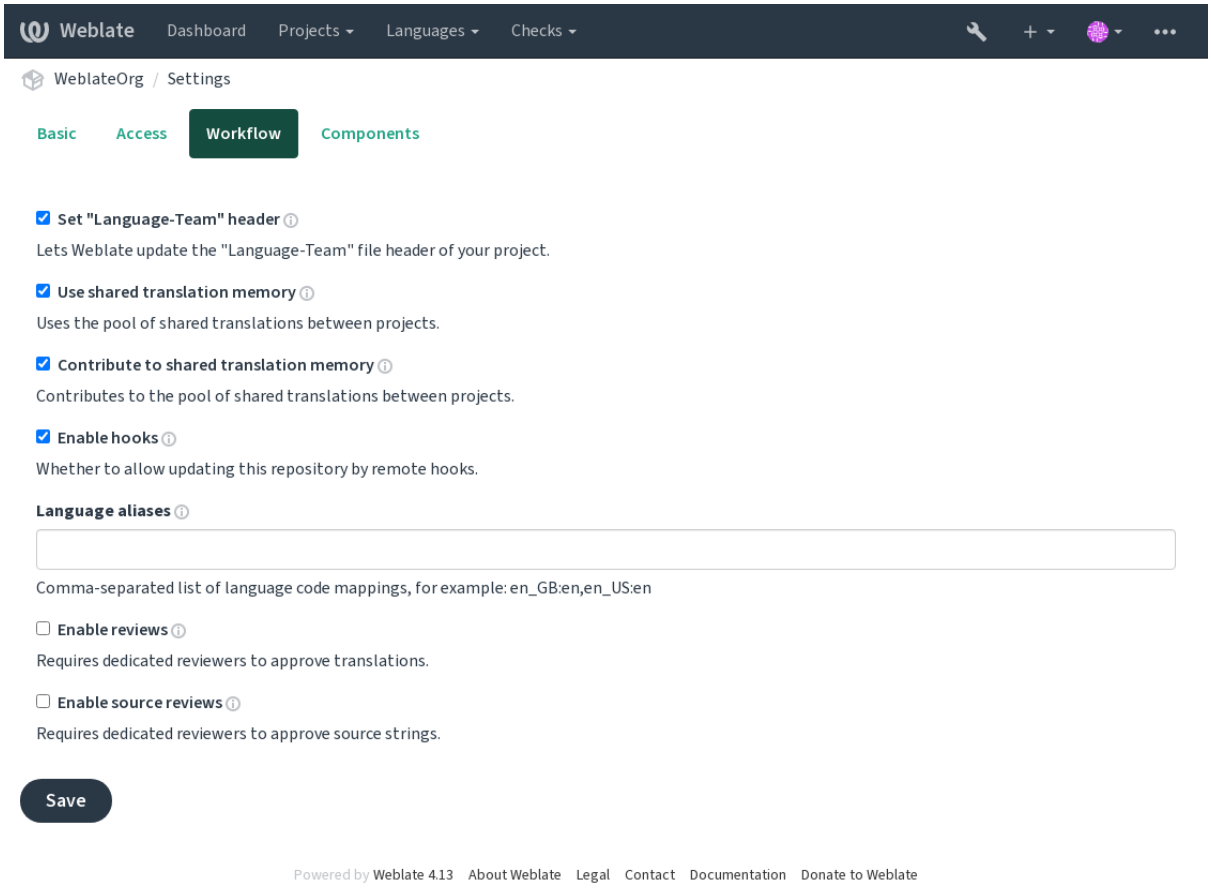
With dedicated reviewers you have two groups of users, one able to submit translations, and one able to review them to ensure translations are consistent and that the quality is good.

- *Any user* can edit unapproved translations.
- *Recenzent* może zatwierdzać / cofać zatwierdzenie ciągów.
- *Reviewer* can edit all translations (including approved ones).
- Suggestions can also be used to suggest changes for approved strings.

Ustawienia	Wartość	Uwaga
Włącz przeglądy	na	Skonfigurowane na poziomie projektu.
Włącz sugestie	wyłączona	It is useful for users to be able to suggest when they are not sure.
Głosowanie na sugestie	wyłączona	
Automatyczne akceptowanie sugestii	0	
Grupa Tłumaczy	*Użytkownicy*	Lub *Przetłumacz* z *per-project access control*.
Grupa recenzentów	*Recenzenci*	Or *Review* with *per-project access control*.

1.8.6 Włączanie przeglądów

Reviews can be turned on in the project configuration, from the *Workflow* subpage of project settings (to be found in the *Manage* → *Settings* menu):

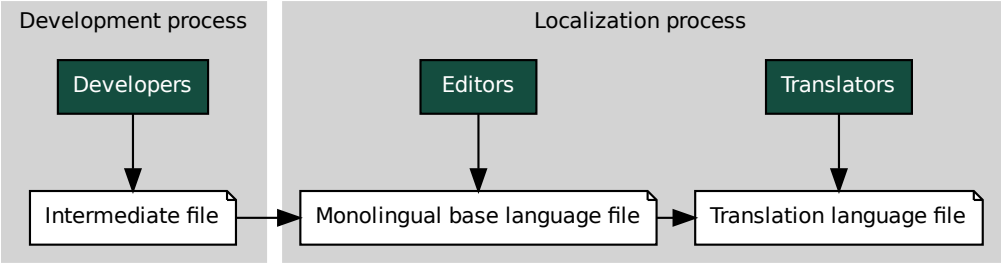


Informacja: Depending on Weblate configuration, the setting might not be available to you. For example on Hosted Weblate this is not available for projects hosted for free.

1.8.7 Quality gateway for the source strings

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 *Plik języka pośredniego*, 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.



Zobacz także:

Plik języka pośredniego, *Jednojęzyczny bazowy plik języka*, *Formaty dwujęzyczne i jednojęzyczne*

1.8.8 Przeglądy ciągów źródłowych

With *Włącz przeglądy źródeł* 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 *Dedykowani recenzenci* - 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.

Zobacz także:

Formaty dwujęzyczne i jednojęzyczne, *Dedykowani recenzenci*, labels, *Komentarze*

1.9 Najczęściej zadawane pytania

1.9.1 Konfiguracja

How to create an automated workflow?

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 *Hooki powiadomień* for info on how to do it.

2. Set a push URL at your *Konfiguracja komponentu* in Weblate, this allows Weblate to push changes to your repository.
3. Turn on *Przesyłaj przy commitowaniu* on your *Konfiguracja komponentu* in Weblate, this will make Weblate push changes to your repository whenever they happen at Weblate.

Zobacz także:

Ciągła lokalizacja, *Unikanie konfliktów scalania*

How to access repositories over SSH?

Please see *Dostęp do repozytoriów* for info on setting up SSH keys.

How to fix merge conflicts in translations?

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 encountered a merge conflict, the easiest way to solve all conflicts locally on your machine, is to 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.

Informacja: Depending on your setup, access to the Weblate repository might require authentication. When using the built-in *Eksporter 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/main

# 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

# Merge main branch:
git checkout main
git merge weblates-second/main
... # 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/main

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

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

Zobacz także:

How to export the Git repository that Weblate uses?, *Ciągła lokalizacja*, *Unikanie konfliktów scalania*, *Klient Weblate*

How do I translate several branches at once?

Weblate supports pushing translation changes within one *Konfiguracja projektu*. For every *Konfiguracja komponentu* 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
```

Zobacz także:

Zachowanie takich samych tłumaczeń między komponentami

How to translate multi-platform projects?

Weblate supports a wide range of file formats (see *Obsługiwane formaty plików*) and the easiest approach is to use the native format for each platform.

Once you have added all platform translation files as components in one project (see *Dodawanie projektów i komponentów tłumaczeniowych*), you can utilize the translation propagation feature (turned on by default, and can be turned off in the *Konfiguracja komponentu*) to translate strings for all platforms at once.

Zobacz także:

Zachowanie takich samych tłumaczeń między komponentami

How to export the Git repository that Weblate uses?

There is nothing special about the repository, it lives under the `DATA_DIR` directory and is named `vcs/<project>/<component>/`. If you have SSH access to this machine, you can use the repository directly.

For anonymous access, you might want to run a Git server and let it serve the repository to the outside world.

Alternatively, you can use *Eksporter Git* inside Weblate to automate this.

What are the options for pushing changes back upstream?

This heavily depends on your setup, Weblate is quite flexible in this area. Here are examples of some workflows used with Weblate:

- Weblate automatically pushes and merges changes (see *How to create an automated workflow?*).
- You manually tell Weblate to push (it needs push access to the upstream repository).
- Somebody manually merges changes from the Weblate git repository into the upstream repository.
- Somebody rewrites history produced by Weblate (e.g. by eliminating merge commits), merges changes, and tells Weblate to reset the content in the upstream repository.

Of course you are free to mix all of these as you wish.

How can I limit Weblate access to only translations, without exposing source code to it?

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

1. Create a repository with your translation files.
2. Add this as a submodule to your code:

```
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.

Zobacz także:

Monitorowanie eblate, *Monitorowanie stanu Celery*

Why are all commits committed by Weblate <noreply@weblate.org>?

This is the default committer name, configured by `DEFAULT_COMMITER_EMAIL` and `DEFAULT_COMMITER_NAME`.

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

For commits where no authorship is known (for example anonymous suggestions or machine translation results), the authorship is credited to the anonymous user (see `ANONYMOUS_USER_NAME`). You can change the name and e-mail in the management interface.

Zobacz także:

Konfiguracja komponentu

How to move files in the repository without losing history in Weblate?

To keep the history, comments, or screenshots linked to strings after changing the files location you need to ensure that these strings are never deleted in Weblate. These removals can happen in case the Weblate repository is updated, but the component configuration still points to the old files. This makes Weblate assume that it should delete all the translations.

The solution to this is to perform the operation in sync with Weblate:

1. Zablokuj powiązany komponent w Weblate.
2. Scommituj wszelkie oczekujące zmiany i scal je z głównym repozytorium.
3. Disable receiving webhooks the *Konfiguracja projektu*; this prevents Weblate from immediately seeing changes in the repository.
4. Do any needed changes in the repo (for example using **`git mv`**), push them to the upstream repository.
5. Change the *Konfiguracja komponentu* to match the new setup; upon changing configuration, Weblate will fetch the updated repository and notice the changed locations while keeping existing strings.

6. Odblokuj komponent i ponownie włącz hooki w konfiguracji projektu.

1.9.2 Zastosowanie

How do I review the translations of others?

- There are several review based workflows available in Weblate, see *Przepływy pracy tłumaczenia*.
- You can subscribe to any changes made in *Powiadomienia* 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.

Zobacz także:

Przepływy pracy tłumaczenia

How do I provide feedback on a source string?

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

Zobacz także:

report-source, *Komentarze*

How can I use existing translations while translating?

- Dzięki wspólnej pamięci tłumaczeń można korzystać ze wszystkich tłumaczeń w Weblate.
- Możesz importować istniejące pliki pamięci tłumaczeniowej do programu Weblate.
- 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.

Zobacz także:

Konfigurowanie automatycznych sugestii, *Automatyczne sugestie*, *Pamięć tłumaczeniowa*

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.

Zobacz także:

updating-target-files

Where do language definitions come from and how can I add my own?

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.

Zobacz także:

Definicje języków

Can Weblate highlight changes in a fuzzy string?

Weblate supports this, however it needs the data to show the difference.

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.

Informacja: 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 add-on *Zaktualizuj pliki PO, aby dopasować POT (msgmerge)*.

Zobacz także:

updating-target-files

1.9.3 Rozwiązywanie problemów

Requests sometimes fail with „too many open files" error

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

The easiest way to do this is to run:

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

Zobacz także:

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"]
```

Zobacz także:

Konfiguracja dozwolonych hostów

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.

- Jeśli wymagane jest tłumaczenie na język źródłowy, *Język źródłowy* ustawieniach komponentu.
- Jeśli plik tłumaczenia dla języka źródłowego nie jest potrzebny, usuń go z repozytorium.
- W przypadku, gdy plik tłumaczeniowy dla języka źródłowego jest potrzebny, ale powinien być ignorowany przez Weblate, proszę ustawićref: *component-language_regex*, aby go wykluczyć.

Podpowiedź: You might get similar error message for other languages as well. In that case the most likely reason is that several files map to single language in Weblate.

This can be caused by using obsolete language codes together with new one (`ja` and `jp` for Japanese) or including both country specific and generic codes (`fr` and `fr_FR`). See *Parsowanie kodów językowych* for more details.

1.9.4 Funkcje

Does Weblate support other VCSes than Git and Mercurial?

Weblate currently does not have native support for anything other than *Git* (with extended support for *GitHub pull requests*, *Gerrit* and *Subversion*) and *Mercurial*, but it is possible to write backends for other VCSes.

You can also use *Git remote helpers* in Git to access other VCSes.

Weblate also supports VCS-less operation, see *Pliki lokalne*.

Informacja: 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.

Zobacz także:

Integracja kontroli wersji

How does Weblate credit translators?

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.

Zobacz także:

`list_translators`, ../devel/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.

Zmienione w wersji 4.2: Tłumacze mogą przetłumaczyć wszystkie elementy projektu na określony język jako całość.

Why does Weblate use language codes such sr_Latn or zh_Hant?

These are language codes defined by **RFC 5646** 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 `zh_Hans`.

Informacja: Weblate defaults to POSIX style language codes with underscore, see *Definicje języków* for more details.

Zobacz także:

Definicje języków, *Styl kodu języka*, *Dodawanie nowych tłumaczeń*

1.10 Obsługiwane formaty plików

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.

Zobacz także:

Translation Related File Formats

Informacja: 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 Formaty dwujęzyczne i jednojęzyczne

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 *Ciągi 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 *Android string resources*). 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 *Jednojęzyczny bazowy plik języka* within Weblate, though the naming might vary in your paradigm.

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

1.10.2 Automatyczne wykrywanie

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 Możliwe typy tłumaczeń

Capabilities of all supported formats:

Format	Lingualność[Strona 77, 1]	Liczba mnoga[Strona 77, 2]	Opisy[Strona 77, 3]	Kontekst[Strona 77, 4]	Lokalizacja[Strona 77, 5]	Flagi[Strona 77, 8]	Dodatkowe stany[Strona 77, 6]
GNU gettext	dwujęzyczny	tak	tak	tak	tak	Tak[9]	wymaga edycji
Jednojęzyczny gettext	mono	tak	tak	tak	tak	Tak[Strona 77, 9]	wymaga edycji
XLIFF	oba	tak	tak	tak	tak	tak[10]	wymaga edycji, zatwierdzono
Właściwości Java	oba	nie	tak	nie	nie	nie	
Pliki lang dla mi18n	mono	nie	tak	nie	nie	nie	
Właściwości GWT	mono	tak	tak	nie	nie	nie	
Tłumaczenia Joomli	mono	nie	tak	nie	tak	nie	
Qt Linguist .ts	oba	tak	tak	nie	tak	tak[Strona 77, 10]	wymaga edycji
Android string resources	mono	tak	tak[7]	nie	nie	tak[Strona 77, 10]	
Ciągi Apple iOS	oba	nie	tak	nie	nie	nie	
Ciągi PHP	mono	nie[11]	tak	nie	nie	nie	
Plik JSON	mono	nie	nie	nie	nie	nie	
JSON i18następne pliki	mono	tak	nie	nie	nie	nie	
go-i18n pliki JSON	mono	tak	nie	nie	nie	nie	
Plik ARB	mono	tak	tak	nie	nie	nie	
WebExtension JSON	mono	tak	tak	nie	nie	nie	

ciąg dalszy na następnej stronie

Tabela 1 – kontynuacja poprzedniej strony

Format	Lingualność[Strona 77, 1]	Liczba mnoga[Strona 77, 2]	Opisy[Strona 77, 3]	Kontekst[Strona 77, 4]	Lokalizacja[Strona 77, 5]	Flagi[Strona 77, 8]	Dodatkowe stany[Strona 77, 6]
Pliki zasobów .XML	mono	nie	tak	nie	nie	tak[Strona 77, 10]	
Pliki Resource-Dictionary	mono	nie	nie	nie	nie	tak[Strona 77, 10]	
Pliki CSV	oba	nie	tak	tak	tak	nie	wymaga edycji
Pliki YAML	mono	nie	tak	nie	nie	nie	
Pliki Ruby YAML	mono	tak	tak	nie	nie	nie	
Pliki DTD	mono	nie	nie	nie	nie	nie	
Jednolity plik XML	mono	nie	nie	nie	nie	tak[Strona 77, 10]	
Pliki RC systemu Windows	mono	nie	tak	nie	nie	nie	
Excel Open XML	mono	nie	tak	tak	tak	nie	wymaga edycji
Pliki metadanych sklepu z aplikacjami	mono	nie	nie	nie	nie	nie	
Pliki napisów	mono	nie	nie	nie	tak	nie	
Pliki HTML	mono	nie	nie	nie	nie	nie	
Format OpenDocument	mono	nie	nie	nie	nie	nie	
Format IDML	mono	nie	nie	nie	nie	nie	
Tłumaczenia INI	mono	nie	nie	nie	nie	nie	
Tłumaczenia Inno Setup INI	mono	nie	nie	nie	nie	nie	
Format TermBase eXchange	dwujęzyczny	nie	tak	nie	nie	tak[Strona 77, 10]	
Pliki tekstowe	mono	nie	nie	nie	nie	nie	
Format stringsdict	mono	tak	tak	nie	nie	nie	
Format Fluent	mono	nie[12]	tak	nie	nie	nie	

Ciągi tylko do odczytu

Nowe w wersji 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 *Android string resources*), but can be emulated in others by adding a `read-only` flag, see *Dostosowywanie zachowania za pomocą flag*.

1.10.4 GNU gettext

Najpopularniejszy format do tłumaczenia wolnego oprogramowania.

Contextual info stored in the file is supported by adjusting its headers or linking to corresponding source files.

The bilingual gettext PO file typically looks like this:

```
#: 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"
```

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`po/*.po`
Jednojęzyczny bazowy plik języka	*Puste*
Szablon dla nowych tłumaczeń	`po/messages.pot`
Format pliku	*Plik PO Gettext*

Zobacz także:

devel/gettext, devel/sphinx, Gettext on Wikipedia, PO Files, *Zaktualizuj zmienną ALL_LINGUAS w pliku „konfiguracyjnym”*, *Dostosuj wyjście gettext*, *Aktualizuj plik LINGUAS*, *Generuj pliki MO*, *Zaktualizuj pliki PO, aby dopasować POT (msgmerge)*

[1] Zobacz *Formaty dwujęzyczne i jednojęzyczne*
[2] Plurals are necessary to properly localize strings with variable count.
[3] Source string descriptions can be used to pass additional info about the string to translate.
[4] 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).
[5] Location of a string in source code might help proficient translators figure out how the string is used.
[8] Zobacz *Dostosowywanie zachowania za pomocą flag*
[6] Additional states supported by the file format in addition to „Untranslated” and „Translated”.
[9] The gettext type comments are used as flags.
[10] The flags are extracted from the non-standard attribute `weblate-flags` for all XML based formats. Additionally `max-length:N` is supported through the `maxwidth` attribute as defined in the XLIFF standard, see *Określanie flag tłumaczenia*.
[7] XML comment placed before the `<string>` element, parsed as a source string description.
[11] The plurals are supported only for Laravel which uses in string syntax to define them, see Localization in Laravel.
[12] Plurals are handled in the syntax of the strings and not exposed as plurals in Weblate.

Jednojęzyczny gettext

Some projects decide to use gettext as monolingual formats—they code just the IDs in their source code and the string then needs to be translated to all languages, including English. This is supported, though you have to choose this file format explicitly when importing components into Weblate.

The monolingual gettext PO file typically looks like this:

```
#: 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"
```

While the base language file will be:

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

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

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

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`po/*.po`
Jednojęzyczny bazowy plik języka	`po/en.po`
Szablon dla nowych tłumaczeń	`po/messages.pot`
Format pliku	*Plik gettext PO (jednojęzyczny)*

1.10.5 XLIFF

XML-based format created to standardize translation files, but in the end it is one of many standards, in this area.

XML Localization Interchange File Format (XLIFF) is usually used as bilingual, but Weblate supports it as monolingual as well.

Weblate obsługuje XLIFF w kilku wariantach:

Plik tłumaczenia XLIFF
Simple XLIFF file where content of the elements is stored as plain text (all XML elements being escaped).

XLIFF with placeables support
Standard XLIFF supporting placeables and other XML elements.

XLIFF z rozszerzeniami gettext
XLIFF enriched by XLIFF 1.2 Representation Guide for Gettext PO to support plurals.

Zobacz także:

XML Localization Interchange File Format (XLIFF) specification, XLIFF 1.2 Representation Guide for Gettext PO

Stan tłumaczenia

Zmienione w wersji 3.3: Weblate ignored the `state` attribute prior to the 3.3 release.

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.

If the translation string has `approved="yes"`, it will also be imported into Weblate as „Approved", anything else will be imported as „Waiting for review" (which matches the XLIFF specification).

While saving, Weblate doesn't add those attributes unless necessary:

- The `state` attribute is only added in case string is marked as needing edit.
- The `approved` attribute is only added in case string has been reviewed.
- In other cases the attributes are not added, but they are updated in case they are present.

That means that when using the XLIFF format, it is strongly recommended to turn on the Weblate review process, in order to see and change the approved state of strings.

Similarly upon importing such files (in the upload form), you should choose *Import as translated* under *Processing of strings needing edit*.

Zobacz także:

Dedykowani recenzenci

Whitespace and newlines in XLIFF

Generally types or amounts of whitespace is not differentiated between in XML formats. If you want to keep it, you have to add the `xml:space="preserve"` flag to the string.

Na przykład:

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

Określanie flag tłumaczenia

You can specify additional translation flags (see *Dostosowywanie zachowania za pomocą flag*) by using the `weblate-flags` attribute. Weblate also understands `maxwidth` and `font` attributes from the XLIFF specification:

```
<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>
```

The `font` attribute is parsed for font family, size and weight, the above example shows all of that, though only font family is required. Any whitespace in the font family is converted to underscore, so `Source Sans Pro` becomes `Source_Sans_Pro`, please keep that in mind when naming the font group (see *Zarządzanie czcionkami*).

Klucze ciągów znaków

Weblate identifies the units in the XLIFF file by `resname` attribute in case it is present and falls back to `id` (together with `file` tag if present).

The `resname` attribute is supposed to be human friendly identifier of the unit making it more suitable for Weblate to display instead of `id`. The `resname` has to be unique in the whole XLIFF file. This is required by Weblate and is not covered by the XLIFF standard - it does not put any uniqueness restrictions on this attribute.

Typical Weblate *Konfiguracja komponentu* for bilingual XLIFF	
Maska pliku	`localizations/*.xliff`
Jednojęzyczny bazowy plik języka	*Puste*
Szablon dla nowych tłumaczeń	`localizations/en-US.xliff`
Format pliku	*Plik tłumaczeniowy XLIFF*

Typical Weblate *Konfiguracja komponentu* for monolingual XLIFF	
Maska pliku	`localizations/*.xliff`
Jednojęzyczny bazowy plik języka	`localizations/en-US.xliff`
Szablon dla nowych tłumaczeń	`localizations/en-US.xliff`
Format pliku	*Plik tłumaczeniowy XLIFF*

Zobacz także:

XLIFF on Wikipedia, XLIFF, font attribute in XLIFF 1.2, maxwidth attribute in XLIFF 1.2

1.10.6 Właściwości Java

Native Java format for translations.

Java properties are usually used as monolingual translations.

Weblate supports ISO-8859-1, UTF-8 and UTF-16 variants of this format. All of them support storing all Unicode characters, it is just differently encoded. In the ISO-8859-1, the Unicode escape sequences are used (for example `zkou\u0161ka`), all others encode characters directly either in UTF-8 or UTF-16.

Informacja: Loading escape sequences works in UTF-8 mode as well, so please be careful choosing the correct encoding set to match your application needs.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`src/app/Bundle_*.properties`
Jednojęzyczny bazowy plik języka	`src/app/Bundle.properties`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Właściwości Java (ISO-8859-1)*

Zobacz także:

Właściwości Java w Wikipedii, Mozilla and Java properties files, *Pliki lang dla mi18n*, *Właściwości GWT*, updating-target-files, *Format the Java properties file*, *Wyczyść pliki tłumaczenia*

1.10.7 Pliki lang dla mi18n

Nowe w wersji 4.7.

File format used for JavaScript localization by mi18n. Syntactically it matches *Właściwości Java*.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`*.lang`
Jednojęzyczny bazowy plik języka	`en-US.lang`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik lang dla mi18n*

Zobacz także:

mi18n, Mozilla and Java properties files, *Właściwości Java*, updating-target-files, *Format the Java properties file*, *Wyczyść pliki tłumaczenia*

1.10.8 Właściwości GWT

Native GWT format for translations.

GWT properties are usually used as monolingual translations.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`src/app/Bundle_*.properties`
Jednojęzyczny bazowy plik języka	`src/app/Bundle.properties`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Właściwości GWT*

Zobacz także:

Przewodnik lokalizacji GWT, Samouczek internacjonalizacji GWT, Mozilla and Java properties files, updating-target-files, *Format the Java properties file*, *Wyczyść pliki tłumaczenia*

1.10.9 Tłumaczenia INI

Nowe w wersji 4.1.

INI file format for translations.

INI translations are usually used as monolingual translations.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`language/*.ini`
Jednojęzyczny bazowy plik języka	`language/en.ini`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik INI*

Informacja: Weblate only extracts keys from sections within an INI file. In case your INI file lacks sections, you might want to use *Tłumaczenia Joomli* or *Właściwości Java* instead.

Zobacz także:

INI Files, *Właściwości Java*, *Tłumaczenia Joomli*, *Tłumaczenia Inno Setup INI*

1.10.10 Tłumaczenia Inno Setup INI

Nowe w wersji 4.1.

Format pliku Inno Setup INI dla tłumaczeń.

Tłumaczenia Inno Setup INI są zwykle używane jako tłumaczenia jednojęzyczne.

Informacja: The only notable difference to *Tłumaczenia INI* is in supporting `%n` and `%t` placeholders for line break and tab.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`language/*.islu`
Jednojęzyczny bazowy plik języka	`language/en.islu`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik Inno Setup INI*

Informacja: Only Unicode files (`.islu`) are currently supported, ANSI variant (`.isl`) is currently not supported.

Zobacz także:

INI Files, *Tłumaczenia Joomli*, *Tłumaczenia INI*

1.10.11 Tłumaczenia Joomli

Nowe w wersji 2.12.

Native Joomla format for translations.

Joomla translations are usually used as monolingual translations.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`language/*/com_foobar.ini`
Jednojęzyczny bazowy plik języka	`language/en-GB/com_foobar.ini`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik językowy Joomla*

Zobacz także:

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

1.10.12 Qt Linguist .ts

Translation format used in Qt based applications.

Qt Linguist files are used as both bilingual and monolingual translations.

Typical Weblate *Konfiguracja komponentu* when using as bilingual	
Maska pliku	`i18n/app.*.ts`
Jednojęzyczny bazowy plik języka	*Puste*
Szablon dla nowych tłumaczeń	`i18n/app.de.ts`
Format pliku	*Plik tłumaczenia języka Qt Linguist*

Typical Weblate *Konfiguracja komponentu* when using as monolingual	
Maska pliku	`i18n/app.*.ts`
Jednojęzyczny bazowy plik języka	`i18n/app.en.ts`
Szablon dla nowych tłumaczeń	`i18n/app.en.ts`
Format pliku	*Plik tłumaczenia języka Qt Linguist*

Zobacz także:

Qt Linguist manual, Qt .ts, *Formaty dwujęzyczne i jednojęzyczne*

1.10.13 Android string resources

Android specific file format for translating applications.

Android string resources are monolingual, the *Jednojęzyczny bazowy plik języka* is stored in a different location from the other files – `res/values/strings.xml`.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`res/values-*/strings.xml`
Jednojęzyczny bazowy plik języka	`res/values/strings.xml`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Android String Resource*

Zobacz także:

Android string resources documentation, Android string resources

---

Informacja: Android *string-array* structures are not currently supported. To work around this, you can break your string arrays apart:

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

wygląda tak:

```
<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>
```

The *string-array* that points to the *string* elements should be stored in a different file, and not be made available for translation.

This script may help pre-process your existing strings.xml files and translations: https://gist.github.com/paour/11291062

1.10.14 Ciągi Apple iOS

File format typically used for translating Apple iOS applications, but also standardized by PWG 5100.13 and used on NeXTSTEP/OpenSTEP.

Ciągi Apple iOS są zwykle używane jako jednojęzyczne.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`Resources/*.lproj/Localizable.strings`
Jednojęzyczny bazowy plik języka	`Resources/en.lproj/Localizable.strings` lub `Resources/Base.lproj/Localizable.strings`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Ciągi iOS (UTF-8)*

Zobacz także:

Format stringsdict, Apple „strings files” documentation, Message Catalog File Format in PWG 5100.13, Mac OSX strings

1.10.15 Ciągi PHP

PHP translations are usually monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Przykładowy plik:

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

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`lang/*/texts.php`
Jednojęzyczny bazowy plik języka	`lang/en/texts.php`
Szablon dla nowych tłumaczeń	`lang/en/texts.php`
Format pliku	*Ciągi PHP*

Ciągi znaków Laravel PHP

Zmienione w wersji 4.1.

The Laravel PHP localization files are supported as well with plurals:

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

Zobacz także:

PHP, Localization in Laravel

1.10.16 Plik JSON

Nowe w wersji 2.0.

Zmienione w wersji 2.16: Since Weblate 2.16 and with translate-toolkit at-least 2.2.4, nested structure JSON files are supported as well.

Zmienione w wersji 4.3: The structure of JSON file is properly preserved even for complex situations which were broken in prior releases.

JSON format is used mostly for translating applications implemented in JavaScript.

Weblate currently supports several variants of JSON translations:

- Simple key / value files, used for example by *vue-i18n* or *react-intl*.
- Pliki z zagnieżdżonymi kluczami.
- *JSON i18następne pliki*
- *go-i18n pliki JSON*
- *WebExtension JSON*
- *Plik ARB*

JSON translations are usually monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Przykładowy plik:

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

Nested files are supported as well (see above for requirements), such a file can look like:

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

Podpowiedź: The *JSON file* and *JSON nested structure file* can both handle same type of files. Both preserve existing JSON structure when translating.

The only difference between them is when adding new strings using Weblate. The nested structure format parses the newly added key and inserts the new string into the matching structure. For example `app.name` key is inserted as:

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

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`langs/translation-*.json`
Jednojęzyczny bazowy plik języka	`langs/translation-en.json`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*JSON nested structure file*

Zobacz także:

JSON, updating-target-files, *Dostosuj wyjście JSON*, *Wyczyść pliki tłumaczenia*,

1.10.17 JSON i18następne pliki

Zmienione w wersji 2.17: Since Weblate 2.17 and with translate-toolkit at-least 2.2.5, i18next JSON files with plurals are supported as well.

i18next is an internationalization framework written in and for JavaScript. Weblate supports its localization files with features such as plurals.

i18next translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Informacja: Weblate supports the i18next JSON v3 format. The v2 and v1 variants are mostly compatible, with exception of how plurals are handled.

The v4 variant uses different approach for storing plurals and is currently not supported.

Przykładowy plik:

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

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`langs/*.json`
Jednojęzyczny bazowy plik języka	`langs/en.json`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*i18next Plik JSON*

Zobacz także:

JSON, Format JSON i18next, updating-target-files, *Dostosuj wyjście JSON*, *Wyczyść pliki tłumaczenia*

1.10.18 go-i18n pliki JSON

Nowe w wersji 4.1.

go-i18n translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Informacja: Weblate supports the go-i18n JSON v1 format, for flat JSON formats please use *Plik JSON*. The v2 format with hash is currently not supported.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`langs/*.json`
Jednojęzyczny bazowy plik języka	`langs/en.json`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*go-i18n Plik JSON*

Zobacz także:

JSON, go-i18n, updating-target-files, *Dostosuj wyjście JSON*, *Wyczyść pliki tłumaczenia*,

1.10.19 Plik ARB

Nowe w wersji 4.1.

ARB translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`lib/l10n/intl_*.arb`
Jednojęzyczny bazowy plik języka	`lib/l10n/intl_en.arb`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik ARB*

Zobacz także:

JSON, Application Resource Bundle Specification, Internationalizing Flutter apps, updating-target-files, *Dostosuj wyjście JSON*, *Wyczyść pliki tłumaczenia*

1.10.20 WebExtension JSON

Nowe w wersji 2.16: This is supported since Weblate 2.16 and with translate-toolkit at-least 2.2.4.

File format used when translating extensions for Mozilla Firefox or Google Chromium.

Informacja: 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.

Przykładowy plik:

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

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
        "example": "5"
      }
    }
  },
  "try": {
    "message": "",
    "description": "Description"
  },
  "thanks": {
    "message": "",
    "description": "Description"
  }
}
```

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`_locales/*/messages.json`
Jednojęzyczny bazowy plik języka	`_locales/en/messages.json`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*WebExtension plik JSON*

Zobacz także:

JSON, Google chrome.i18n, Mozilla Extensions Internationalization

1.10.21 Pliki zasobów .XML

Nowe w wersji 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.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`Resources/Language.*.resx`
Jednojęzyczny bazowy plik języka	`Resources/Language.resx`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik zasobów .Net*

Zobacz także:

.NET Resource files (.resx), updating-target-files, *Wyczyść pliki tłumaczenia*

1.10.22 Pliki ResourceDictionary

Nowe w wersji 4.13.

ResourceDictionary is a monolingual XML file format used to package localizable string resources for Windows Presentation Foundation (WPF) applications.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`Languages/*.xaml`
Jednojęzyczny bazowy plik języka	`Language/en.xaml`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik ResourceDictionary*

Zobacz także:

Flat XML, *Jednolity plik XML*, updating-target-files, *Wyczyść pliki tłumaczenia*

1.10.23 Pliki CSV

Nowe w wersji 2.4.

CSV files can contain a simple list of source and translation. Weblate supports the following files:

- Files with header defining fields (`location`, `source`, `target`, `ID`, `fuzzy`, `context`, `translator_comments`, `developer_comments`). This is the recommended approach, as it is the least error prone. Choose *CSV file* as a file format.
- Files with two fields—source and translation (in this order). Choose *Simple CSV file* as a file format.
- Headerless files with fields in order defined by the translate-toolkit: `location`, `source`, `target`, `ID`, `fuzzy`, `context`, `translator_comments`, `developer_comments`. Choose *CSV file* as a file format.
- Remember to define *Jednojęzyczny bazowy plik języka* when your files are monolingual (see *Formaty dwujęzyczne i jednojęzyczne*).

Podpowiedź: By default, the CSV format does autodetection of file encoding. This can be unreliable in some corner cases and causes performance penalty. Please choose file format variant with encoding to avoid this (for example *CSV file (UTF-8)*).

Ostrzeżenie: 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.

Przykładowy plik:

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

Typical Weblate *Konfiguracja komponentu* for bilingual CSV	
Maska pliku	`locale/*.csv`
Jednojęzyczny bazowy plik języka	*Puste*
Szablon dla nowych tłumaczeń	`locale/en.csv`
Format pliku	*Plik CSV*

Typical Weblate *Konfiguracja komponentu* for monolingual CSV	
Maska pliku	`locale/*.csv`
Jednojęzyczny bazowy plik języka	`locale/en.csv`
Szablon dla nowych tłumaczeń	`locale/en.csv`
Format pliku	*Zwykły plik CSV*

Multivalue CSV file

Nowe w wersji 4.13.

This variant of the CSV files allows storing multiple translations per string.

Zobacz także:

CSV

1.10.24 Pliki YAML

Nowe w wersji 2.9.

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

Example of a YAML file:

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

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`translations/messages.*.yml`
Jednojęzyczny bazowy plik języka	`translations/messages.en.yml`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Pliki YAML*

Zobacz także:

YAML, *Pliki Ruby YAML*

1.10.25 Pliki Ruby YAML

Nowe w wersji 2.9.

Ruby i18n YAML files with language as root node.

Example Ruby i18n YAML file:

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

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`translations/messages.*.yml`
Jednojęzyczny bazowy plik języka	`translations/messages.en.yml`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik Ruby YAML*

Zobacz także:

YAML, *Pliki YAML*

1.10.26 Pliki DTD

Nowe w wersji 2.18.

Przykładowy plik DTD:

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

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`locale/*.dtd`
Jednojęzyczny bazowy plik języka	`locale/en.dtd`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Pliki DTD*

Zobacz także:

Mozilla DTD format

1.10.27 Jednolity plik XML

Nowe w wersji 3.9.

Example of a flat XML file:

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

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`locale/*.xml`
Jednojęzyczny bazowy plik języka	`locale/en.xml`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Jednolity plik XML*

Zobacz także:

Flat XML

1.10.28 Pliki RC systemu Windows

Zmienione w wersji 4.1: Support for Windows RC files has been rewritten.

Informacja: Support for this format is currently in beta, feedback from testing is welcome.

Przykładowy plik RC systemu Windows:

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT

STRINGTABLE
BEGIN
    IDS_MSG1                "Hello, world!\n"
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    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
```

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`lang/*.rc`
Jednojęzyczny bazowy plik języka	`lang/en-US.rc`
Szablon dla nowych tłumaczeń	`lang/en-US.rc`
Format pliku	*Pliki RC*

Zobacz także:

Windows RC files

1.10.29 Pliki metadanych sklepu z aplikacjami

Nowe w wersji 3.5.

Metadata used for publishing apps in various app stores can be translated. Currently the following tools are compatible:

- Triple-T gradle-play-publisher
- Fastlane
- F-Droid

The metadata consists of several textfiles, which Weblate will present as separate strings to translate.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`fastlane/android/metadata/*`
Jednojęzyczny bazowy plik języka	`fastlane/android/metadata/en-US`
Szablon dla nowych tłumaczeń	`fastlane/android/metadata/en-US`
Format pliku	*Pliki metadanych sklepu z aplikacjami*

Podpowiedź: In case you don't want to translate certain strings (for example changelogs), mark them read-only (see *Dostosowywanie zachowania za pomocą flag*). This can be automated by the *Edycja zbiorcza*.

1.10.30 Pliki napisów

Nowe w wersji 3.7.

Weblate może tłumaczyć różne pliki z napisami:

- Plik napisów SubRip (`*.srt`)
- Plik napisów MicroDVD (`*.sub`)
- Advanced Substation Alpha subtitles file (`*.ass`)
- Substation Alpha subtitle file (`*.ssa`)

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`path/*.srt`
Jednojęzyczny bazowy plik języka	`path/en.srt`
Szablon dla nowych tłumaczeń	`path/en.srt`
Format pliku	*Plik z napisami SubRip*

Zobacz także:

Subtitles

1.10.31 Excel Open XML

Nowe w wersji 3.2.

Excel Open XML (.xlsx) files can be imported and exported.

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.

1.10.32 Pliki HTML

Nowe w wersji 4.1.

Informacja: Support for this format is currently in beta, feedback from testing is welcome.

The translatable content is extracted from the HTML files and offered for the translation.

Zobacz także:

HTML

1.10.33 Pliki tekstowe

Nowe w wersji 4.6.

Informacja: Support for this format is currently in beta, feedback from testing is welcome.

The translatable content is extracted from the plain text files and offered for the translation. Each paragraph is translated as a separate string.

Istnieją trzy odmiany tego formatu:

- Zwykły plik tekstowy
- Plik tekstowy DokuWiki
- Plik tekstowy MediaWiki

Zobacz także:

Simple Text Documents

1.10.34 Format OpenDocument

Nowe w wersji 4.1.

Informacja: Support for this format is currently in beta, feedback from testing is welcome.

The translatable content is extracted from the OpenDocument files and offered for the translation.

Zobacz także:

OpenDocument Format

1.10.35 Format IDML

Nowe w wersji 4.1.

Informacja: Support for this format is currently in beta, feedback from testing is welcome.

The translatable content is extracted from the Adobe InDesign Markup Language files and offered for the translation.

1.10.36 Format TermBase eXchange

Nowe w wersji 4.5.

TBX is an XML format for the exchange of terminology data.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`tbx/*.tbx`
Jednojęzyczny bazowy plik języka	*Puste*
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik TermBase eXchange*

Zobacz także:

TBX w Wikipedii, TBX, *Słownik*

1.10.37 Format stringsdict

Nowe w wersji 4.8.

Informacja: Support for this format is currently in beta, feedback from testing is welcome.

XML based format used by Apple which is able to store plural forms of a string.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`Resources/*.lproj/Localizable.stringsdict`
Jednojęzyczny bazowy plik języka	`Resources/en.lproj/Localizable.stringsdict` lub `Resources/Base.lproj/Localizable.stringsdict`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	*Plik stringsdict*

Zobacz także:

Ciągi Apple iOS, Format pliku Stringsdict

1.10.38 Format Fluent

Nowe w wersji 4.8.

Informacja: Support for this format is currently in beta, feedback from testing is welcome.

Fluent is a monolingual text format that focuses on asymmetric localization: a simple string in one language can map to a complex multi-variant translation in another language.

Typowy Weblate *Konfiguracja komponentu*	
Maska pliku	`locales/*/messages.ftl`
Jednojęzyczny bazowy plik języka	`locales/en/messages.ftl`
Szablon dla nowych tłumaczeń	*Puste*
Format pliku	`Plik Fluent`

Zobacz także:

Strona Project Fluent

1.10.39 Obsługa innych formatów

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.

To add support for a new format, the preferred approach is to first implement support for it in the translate-toolkit.

Zobacz także:

Translation Related File Formats

1.11 Integracja kontroli wersji

Weblate currently supports *Git* (with extended support for *GitHub pull requests*, *Gerrit* and *Subversion*) and *Mercurial* as version control back-ends.

1.11.1 Dostęp do repozytoriów

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.

Accessing repositories from Hosted Weblate

For Hosted Weblate there is a dedicated push user registered on GitHub, Bitbucket, Codeberg and GitLab (with the username *weblate*, e-mail `hosted@weblate.org` and, 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 on the Weblate side.

The *weblate* user on GitHub accepts invitations automatically within five minutes. Manual processing might be needed on the other services, so please be patient.

Once the *weblate* user is added, you can configure *Repozytorium kodu źródłowego* and *URL repozytorium dla push* using the SSH protocol (for example `git@github.com:WeblateOrg/weblate.git`).

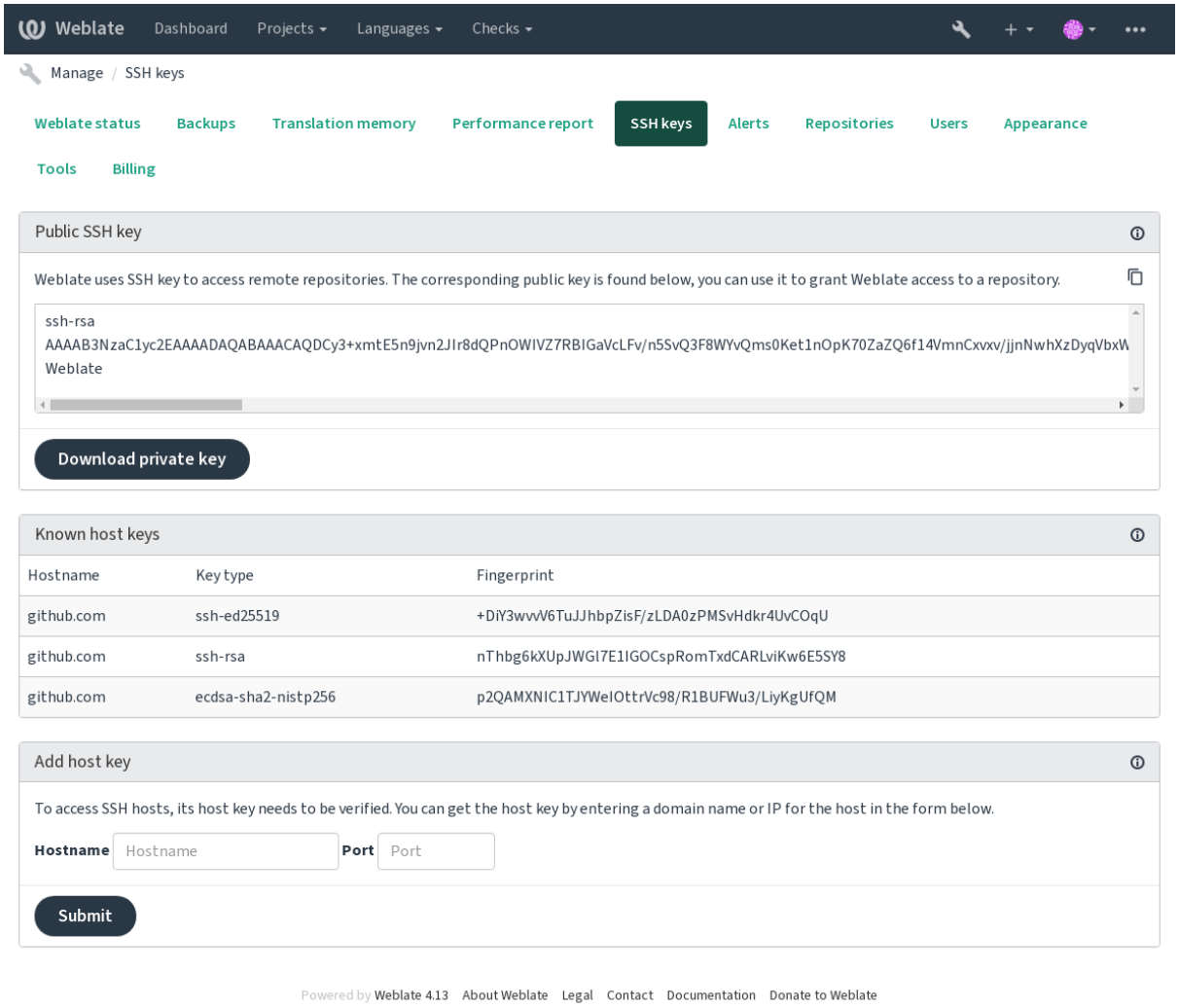
Repozytoria SSH

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

Ostrzeżenie: On GitHub, each key can only be used once, see *Repozytoria GitHub* and *Accessing repositories from 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 *Weryfikacja kluczy hosta SSH*).

In case adjustment is needed, do so from the Weblate admin interface:



Weblate Klucz SSH

The Weblate public key is visible to all users browsing the *About* page.

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

Informacja: The corresponding private SSH key can not currently have a password, so make sure it is well protected.

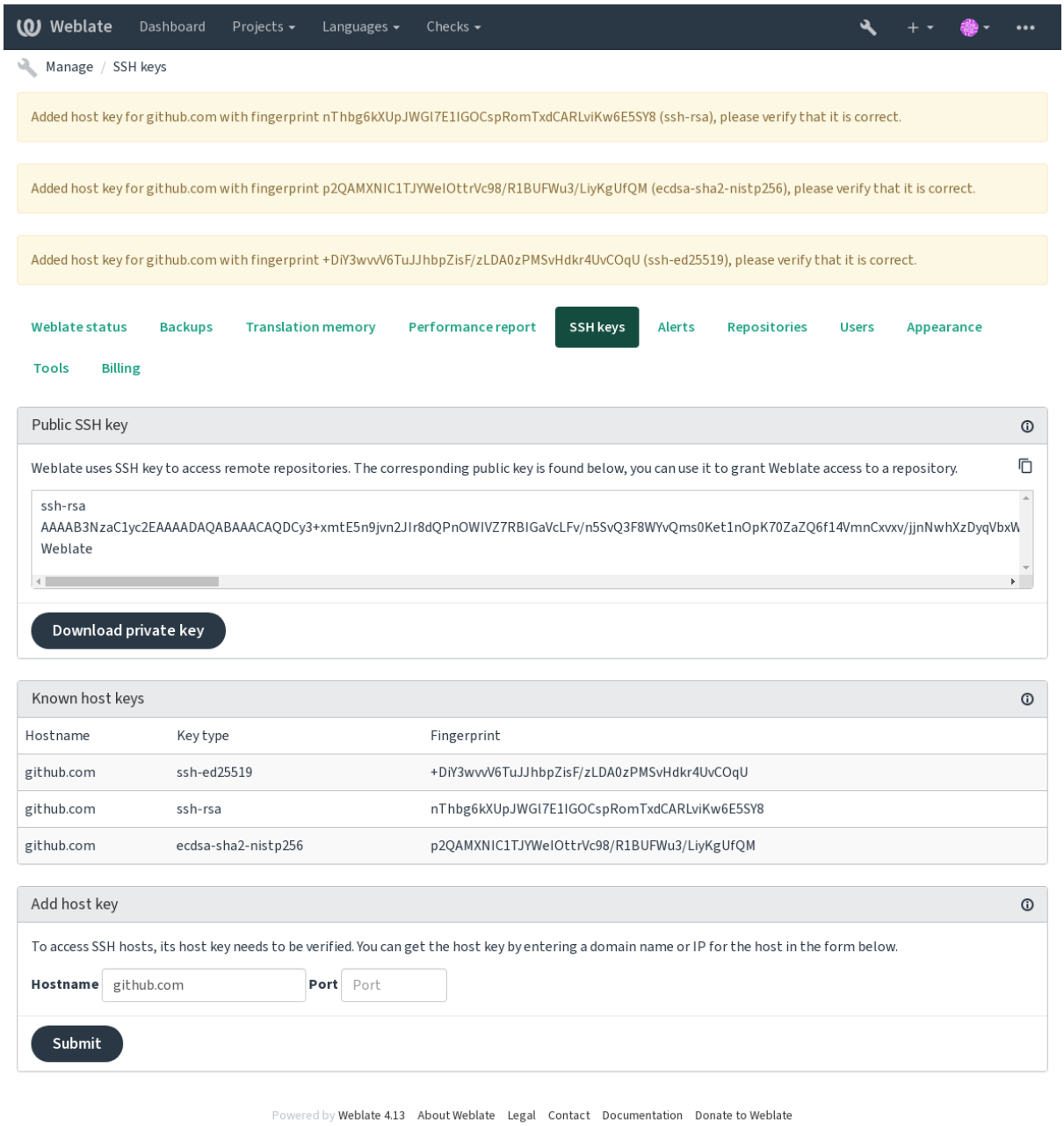
Podpowiedź: Make a backup of the generated private Weblate SSH key.

Weryfikacja kluczy hosta SSH

Weblate automatically stores the SSH host keys on first access and remembers them for further use.

In case you want to verify the key fingerprint before connecting to the repository, add 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.

The added keys with fingerprints are shown in the confirmation message:



Repozytoria GitHub

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

In case the *Wypchnij gałąź* 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 *Weblate Klucz SSH*) 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.

Zobacz także:

Accessing repositories from Hosted Weblate

Wewnętrzne adresy URL Weblate

Share one repository setup between different components by referring to its placement as `weblate://project/component` in other(linked) components. This way linked components use the VCS repository configuration of the main(referenced) component.

Ostrzeżenie: Removing main component also removes linked components.

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

Powody, dla których warto skorzystać z tej funkcji:

- Saves disk space on the server, the repository is stored just once.
- Makes the updates faster, only one repository is updated.
- There is just single exported repository with Weblate translations (see *Eksporter Git*).
- Some add-ons can operate on multiple components sharing one repository, for example *Zesquashowane commity na Git*.

Repozytoria 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`.

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

Korzystanie z proxy

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
```

Informacja: The proxy configuration needs to be done under user running Weblate (see also *Uprawnienia systemu plików*) and with `HOME=$DATA_DIR/home` (see *DATA_DIR*), otherwise Git executed by Weblate will not use it.

Zobacz także:

The cURL manpage, Git config documentation

1.11.2 Git

Podpowiedź: Weblate needs Git 2.12 or newer.

Zobacz także:

See *Dostęp do repozytoriów* for info on how to access different kinds of repositories.

Git z wymuszonym wypchnięciem

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.

Ostrzeżenie: Use with caution, as this easily leads to lost commits in your upstream repository.

Dostosowywanie konfiguracji 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.

To clone the `gnuhello` project from Launchpad using Bazaar:

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

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

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

Ostrzeżenie: 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 pull requests

Nowe w wersji 2.3.

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

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

You need to configure API credentials (`GITHUB_CREDENTIALS`) in the Weblate settings to make this work. Once configured, you will see a *GitHub* option when selecting *System kontroli wersji*.

Zobacz także:

Wypychanie zmian z Weblate, `GITHUB_USERNAME`, `GITHUB_TOKEN`, `GITHUB_CREDENTIALS`

1.11.4 GitLab merge requests

Nowe w wersji 3.9.

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

There is no need to use this to 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 merge requests* creates merge request.

You need to configure API credentials (`GITLAB_CREDENTIALS`) in the Weblate settings to make this work. Once configured, you will see a *GitLab* option when selecting *System kontroli wersji*.

Zobacz także:

Wypychanie zmian z Weblate, `GITLAB_USERNAME`, `GITLAB_TOKEN`, `GITLAB_CREDENTIALS`

1.11.5 Gitea pull requests

Nowe w wersji 4.12.

This just adds a thin layer atop *Git* using the Gitea API to allow pushing translation changes as pull requests instead of pushing directly to the repository.

There is no need to use this to 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 *Gitea pull requests* creates pull requests.

You need to configure API credentials (`GITEA_CREDENTIALS`) in the Weblate settings to make this work. Once configured, you will see a *Gitea* option when selecting *System kontroli wersji*.

Zobacz także:

Wypychanie zmian z Weblate, `GITEA_USERNAME`, `GITEA_TOKEN`, `GITEA_CREDENTIALS`

1.11.6 Pagure merge requests

Nowe w wersji 4.3.2.

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

There is no need to use this to 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 *Pagure merge requests* creates merge request.

You need to configure API credentials (`PAGURE_CREDENTIALS`) in the Weblate settings to make this work. Once configured, you will see a *Pagure* option when selecting *System kontroli wersji*.

Zobacz także:

Wypychanie zmian z Weblate, `PAGURE_USERNAME`, `PAGURE_TOKEN`, `PAGURE_CREDENTIALS`

1.11.7 Gerrit

Nowe w wersji 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 them directly to the repository.

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

1.11.8 Mercurial

Nowe w wersji 2.1.

Mercurial is another VCS you can use directly in Weblate.

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

Zobacz także:

See *Dostęp do repozytoriów* for info on how to access different kinds of repositories.

1.11.9 Subversion

Nowe w wersji 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.

Informacja: 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.

Zmienione w wersji 2.19: Before this, only repositories using the standard layout were supported.

Poświadczenia Subversion

Weblate expects you to have accepted the certificate up-front (and your credentials if needed). 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
```

Zobacz także:

`DATA_DIR`

1.11.10 Pliki lokalne

1.11.11 Git

Podpowiedź: Underneath, this uses *Git*. It requires Git installed and allows you to switch to using Git natively with full history of your translations.

Nowe w wersji 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 REST API Weblate

Nowe w wersji 2.6: The REST API is available since Weblate 2.6.

The API is accessible on the `/api/` URL and it is based on Django REST framework. You can use it directly or by *Klient Weblate*.

1.12.1 Uwierzytelnianie i parametry ogólne

The public project API is available without authentication, though unauthenticated requests are heavily throttled (by default to 100 requests per day), so it is recommended to use authentication. The authentication uses a token, which you can get in your profile. Use it in the `Authorization` header:

ANY /

Generic request behaviour for the API, the headers, status codes and parameters here apply to all endpoints as well.

Parametry zapytania

- **format** – Response format (overrides Accept). Possible values depends on REST framework setup, by default `json` and `api` are supported. The latter provides web browser interface for API.
- **page** – Returns given page of paginated results (use *next* and *previous* fields in response to automate the navigation).

Nagłówki żądań

- Accept – the response content type depends on Accept header

- Authorization – opcjonalny token do uwierzytelniania jako `Authorization: Token TWÓJ-TOKEN`

Nagłówki odpowiedzi

- Content-Type – this depends on Accept header of request
- Allow – list of allowed HTTP methods on object

Odpowiedź Obiekt JSON

- **detail** (*string*) – pełny opis wyniku (dla kodów statusu HTTP innych niż 200 OK)
- **count** (*int*) – total item count for object lists
- **next** (*string*) – next page URL for object lists
- **previous** (*string*) – previous page URL for object lists
- **results** (*array*) – wyniki dla list obiektów
- **url** (*string*) – URL to access this resource using API
- **web_url** (*string*) – URL to access this resource using web browser

Kody stanu

- 200 OK – gdy żądanie zostało poprawnie obsłużone
- 201 Created – gdy nowy obiekt został pomyślnie utworzony
- 204 No Content – gdy obiekt został pomyślnie usunięty
- 400 Bad Request – gdy brakuje parametrów formularza
- 403 Forbidden – gdy odmowa dostępu
- 429 Too Many Requests – when throttling is in place

Tokeny uwierzytelniające

Zmienione w wersji 4.10: Project scoped tokens were introduced in the 4.10 release.

Each user has his personal access token which can be obtained in the user profile. Newly generated user tokens have the `wlu_` prefix.

It is possible to create project scoped tokens for API access to given project only. These tokens can be identified by the `wlp_` prefix.

Przykłady uwierzytelniania

Przykładowe żądanie:

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

Przykładowa odpowiedź:

```
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
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
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/"
}
```

Przykład CURL:

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

Przykłady przekazywania parametrów

For the POST method the parameters can be specified either as form submission (*application/x-www-form-urlencoded*) or as JSON (*application/json*).

Przykład żądania formularza:

```
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
```

Przykład żądania 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"}
```

Przykład CURL:

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

Przykład JSON w CURL:

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

Ograniczanie szybkości interfejsu API

The API requests are rate limited; the default configuration limits it to 100 requests per day for anonymous users and 5000 requests per hour for authenticated users.

Rate limiting can be adjusted in the `settings.py`; see Throttling in Django REST framework documentation for more details how to configure it.

In the Docker container this can be configured using *WEBLATE_API_RATELIMIT_ANON* and *WEBLATE_API_RATELIMIT_USER*.

The status of rate limiting is reported in following headers:

`X-RateLimit-Limit`	Rate limiting limit of requests to perform
`X-RateLimit-Remaining`	Pozostały limit żądań
`X-RateLimit-Reset`	Number of seconds until ratelimit window resets

Zmienione w wersji 4.1: Dodano nagłówki statusu ograniczenia szybkości.

Zobacz także:

Ograniczenie szybkości, *Ograniczenie szybkości*, *WEBLATE_API_RATELIMIT_ANON*, *WEBLATE_API_RATELIMIT_USER*

1.12.2 Punkt wejścia interfejsu API

GET /api/

The API root entry point.

Przykładowe żądanie:

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

Przykładowa odpowiedź:

```
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 Użytkownicy

Nowe w wersji 4.0.

GET /api/users/

Returns a list of users if you have permissions to see manage users. If not, then you get to see only your own details.

Zobacz także:

Users object attributes are documented at *GET /api/users/(str:username)/*.

POST /api/users/

Utwórz nowego użytkownika.

Parametry

- **username** (*string*) – Nazwa użytkownika
- **full_name** (*string*) – Pełna nazwa użytkownika
- **email** (*string*) – E-mail użytkownika
- **is_superuser** (*boolean*) – Czy użytkownik jest superużytkownikiem? (opcjonalnie)
- **is_active** (*boolean*) – Czy użytkownik jest aktywny? (opcjonalnie)
- **is_bot** (*boolean*) – Is user bot? (optional) (used for project scoped tokens)

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

Zwraca informacje o użytkownikach.

Parametry

- **username** (*string*) – Nazwa użytkownika

Odpowiedź Obiekt JSON

- **username** (*string*) – nazwa użytkownika
- **full_name** (*string*) – pełna nazwa użytkownika
- **email** (*string*) – e-mail użytkownika
- **is_superuser** (*boolean*) – whether the user is a super user
- **is_active** (*boolean*) – czy użytkownik jest aktywny
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – data utworzenia użytkownika
- **groups** (*array*) – link to associated groups; see *GET /api/groups/(int:id)/*

Przykładowe dane JSON:

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

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    "statistics_url": "http://example.com/api/users/exampleusername/statistics/
↪"
}
```

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

Zmienia parametry użytkownika.

Parametry

- **username** (*string*) – Nazwa użytkownika

Odpowiedź Obiekt JSON

- **username** (*string*) – nazwa użytkownika
- **full_name** (*string*) – pełna nazwa użytkownika
- **email** (*string*) – e-mail użytkownika
- **is_superuser** (*boolean*) – whether the user is a super user
- **is_active** (*boolean*) – czy użytkownik jest aktywny
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – data utworzenia użytkownika

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

Zmienia parametry użytkownika.

Parametry

- **username** (*string*) – Nazwa użytkownika

Odpowiedź Obiekt JSON

- **username** (*string*) – nazwa użytkownika
- **full_name** (*string*) – pełna nazwa użytkownika
- **email** (*string*) – e-mail użytkownika
- **is_superuser** (*boolean*) – whether the user is a super user
- **is_active** (*boolean*) – czy użytkownik jest aktywny
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – data utworzenia użytkownika

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

Deletes all user information and marks the user inactive.

Parametry

- **username** (*string*) – Nazwa użytkownika

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

Powiąż grupy z użytkownikiem.

Parametry

- **username** (*string*) – Nazwa użytkownika

Parametry formularza

- **string group_id** – Unikalny identyfikator grupy

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

List statistics of a user.

Parametry

- **username** (*string*) – Nazwa użytkownika

Odpowiedź Obiekt JSON

- **translated** (*int*) – Liczba tłumaczeń według użytkownika
- **suggested** (*int*) – Liczba sugestii według użytkownika
- **uploaded** (*int*) – Liczba przesłańedług użytkownika
- **commented** (*int*) – Liczba komentarzy wedługużytkownika
- **languages** (*int*) – Liczba języków, które użytkownik może przetłumaczyć

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

Wyświetl listę subskrypcji użytkownika.

Parametry

- **username** (*string*) – Nazwa użytkownika

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

Powiąż subskrypcje z użytkownikiem.

Parametry

- **username** (*string*) – Nazwa użytkownika

Żądanie obiektu JSON

- **notification** (*string*) – Nazwa zarejestrowanego powiadomienia
- **scope** (*int*) – Scope of notification from the available choices
- **frequency** (*int*) – Wybór częstotliwości powiadomień

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

Get a subscription associated with a user.

Parametry

- **username** (*string*) – Nazwa użytkownika
- **subscription_id** (*int*) – ID zarejestrowanego powiadomienia

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

Edit a subscription associated with a user.

Parametry

- **username** (*string*) – Nazwa użytkownika
- **subscription_id** (*int*) – ID zarejestrowanego powiadomienia

Żądanie obiektu JSON

- **notification** (*string*) – Nazwa zarejestrowanego powiadomienia
- **scope** (*int*) – Scope of notification from the available choices
- **frequency** (*int*) – Wybór częstotliwości powiadomień

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

Edit a subscription associated with a user.

Parametry

- **username** (*string*) – Nazwa użytkownika
- **subscription_id** (*int*) – ID zarejestrowanego powiadomienia

Żądanie obiektu JSON

- **notification** (*string*) – Nazwa zarejestrowanego powiadomienia
- **scope** (*int*) – Scope of notification from the available choices
- **frequency** (*int*) – Wybór częstotliwości powiadomień

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

Delete a subscription associated with a user.

Parametry

- **username** (*string*) – Nazwa użytkownika
- **subscription_id** – Nazwa zarejestrowanego powiadomienia
- **subscription_id** – int

1.12.4 Grupy

Nowe w wersji 4.0.

GET /api/groups/

Returns a list of groups if you have permissions to see manage groups. If not, then you get to see only the groups the user is a part of.

Zobacz także:

Group object attributes are documented at *GET /api/groups/(int:id)/*.

POST /api/groups/

Tworzy nową grupę.

Parametry

- **name** (*string*) – Nazwa grupy
- **project_selection** (*int*) – Group of project selection from given options
- **language_selection** (*int*) – Group of languages selected from given options
- **defining_project** (*str*) – link do definiowania projektu, używany dla *Zarządzanie kontrolą dostępu dla poszczególnych projektów*; patrz *GET /api/projects/(string:project)/*

GET /api/groups/(**int:** *id*)**/**

Zwraca informacje o grupie.

Parametry

- **id** (*int*) – Identyfikator grupy

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa grupy
- **project_selection** (*int*) – integer corresponding to group of projects
- **language_selection** (*int*) – integer corresponding to group of languages
- **roles** (*array*) – link to associated roles; see *GET /api/roles/(int:id)/*

- **projects** (*array*) – link to associated projects; see *GET /api/projects/(string:project)/*
- **components** (*array*) – link to associated components; see *GET /api/components/(string:project)/(string:component)/*
- **componentlists** (*array*) – link to associated componentlist; see *GET /api/component-lists/(str:slug)/*
- **defining_project** (*str*) – link do definiowania projektu, używany dla *Zarządzanie kontrolą dostępu dla poszczególnych projektów*; patrz *GET /api/projects/(string:project)/*

Przykładowe dane JSON:

```
{
    "name": "Guests",
    "defining_project": null,
    "project_selection": 3,
    "language_selection": 1,
    "url": "http://example.com/api/groups/1/",
    "roles": [
        "http://example.com/api/roles/1/",
        "http://example.com/api/roles/2/"
    ],
    "languages": [
        "http://example.com/api/languages/en/",
        "http://example.com/api/languages/cs/",
    ],
    "projects": [
        "http://example.com/api/projects/demo1/",
        "http://example.com/api/projects/demo/"
    ],
    "componentlist": "http://example.com/api/component-lists/new/",
    "components": [
        "http://example.com/api/components/demo/weblate/"
    ]
}
```

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

Zmienia parametry grupy.

Parametry

- **id** (*int*) – Identyfikator grupy

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa grupy
- **project_selection** (*int*) – integer corresponding to group of projects
- **language_selection** (*int*) – integer corresponding to group of Languages

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

Zmienia parametry grupy.

Parametry

- **id** (*int*) – Identyfikator grupy

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa grupy
- **project_selection** (*int*) – integer corresponding to group of projects
- **language_selection** (*int*) – integer corresponding to group of languages

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

Usuwa grupę.

Parametry

- **id** (*int*) – Identyfikator grupy

POST /api/groups/(**int:** *id*)**/roles/**

Powiąż role z grupą.

Parametry

- **id** (*int*) – Identyfikator grupy

Parametry formularza

- **string role_id** – Unikatowy identyfikator roli

POST /api/groups/(**int:** *id*)**/components/**

Powiąż komponenty z grupą.

Parametry

- **id** (*int*) – Identyfikator grupy

Parametry formularza

- **string component_id** – Unikatowy identyfikator komponentu

DELETE /api/groups/(**int:** *id*)**/components/**
int: *component_id*

Usuń komponent z grupy.

Parametry

- **id** (*int*) – Identyfikator grupy
- **component_id** (*int*) – Unikatowy identyfikator komponentu

POST /api/groups/(**int:** *id*)**/projects/**

Powiąż projekty z grupą.

Parametry

- **id** (*int*) – Identyfikator grupy

Parametry formularza

- **string project_id** – Unikalny identyfikator projektu

DELETE /api/groups/(**int:** *id*)**/projects/**
int: *project_id*

Usuń projekt z grupy.

Parametry

- **id** (*int*) – Identyfikator grupy
- **project_id** (*int*) – Unikalny identyfikator projektu

POST /api/groups/(**int:** *id*)**/languages/**

Powiąż języki z grupą.

Parametry

- **id** (*int*) – Identyfikator grupy

Parametry formularza

- **string language_code** – Unikatowy kod języka

DELETE /api/groups/(**int:** *id*)**/languages/**
string: *language_code*

Usuń język z grupy.

Parametry

- **id** (*int*) – Identyfikator grupy
- **language_code** (*string*) – Unikatowy kod języka

POST /api/groups/(int: *id*)/componentlists/

Powiąż listy komponentów z grupą.

Parametry

- **id** (*int*) – Identyfikator grupy

Parametry formularza

- **string component_list_id** – Unikatowy identyfikator listy komponentów

DELETE /api/groups/(int: *id*)/componentlists/
int: *component_list_id*

Usuń listę komponentów z grupy.

Parametry

- **id** (*int*) – Identyfikator grupy
- **component_list_id** (*int*) – Unikatowy identyfikator listy komponentów

1.12.5 Role

GET /api/roles/

Returns a list of all roles associated with user. If user is superuser, then list of all existing roles is returned.

Zobacz także:

Roles object attributes are documented at *GET /api/roles/(int:id)/*.

POST /api/roles/

Tworzy nową rolę.

Parametry

- **name** (*string*) – Nazwa roli
- **permissions** (*array*) – List of codenames of permissions

GET /api/roles/(int: *id*)/

Zwraca informacje o roli.

Parametry

- **id** (*int*) – Identyfikator roli

Odpowiedź Obiekt JSON

- **name** (*string*) – Nazwa roli
- **permissions** (*array*) – list of codenames of permissions

Przykładowe dane JSON:

```
{
    "name": "Access repository",
    "permissions": [
        "vcs.access",
        "vcs.view"
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    ],
    "url": "http://example.com/api/roles/1/",
}
```

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

Zmienia parametry roli.

Parametry

- **id** (*int*) – Identyfikator roli

Odpowiedź Obiekt JSON

- **name** (*string*) – Nazwa roli
- **permissions** (*array*) – list of codenames of permissions

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

Zmienia parametry roli.

Parametry

- **id** (*int*) – Identyfikator roli

Odpowiedź Obiekt JSON

- **name** (*string*) – Nazwa roli
- **permissions** (*array*) – list of codenames of permissions

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

Usuwa rolę.

Parametry

- **id** (*int*) – Identyfikator roli

1.12.6 Języki

GET /api/languages/

Returns a list of all languages.

Zobacz także:

Language object attributes are documented at *GET /api/languages/(string:language)/*.

POST /api/languages/

Tworzy nowy język.

Parametry

- **code** (*string*) – Nazwa języka
- **name** (*string*) – Nazwa języka
- **direction** (*string*) – Kierunek tekstu
- **population** (*int*) – Liczba mówców
- **plural** (*object*) – Language plural formula and number

GET /api/languages/(string: *language*)/

Zwraca informacje o języku.

Parametry

- **language** (*string*) – Kod języka

Odpowiedź Obiekt JSON

- **code** (*string*) – Kod języka
- **direction** (*string*) – Kierunek tekstu
- **plural** (*object*) – Object of language plural information
- **aliases** (*array*) – Tablica aliasów dla języka

Żądanie obiektu JSON

- **population** (*int*) – Liczba mówców

Przykładowe dane JSON:

```
{
    "code": "en",
    "direction": "ltr",
    "name": "English",
    "population": 159034349015,
    "plural": {
        "id": 75,
        "source": 0,
        "number": 2,
        "formula": "n != 1",
        "type": 1
    },
    "aliases": [
        "english",
        "en_en",
        "base",
        "source",
        "eng"
    ],
    "url": "http://example.com/api/languages/en/",
    "web_url": "http://example.com/languages/en/",
    "statistics_url": "http://example.com/api/languages/en/statistics/"
}
```

PUT /api/languages/(string: *language*)/

Zmienia parametry języka.

Parametry

- **language** (*string*) – Kod języka

Żądanie obiektu JSON

- **name** (*string*) – Nazwa języka
- **direction** (*string*) – Kierunek tekstu
- **population** (*int*) – Liczba mówców
- **plural** (*object*) – Szczegóły dotyczące liczby mnogiej w języku

PATCH /api/languages/(string: *language*)/

Zmienia parametry języka.

Parametry

- **language** (*string*) – Kod języka

Żądanie obiektu JSON

- **name** (*string*) – Nazwa języka
- **direction** (*string*) – Kierunek tekstu

- **population** (*int*) – Liczba mówców
- **plural** (*object*) – Szczegóły dotyczące liczby mnogiej w języku

DELETE /api/languages/(string: *language*)/

Usuwa język.

Parametry

- **language** (*string*) – Kod języka

GET /api/languages/(string: *language*)/statistics/

Zwraca statystyki dla języka.

Parametry

- **language** (*string*) – Kod języka

Odpowiedź Obiekt JSON

- **total** (*int*) – całkowita liczba ciągów
- **total_words** (*int*) – całkowita liczba słów
- **last_change** (*timestamp*) – ostatnie zmiany w języku
- **recent_changes** (*int*) – całkowita liczba zmian
- **translated** (*int*) – liczba przetłumaczonych ciągów tekstowych
- **translated_percent** (*float*) – procent przetłumaczonych ciągów
- **translated_words** (*int*) – liczba przetłumaczonych słów
- **translated_words_percent** (*int*) – procent przetłumaczonych słów
- **translated_chars** (*int*) – liczba przetłumaczonych znaków
- **translated_chars_percent** (*int*) – procent przetłumaczonych znaków
- **total_chars** (*int*) – liczba całkowitych znaków
- **fuzzy** (*int*) – liczba rozmytych (zaznaczonych do edycji) ciągów
- **fuzzy_percent** (*int*) – percentage of fuzzy (marked for edit) strings
- **failing** (*int*) – number of failing strings
- **failing** – percentage of failing strings

1.12.7 Projekty

GET /api/projects/

Returns a list of all projects.

Zobacz także:

Project object attributes are documented at *GET /api/projects/(string:project)/*.

POST /api/projects/

Nowe w wersji 3.9.

Tworzy nowy projekt.

Parametry

- **name** (*string*) – Nazwa projektu
- **slug** (*string*) – Końcówka adresu URL projektu
- **web** (*string*) – Strona internetowa projektu

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

Zwraca informacje o projekcie.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa projektu
- **slug** (*string*) – końcówka adresu URL projektu
- **web** (*string*) – strona internetowa projektu
- **components_list_url** (*string*) – URL to components list; see *GET /api/projects/(string:project)/components/*
- **repository_url** (*string*) – URL to repository status; see *GET /api/projects/(string:project)/repository/*
- **changes_list_url** (*string*) – URL to changes list; see *GET /api/projects/(string:project)/changes/*
- **translation_review** (*boolean*) – *Włącz przeglądy*
- **source_review** (*boolean*) – *Włącz przeglądy źródeł*
- **set_language_team** (*boolean*) – *Ustaw nagłówek „Zespół Tłumaczy”*
- **enable_hooks** (*boolean*) – *Włącz hooki*
- **instructions** (*string*) – *Instrukcje tłumaczenia*
- **language_aliases** (*string*) – *Aliasy językowe*

Przykładowe dane JSON:

```
{
    "name": "Hello",
    "slug": "hello",
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
}
```

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

Nowe w wersji 4.3.

Edit a project by a PATCH request.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

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

Nowe w wersji 4.3.

Edit a project by a PUT request.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

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

Nowe w wersji 3.9.

Usuwa projekt.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

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

Returns a list of project changes. This is essentially a project scoped `GET /api/changes/` accepting same params.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

Odpowiedź Obiekt JSON

- **results** (*array*) – array of component objects; see `GET /api/changes/(int:id)/`

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

Returns information about VCS repository status. This endpoint contains only an overall summary for all repositories for the project. To get more detailed status use `GET /api/components/(string:project)/(string:component)/repository/`.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

Odpowiedź Obiekt JSON

- **needs_commit** (*boolean*) – whether there are any pending changes to commit
- **needs_merge** (*boolean*) – whether there are any upstream changes to merge
- **needs_push** (*boolean*) – whether there are any local changes to push

Przykładowe dane JSON:

```
{
    "needs_commit": true,
    "needs_merge": false,
    "needs_push": true
}
```

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

Performs given operation on the VCS repository.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

Żądanie obiektu JSON

- **operation** (*string*) – Operation to perform: one of `push`, `pull`, `commit`, `reset`, `cleanup`, `file-sync`

Odpowiedź Obiekt JSON

- **result** (*boolean*) – wynik operacji

Przykład CURL:

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

Przykład żądania 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"}
```

Przykład odpowiedzi 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/

Returns a list of translation components in the given project.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

Odpowiedź Obiekt JSON

- **results** (*array*) – array of component objects; see *GET /api/components/(string:project)/(string:component)/*

POST /api/projects/(string: *project*)/components/

Nowe w wersji 3.9.

Zmienione w wersji 4.3: The `zipfile` and `docfile` parameters are now accepted for VCS-less components, see *Pliki lokalne*.

Zmienione w wersji 4.6: The cloned repositories are now automatically shared within a project using *Wewnętrzne adresy URL Weblate*. Use `disable_autoshare` to turn off this.

Creates translation components in the given project.

Podpowiedź: Use *Wewnętrzne adresy URL Weblate* when creating multiple components from a single VCS repository.

Informacja: Most of the component creation happens in the background. Check the `task_url` attribute of created component and follow the progress there.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

Parametry formularza

- **file zipfile** – ZIP file to upload into Weblate for translations initialization
- **file docfile** – Dokument do przetłumaczenia

- **boolean disable_autoshare** – Disables automatic repository sharing via *Wewnętrzne adresy URL Weblate*.

Żądanie obiektu JSON

- **object** – Parametry komponentów, patrz *GET /api/components/(string:project)/(string:component)/*

Odpowiedź Obiekt JSON

- **result** (*object*) – Created component object; see *GET /api/components/(string:project)/(string:component)/*

JSON can not be used when uploading the files using the `zipfile` and `docfile` parameters. The data has to be uploaded as *multipart/form-data*.

Przykład żądania formularza w CURL:

```
curl \
    --form docfile=@strings.html \
    --form name=Weblate \
    --form slug=weblate \
    --form file_format=html \
    --form new_lang=add \
    -H "Authorization: Token TOKEN" \
    http://example.com/api/projects/hello/components/
```

Przykład żądania JSON w CURL:

```
curl \
    --data-binary '{
        "branch": "main",
        "file_format": "po",
        "filemask": "po/*.po",
        "name": "Weblate",
        "slug": "weblate",
        "repo": "https://github.com/WeblateOrg/hello.git",
        "template": "",
        "new_base": "po/hello.pot",
        "vcs": "git"
    }' \
    -H "Content-Type: application/json" \
    -H "Authorization: Token TOKEN" \
    http://example.com/api/projects/hello/components/
```

JSON request to create a new component from Git:

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

{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "name": "Weblate",
    "slug": "weblate",
    "repo": "https://github.com/WeblateOrg/hello.git",
    "template": "",
    "new_base": "po/hello.pot",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    "vcs": "git"
}
```

JSON request to create a new component from another one:

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

{
    "file_format": "po",
    "filemask": "po/*.po",
    "name": "Weblate",
    "slug": "weblate",
    "repo": "weblate://weblate/hello",
    "template": "",
    "new_base": "po/hello.pot",
    "vcs": "git"
}
```

Przykład odpowiedzi JSON:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "slug": "weblate",
    "project": {
        "name": "Hello",
        "slug": "hello",
        "source_language": {
            "code": "en",
            "direction": "ltr",
             "population": 159034349015,
            "name": "English",
            "url": "http://example.com/api/languages/en/",
            "web_url": "http://example.com/languages/en/"
        },
        "url": "http://example.com/api/projects/hello/",
        "web": "https://weblate.org/",
        "web_url": "http://example.com/projects/hello/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
    "new_base": "",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
}
```

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

Returns paginated statistics for all languages within a project.

Nowe w wersji 3.8.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

Odpowiedź Obiekt JSON

- **results** (*array*) – tablica obiektów statystyk tłumaczenia
- **language** (*string*) – nazwa języka
- **code** (*string*) – kod języka
- **total** (*int*) – całkowita liczba ciągów
- **translated** (*int*) – liczba przetłumaczonych ciągów tekstowych
- **translated_percent** (*float*) – procent przetłumaczonych ciągów
- **total_words** (*int*) – całkowita liczba słów
- **translated_words** (*int*) – liczba przetłumaczonych słów
- **words_percent** (*float*) – procent przetłumaczonych słów

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

Zwraca statystyki dla projektu.

Nowe w wersji 3.8.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

Odpowiedź Obiekt JSON

- **total** (*int*) – całkowita liczba ciągów
- **translated** (*int*) – liczba przetłumaczonych ciągów tekstowych
- **translated_percent** (*float*) – procent przetłumaczonych ciągów
- **total_words** (*int*) – całkowita liczba słów
- **translated_words** (*int*) – liczba przetłumaczonych słów
- **words_percent** (*float*) – procent przetłumaczonych słów

1.12.8 Komponenty

Podpowiedź: Użyj *POST /api/projects/(string:project)/components/*, aby utworzyć nowe komponenty.

GET /api/components/

Returns a list of translation components.

Zobacz także:

Component object attributes are documented at `GET /api/components/(string:project)/(string:component)/`.

GET /api/components/(string: *project*)/string: *component*/

Zwraca informacje o komponencie tłumaczenia.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Odpowiedź Obiekt JSON

- **project** (*object*) – the translation project; see `GET /api/projects/(string:project)/`
- **name** (*string*) – *Nazwa komponentu*
- **slug** (*string*) – *Końcówka adresu URL komponentu*
- **vcs** (*string*) – *System kontroli wersji*
- **repo** (*string*) – *Repozytorium kodu źródłowego*
- **git_export** (*string*) – *URL repozytorium eksportu*
- **branch** (*string*) – *Gałąź repozytorium*
- **push_branch** (*string*) – *Wypchnij gałąź*
- **filemask** (*string*) – *Maska pliku*
- **template** (*string*) – *Jednojęzyczny bazowy plik języka*
- **edit_template** (*string*) – *Edytuj plik bazowy*
- **intermediate** (*string*) – *Plik języka pośredniego*
- **new_base** (*string*) – *Szablon dla nowych tłumaczeń*
- **file_format** (*string*) – *Format pliku*
- **license** (*string*) – *Licencja tłumaczenia*
- **agreement** (*string*) – *Zgoda współpracownika*
- **new_lang** (*string*) – *Dodawanie nowego tłumaczenia*
- **language_code_style** (*string*) – *Styl kodu języka*
- **source_language** (*object*) – obiekt języka źródłowego; zobacz `GET /api/languages/(string:language)/`
- **push** (*string*) – *URL repozytorium dla push*
- **check_flags** (*string*) – *Flagi tłumaczeń*
- **priority** (*string*) – *Priorytet*
- **enforced_checks** (*string*) – *Wymuszone kontrole*
- **restricted** (*string*) – *Ograniczony dostęp*
- **repoweb** (*string*) – *Przeglądarka repozytorium*
- **report_source_bugs** (*string*) – *Adres do zgłaszania błędów łańcucha źródłowego*

- **merge_style** (*string*) – *Styl scalania*
- **commit_message** (*string*) – *Commituj, dodawaj, usuwaj, scalaj, dodawaj i scalaj żądania z wiadomościami*
- **add_message** (*string*) – *Commituj, dodawaj, usuwaj, scalaj, dodawaj i scalaj żądania z wiadomościami*
- **delete_message** (*string*) – *Commituj, dodawaj, usuwaj, scalaj, dodawaj i scalaj żądania z wiadomościami*
- **merge_message** (*string*) – *Commituj, dodawaj, usuwaj, scalaj, dodawaj i scalaj żądania z wiadomościami*
- **addon_message** (*string*) – *Commituj, dodawaj, usuwaj, scalaj, dodawaj i scalaj żądania z wiadomościami*
- **pull_message** (*string*) – *Commituj, dodawaj, usuwaj, scalaj, dodawaj i scalaj żądania z wiadomościami*
- **allow_translation_propagation** (*string*) – *Zezwól na propagację tłumaczenia*
- **enable_suggestions** (*string*) – *Włącz sugestie*
- **suggestion_voting** (*string*) – *Głosowanie na sugestie*
- **suggestion_autoaccept** (*string*) – *Automatyczne akceptowanie sugestii*
- **push_on_commit** (*string*) – *Przesyłaj przy commitowaniu*
- **commit_pending_age** (*string*) – *Wiek zmian do scommitowania*
- **auto_lock_error** (*string*) – *Błąd blokowania*
- **language_regex** (*string*) – *Filtr języka*
- **variant_regex** (*string*) – *Warianty wyrażenia regularnego*
- **repository_url** (*string*) – URL to repository status; see `GET /api/components/(string:project)/(string:component)/repository/`
- **translations_url** (*string*) – URL to translations list; see `GET /api/components/(string:project)/(string:component)/translations/`
- **lock_url** (*string*) – URL to lock status; see `GET /api/components/(string:project)/(string:component)/lock/`
- **changes_list_url** (*string*) – URL to changes list; see `GET /api/components/(string:project)/(string:component)/changes/`
- **task_url** (*string*) – URL to a background task (if any); see `GET /api/tasks/(str:uuid)/`

Przykładowe dane JSON:

```
{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "slug": "weblate",
    "project": {
        "name": "Hello",
        "slug": "hello",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
        "source_language": {
            "code": "en",
            "direction": "ltr",
             "population": 159034349015,
            "name": "English",
            "url": "http://example.com/api/languages/en/",
            "web_url": "http://example.com/languages/en/"
        },
        "url": "http://example.com/api/projects/hello/",
        "web": "https://weblate.org/",
        "web_url": "http://example.com/projects/hello/"
    },
    "source_language": {
        "code": "en",
        "direction": "ltr",
        "population": 159034349015,
        "name": "English",
        "url": "http://example.com/api/languages/en/",
        "web_url": "http://example.com/languages/en/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
    "new_base": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
}
```

PATCH /api/components/(string: *project*)/
string: *component*/

Edit a component by a PATCH request.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **source_language** (*string*) – Kod języka źródłowego projektu (opcjonalnie)

Żądanie obiektu JSON

- **name** (*string*) – nazwa komponentu
- **slug** (*string*) – końcówka adresu URL komponentu
- **repo** (*string*) – Adres URL repozytorium VCS

Przykład CURL:

```
curl \
    --data-binary '{"name": "new name"}' \
    -H "Content-Type: application/json" \
    -H "Authorization: Token TOKEN" \
    PATCH http://example.com/api/projects/hello/components/
```

Przykład żądania 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
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
{
    "name": "new name"
}
```

Przykład odpowiedzi JSON:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "new name",
    "slug": "weblate",
    "project": {
        "name": "Hello",
        "slug": "hello",
        "source_language": {
            "code": "en",
            "direction": "ltr",
            "population": 159034349015,
            "name": "English",
            "url": "http://example.com/api/languages/en/",
            "web_url": "http://example.com/languages/en/"
        },
        "url": "http://example.com/api/projects/hello/",
        "web": "https://weblate.org/",
        "web_url": "http://example.com/projects/hello/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
    "new_base": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
}
```

PUT /api/components/(string: *project*)/string: *component*/

Edit a component by a PUT request.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Żądanie obiektu JSON

- **branch** (*string*) – Gałąź repozytorium VCS
- **file_format** (*string*) – format plików z tłumaczeniami

- **filemask** (*string*) – mask of translation files in the repository
- **name** (*string*) – nazwa komponentu
- **slug** (*string*) – końcówka adresu URL komponentu
- **repo** (*string*) – Adres URL repozytorium VCS
- **template** (*string*) – plik podstawowy do tłumaczeń jednojęzycznych
- **new_base** (*string*) – base file for adding new translations
- **vcs** (*string*) – system kontroli wersji

DELETE /api/components/(string: *project*)/string: *component*/

Nowe w wersji 3.9.

Usuwa komponent.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

GET /api/components/(string: *project*)/string: *component*/changes/

Returns a list of component changes. This is essentially a component scoped *GET /api/changes/* accepting same params.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Odpowiedź Obiekt JSON

- **results** (*array*) – array of component objects; see *GET /api/changes/(int:id)/*

GET /api/components/(string: *project*)/string: *component*/file/

Nowe w wersji 4.9.

Downloads all available translations associated with the component as an archive file using the requested format.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Parametry zapytania

- **format** (*string*) – The archive format to use; If not specified, defaults to `zip`; Supported formats: `zip`
- **q** (*string*) – Filter downloaded strings, see search.

GET /api/components/(string: *project*)/string: *component*/screenshots/

Returns a list of component screenshots.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Odpowiedź Obiekt JSON

- **results** (*array*) – array of component screenshots; see *GET /api/screenshots/(int:id)/*

GET /api/components/(string: *project*)/string: *component*/lock/

Zwraca stan blokady komponentu.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Odpowiedź Obiekt JSON

- **locked** (*boolean*) – whether component is locked for updates

Przykładowe dane JSON:

```
{
    "locked": false
}
```

POST /api/components/(string: *project*)/string: *component*/lock/

Ustawia stan blokady komponentów.

Response is same as *GET /api/components/(string:project)/(string:component)/lock/*.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Żądanie obiektu JSON

- **lock** – Boolean whether to lock or not.

Przykład CURL:

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

Przykład żądania 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}
```

Przykład odpowiedzi 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
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"locked":true}
```

GET /api/components/(string: *project*)/(string: *component*)/repository/

Returns information about VCS repository status.

The response is same as for *GET /api/projects/(string:project)/repository/*.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Odpowiedź Obiekt JSON

- **needs_commit** (*boolean*) – whether there are any pending changes to commit
- **needs_merge** (*boolean*) – whether there are any upstream changes to merge
- **needs_push** (*boolean*) – whether there are any local changes to push
- **remote_commit** (*string*) – Informacje o zdalnym commicie
- **status** (*string*) – VCS repository status as reported by VCS
- **merge_failure** – Text describing merge failure or null if there is none

POST /api/components/(string: *project*)/(string: *component*)/repository/

Performs the given operation on a VCS repository.

Zobacz *POST /api/projects/(string:project)/repository/* w celu uzyskania dokumentacji.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Żądanie obiektu JSON

- **operation** (*string*) – Operation to perform: one of `push`, `pull`, `commit`, `reset`, `cleanup`

Odpowiedź Obiekt JSON

- **result** (*boolean*) – wynik operacji

Przykład CURL:

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

Przykład żądania 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
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

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

Przykład odpowiedzi 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/

Downloads base file for monolingual translations.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

GET /api/components/(string: *project*)/ string: *component*/new_template/

Downloads template file for new translations.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

GET /api/components/(string: *project*)/ string: *component*/translations/

Returns a list of translation objects in the given component.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Odpowiedź Obiekt JSON

- **results** (*array*) – array of translation objects; see *GET /api/translations/(string:project)/(string:component)/(string:language)/*

POST /api/components/(string: *project*)/ string: *component*/translations/

Creates new translation in the given component.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Żądanie obiektu JSON

- **language_code** (*string*) – translation language code; see *GET /api/languages/(string:language)/*

Odpowiedź Obiekt JSON

- **result** (*object*) – tworzony nowy obiekt tłumaczeniowy

Przykład CURL:

```
curl \
    -d language_code=cs \
    -H "Authorization: Token TOKEN" \
    http://example.com/api/projects/hello/components/
```

Przykład żądania 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"}
```

Przykład odpowiedzi JSON:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
    "failing_checks": 0,
    "failing_checks_percent": 0,
    "failing_checks_words": 0,
    "filename": "po/cs.po",
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "fuzzy_words": 0,
    "have_comment": 0,
    "have_suggestion": 0,
    "is_template": false,
    "is_source": false,
    "language": {
        "code": "cs",
        "direction": "ltr",
        "population": 1303174280
        "name": "Czech",
        "url": "http://example.com/api/languages/cs/",
        "web_url": "http://example.com/languages/cs/"
    },
    "language_code": "cs",
    "id": 125,
    "last_author": null,
    "last_change": null,
    "share_url": "http://example.com/engage/hello/cs/",
    "total": 4,
    "total_words": 15,
    "translate_url": "http://example.com/translate/hello/weblate/cs/",
    "translated": 0,
    "translated_percent": 0.0,
    "translated_words": 0,
    "url": "http://example.com/api/translations/hello/weblate/cs/",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    "web_url": "http://example.com/projects/hello/weblate/cs/"
}
```

GET /api/components/(string: *project*)/string: *component*/statistics/

Returns paginated statistics for all translations within component.

Nowe w wersji 2.7.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Odpowiedź Obiekt JSON

- **results** (*array*) – array of translation statistics objects; see *GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/*

GET /api/components/(string: *project*)/string: *component*/links/

Zwraca projekty powiązane z danym komponentem.

Nowe w wersji 4.5.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Odpowiedź Obiekt JSON

- **projects** (*array*) – associated projects; see *GET /api/projects/(string:project)/*

POST /api/components/(string: *project*)/string: *component*/links/

Powiąż projekt z komponentem.

Nowe w wersji 4.5.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu

Parametry formularza

- **string project_slug** – Końcówka adresu URL projektu

DELETE /api/components/(string: *project*)/string: *component*/links/string: *project_slug*/

Usuń powiązanie projektu z komponentem.

Nowe w wersji 4.5.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **project_slug** (*string*) – Końcówka adresu URL projektu do usunięcia

1.12.9 Tłumaczenia

GET /api/translations/

Zwraca listę tłumaczeń.

Zobacz także:

Translation object attributes are documented at *GET /api/translations/(string:project)/(string:component)/(string:language)/*.

GET /api/translations/(string: *project***)/string:** *component***/string:** *language***/**

Zwraca informacje o tłumaczeniu.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

Odpowiedź Obiekt JSON

- **component** (*object*) – obiekt komponentu; zobacz *GET /api/components/(string:project)/(string:component)/*
- **failing_checks** (*int*) – Liczba ciągów, które nie przeszły kontroli
- **failing_checks_percent** (*float*) – procent ciągów, które nie przeszły kontroli
- **failing_checks_words** (*int*) – liczba słów z nieudanymi kontrolami
- **filename** (*string*) – nazwa pliku tłumaczenia
- **fuzzy** (*int*) – liczba rozmytych (zaznaczonych do edycji) ciągów
- **fuzzy_percent** (*float*) – percentage of fuzzy (marked for edit) strings
- **fuzzy_words** (*int*) – number of words in fuzzy (marked for edit) strings
- **have_comment** (*int*) – liczba ciągów z komentarzem
- **have_suggestion** (*int*) – liczba ciągów z sugestią
- **is_template** (*boolean*) – czy tłumaczenie ma jednojęzyczną podstawę
- **language** (*object*) – obiekt języka źródłowego; zobacz *GET /api/languages/(string:language)/*
- **language_code** (*string*) – language code used in the repository; this can be different from language code in the language object
- **last_author** (*string*) – nazwisko ostatniego autora
- **last_change** (*timestamp*) – znacznik czasu ostatniej zmiany
- **revision** (*string*) – revision hash for the file
- **share_url** (*string*) – URL for sharing leading to engagement page
- **total** (*int*) – całkowita liczba ciągów
- **total_words** (*int*) – całkowita liczba słów
- **translate_url** (*string*) – Adres URL tłumaczenia
- **translated** (*int*) – liczba przetłumaczonych ciągów tekstowych
- **translated_percent** (*float*) – procent przetłumaczonych ciągów
- **translated_words** (*int*) – liczba przetłumaczonych słów

- **repository_url** (*string*) – URL to repository status; see *GET /api/translations/(string:project)/(string:component)/(string:language)/repository/*
- **file_url** (*string*) – URL to file object; see *GET /api/translations/(string:project)/(string:component)/(string:language)/file/*
- **changes_list_url** (*string*) – URL to changes list; see *GET /api/translations/(string:project)/(string:component)/(string:language)/changes/*
- **units_list_url** (*string*) – URL to strings list; see *GET /api/translations/(string:project)/(string:component)/(string:language)/units/*

Przykładowe dane JSON:

```
{
    "component": {
        "branch": "main",
        "file_format": "po",
        "filemask": "po/*.po",
        "git_export": "",
        "license": "",
        "license_url": "",
        "name": "Weblate",
        "new_base": "",
        "project": {
            "name": "Hello",
            "slug": "hello",
            "source_language": {
                "code": "en",
                "direction": "ltr",
                "population": 159034349015,
                "name": "English",
                "url": "http://example.com/api/languages/en/",
                "web_url": "http://example.com/languages/en/"
            },
            "url": "http://example.com/api/projects/hello/",
            "web": "https://weblate.org/",
            "web_url": "http://example.com/projects/hello/"
        },
        "repo": "file:///home/nijel/work/weblate-hello",
        "slug": "weblate",
        "template": "",
        "url": "http://example.com/api/components/hello/weblate/",
        "vcs": "git",
        "web_url": "http://example.com/projects/hello/weblate/"
    },
    "failing_checks": 3,
    "failing_checks_percent": 75.0,
    "failing_checks_words": 11,
    "filename": "po/cs.po",
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "fuzzy_words": 0,
    "have_comment": 0,
    "have_suggestion": 0,
    "is_template": false,
    "language": {
        "code": "cs",
        "direction": "ltr",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)


```
        "population": 1303174280
        "name": "Czech",
        "url": "http://example.com/api/languages/cs/",
        "web_url": "http://example.com/languages/cs/"
    },
    "language_code": "cs",
    "last_author": "Weblate Admin",
    "last_change": "2016-03-07T10:20:05.499",
    "revision": "7ddfafe6daaf57fc8654cc852ea6be212b015792",
    "share_url": "http://example.com/engage/hello/cs/",
    "total": 4,
    "total_words": 15,
    "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*)/

Nowe w wersji 3.9.

Usuwa tłumaczenie.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

GET /api/translations/(string: *project*)/(string: *component*)/(string: *language*)/changes/

Returns a list of translation changes. This is essentially a translations-scoped *GET /api/changes/* accepting the same parameters.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

Odpowiedź Obiekt JSON

- **results** (*array*) – array of component objects; see *GET /api/changes/(int:id)/*

GET /api/translations/(string: *project*)/(string: *component*)/(string: *language*)/units/

Returns a list of translation units.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia
- **q** (*string*) – Search query string *Wyszukiwanie* (optional)

Odpowiedź Obiekt JSON

- **results** (*array*) – array of component objects; see *GET /api/units/(int:id)/*

POST /api/translations/(string: *project*)/string: *component*/string: *language*/units/

Dodaj nową jednostkę.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

Żądanie obiektu JSON

- **key** (*string*) – Nazwa jednostki tłumaczeniowej (używana jako klucz lub kontekst)
- **value** (*array*) – Source strings (use single string if not creating plural)

Odpowiedź Obiekt JSON

- **unit** (*object*) – newly created unit; see *GET /api/units/(int:id)/*

Zobacz także:

Zarządzaj ciągami, adding-new-strings

POST /api/translations/(string: *project*)/string: *component*/string: *language*/autotranslate/

Wyzwalanie automatycznego tłumaczenia.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

Żądanie obiektu JSON

- **mode** (*string*) – Tryb automatycznego tłumaczenia
- **filter_type** (*string*) – Typ filtru automatycznego tłumaczenia
- **auto_source** (*string*) – Źródło automatycznego tłumaczenia - `mt` lub `others`
- **component** (*string*) – Włącz wkład do wspólnej pamięci tłumaczeniowej dla projektu, aby uzyskać dostęp do dodatkowych komponentów.
- **engines** (*array*) – Silniki tłumaczenia maszynowego
- **threshold** (*string*) – Ocena wyniku

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

Download current translation file as it is stored in the VCS (without the `format` parameter) or converted to another format (see *Pobieranie tłumaczeń*).

Informacja: This API endpoint uses different logic for output than rest of API as it operates on whole file rather than on data. Set of accepted `format` parameter differs and without such parameter you get translation file as stored in VCS.

Parametry zapytania

- **format** – File format to use; if not specified no format conversion happens; supported file formats: `po`, `mo`, `xliff`, `xliff11`, `tbx`, `csv`, `xlsx`, `json`, `aresource`, `strings`

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

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

Prześlij nowy plik z tłumaczeniami.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

Parametry formularza

- **string conflict** – How to deal with conflicts (`ignore`, `replace-translated` or `replace-approved`)
- **file file** – Przesłany plik
- **string email** – Adres e-mail autora
- **string author** – Imię i nazwisko autora
- **string method** – Upload method (`translate`, `approve`, `suggest`, `fuzzy`, `replace`, `source`, `add`), see *Metody importu*
- **string fuzzy** – Fuzzy (marked for edit) strings processing (*empty*, `process`, `approve`)

Przykład 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/

Returns information about VCS repository status.

The response is same as for *GET /api/components/(string:project)/(string:component)/repository/*.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

POST /api/translations/(string: *project*)/(string: *component*)/(string: *language*)/repository/

Performs given operation on the VCS repository.

Zobacz *POST /api/projects/(string:project)/repository/* w celu uzyskania dokumentacji.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu

- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

Żądanie obiektu JSON

- **operation** (*string*) – Operation to perform: one of `push`, `pull`, `commit`, `reset`, `cleanup`

Odpowiedź Obiekt JSON

- **result** (*boolean*) – wynik operacji

GET /api/translations/(string: *project*)/string: *component*/string: *language*/statistics/

Zwraca szczegółowe dane statystyczne dotyczące tłumaczeń.

Nowe w wersji 2.7.

Parametry

- **project** (*string*) – Końcówka adresu URL projektu
- **component** (*string*) – Końcówka adresu URL komponentu
- **language** (*string*) – Kod języka tłumaczenia

Odpowiedź Obiekt JSON

- **code** (*string*) – kod języka
- **failing** (*int*) – liczba nieudanych kontroli
- **failing_percent** (*float*) – odsetek nieudanych kontroli
- **fuzzy** (*int*) – liczba rozmytych (zaznaczonych do edycji) ciągów
- **fuzzy_percent** (*float*) – percentage of fuzzy (marked for edit) strings
- **total_words** (*int*) – całkowita liczba słów
- **translated_words** (*int*) – liczba przetłumaczonych słów
- **last_author** (*string*) – nazwisko ostatniego autora
- **last_change** (*timestamp*) – data ostatniej zmiany
- **name** (*string*) – nazwa języka
- **total** (*int*) – całkowita liczba ciągów
- **translated** (*int*) – liczba przetłumaczonych ciągów tekstowych
- **translated_percent** (*float*) – procent przetłumaczonych ciągów
- **url** (*string*) – URL to access the translation (engagement URL)
- **url_translate** (*string*) – URL to access the translation (real translation URL)

1.12.10 Jednostki

A *unit* is a single piece of a translation which pairs a source string with a corresponding translated string and also contains some related metadata. The term is derived from the Translate Toolkit and XLIFF.

Nowe w wersji 2.10.

GET /api/units/

Returns list of translation units.

Zobacz także:

Unit object attributes are documented at `GET /api/units/(int:id)/`.

GET /api/units/(int: *id*)/

Zmienione w wersji 4.3: The `target` and `source` are now arrays to properly handle plural strings.

Returns information about translation unit.

Parametry

- **id** (*int*) – Identyfikator jednostki

Odpowiedź Obiekt JSON

- **translation** (*string*) – URL of a related translation object
- **source** (*array*) – ciąg źródłowy
- **previous_source** (*string*) – previous source string used for fuzzy matching
- **target** (*array*) – ciąg docelowy
- **id_hash** (*string*) – niepowtarzalny identyfikator jednostki
- **content_hash** (*string*) – unique identifier of the source string
- **location** (*string*) – location of the unit in source code
- **context** (*string*) – ontekst tłumaczenia
- **note** (*string*) – notatka o jednostce tłumaczeniowej
- **flags** (*string*) – flagi jednostek tłumaczeniowych
- **state** (*int*) – unit state, 0 - untranslated, 10 - needs editing, 20 - translated, 30 - approved, 100 - read only
- **fuzzy** (*boolean*) – czy jednostka jest rozmyta lub oznaczona do przeglądu
- **translated** (*boolean*) – czy dana jednostka jest tłumaczona
- **approved** (*boolean*) – czy tłumaczenie jest zatwierdzone
- **position** (*int*) – pozycja jednostki w pliku tłumaczenia
- **has_suggestion** (*boolean*) – czy jednostka ma sugestie
- **has_comment** (*boolean*) – czy jednostka ma komentarze
- **has_failing_check** (*boolean*) – czy jednostka nie przeszła pomyślnie testów
- **num_words** (*int*) – liczba słów źródłowych
- **priority** (*int*) – priorytet tłumaczenia; 100 to wartość domyślna
- **id** (*int*) – identyfikator jednostki
- **explanation** (*string*) – String explanation, available on source units, see *Dodatkowe informacje o ciągach źródłowych*
- **extra_flags** (*string*) – Additional string flags, available on source units, see *Dostosowywanie zachowania za pomocą flag*
- **web_url** (*string*) – URL where the unit can be edited
- **source_unit** (*string*) – Source unit link; see `GET /api/units/(int:id)/`
- **pending** (*boolean*) – czy jednostka oczekuje na zapis
- **timestamp** (*timestamp*) – wiek ciągu

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

Nowe w wersji 4.3.

Wykonuje częściową aktualizację jednostki tłumaczenia.

Parametry

- **id** (*int*) – Identyfikator jednostki

Żądanie obiektu JSON

- **state** (*int*) – unit state, 0 - untranslated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see *Dedykowani recenzenci*)
- **target** (*array*) – ciąg docelowy
- **explanation** (*string*) – String explanation, available on source units, see *Dodatkowe informacje o ciągach źródłowych*
- **extra_flags** (*string*) – Additional string flags, available on source units, see *Dostosowywanie zachowania za pomocą flag*

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

Nowe w wersji 4.3.

Wykonuje pełną aktualizację jednostki tłumaczeniowej.

Parametry

- **id** (*int*) – Identyfikator jednostki

Żądanie obiektu JSON

- **state** (*int*) – unit state, 0 - untranslated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see *Dedykowani recenzenci*)
- **target** (*array*) – ciąg docelowy
- **explanation** (*string*) – String explanation, available on source units, see *Dodatkowe informacje o ciągach źródłowych*
- **extra_flags** (*string*) – Additional string flags, available on source units, see *Dostosowywanie zachowania za pomocą flag*

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

Nowe w wersji 4.3.

Usuwa jednostkę tłumaczeniową.

Parametry

- **id** (*int*) – Identyfikator jednostki

1.12.11 Zmiany

Nowe w wersji 2.10.

GET /api/changes/

Zmienione w wersji 4.1: Filtering of changes was introduced in the 4.1 release.

Returns a list of translation changes.

Zobacz także:

Change object attributes are documented at `GET /api/changes/(int:id)/`.

Parametry zapytania

- **user** (*string*) – Nazwa użytkownika do filtrów
- **action** (*int*) – Action to filter, can be used several times
- **timestamp_after** (*timestamp*) – ISO 8601 formatted timestamp to list changes after

- **timestamp_before** (*timestamp*) – ISO 8601 formatted timestamp to list changes before

GET /api/changes/(**int:** *id*)**/**

Zwraca informacje o zmianie tłumaczenia.

Parametry

- **id** (*int*) – Zmiana identyfikatora

Odpowiedź Obiekt JSON

- **unit** (*string*) – URL of a related unit object
- **translation** (*string*) – URL of a related translation object
- **component** (*string*) – URL of a related component object
- **user** (*string*) – URL of a related user object
- **author** (*string*) – URL of a related author object
- **timestamp** (*timestamp*) – znacznik czasu zdarzenia
- **action** (*int*) – numeryczna identyfikacja działania
- **action_name** (*string*) – opis tekstowy działania
- **target** (*string*) – event changed text or detail
- **id** (*int*) – zmiana identyfikatora

1.12.12 Zrzuty ekranu

Nowe w wersji 2.14.

GET /api/screenshots/

Returns a list of screenshot string information.

Zobacz także:

Screenshot object attributes are documented at *GET /api/screenshots/(int:id)/*.

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

Zwraca informacje o zrzutach ekranu.

Parametry

- **id** (*int*) – Identyfikator zrzutu ekranu

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa zrzutu ekranu
- **component** (*string*) – URL of a related component object
- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

GET /api/screenshots/(**int:** *id*)**/file/**

Pobierz zrzut ekranu.

Parametry

- **id** (*int*) – Identyfikator zrzutu ekranu

POST /api/screenshots/(int: *id*)/file/

Zastąp obraz zrzutu ekranu.

Parametry

- **id** (*int*) – Identyfikator zrzutu ekranu

Parametry formularza

- **file image** – Przesłany plik

Przykład 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/

Powiąż ciąg źródłowy ze zrzutem ekranu.

Parametry

- **id** (*int*) – Identyfikator zrzutu ekranu

Parametry formularza

- **string unit_id** – Identyfikator jednostki

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa zrzutu ekranu
- **translation** (*string*) – URL of a related translation object
- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

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

Usuń powiązanie zrzutu ekranu z ciągami źródłowymi.

Parametry

- **id** (*int*) – Identyfikator zrzutu ekranu
- **unit_id** – ID jednostki ciągu źródłowego

POST /api/screenshots/

Tworzy nowy zrzut ekranu.

Parametry formularza

- **file image** – Przesłany plik
- **string name** – Nazwa zrzutu ekranu
- **string project_slug** – Końcówka adresu URL projektu
- **string component_slug** – Końcówka adresu URL komponentu
- **string language_code** – Kod języka

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa zrzutu ekranu
- **component** (*string*) – URL of a related component object

- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

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

Edytuj częściowe informacje o zrzucie ekranu.

Parametry

- **id** (*int*) – Identyfikator zrzutu ekranu

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa zrzutu ekranu
- **component** (*string*) – URL of a related component object
- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

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

Edytuj pełne informacje o zrzucie ekranu.

Parametry

- **id** (*int*) – Identyfikator zrzutu ekranu

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa zrzutu ekranu
- **component** (*string*) – URL of a related component object
- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

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

Usuń zrzut ekranu.

Parametry

- **id** (*int*) – Identyfikator zrzutu ekranu

1.12.13 Dodatki

Nowe w wersji 4.4.1.

GET /api/addons/

Zwraca listę dodatków.

Zobacz także:

Add-on object attributes are documented at *GET /api/addons/(int:id)/*.

GET /api/addons/(int: *id*)/

Zwraca informacje o dodatku.

Parametry

- **id** (*int*) – ID dodatku

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa dodatku
- **component** (*string*) – URL of a related component object
- **configuration** (*object*) – Opcjonalna konfiguracja dodatku

Zobacz także:

Dodatki

POST /api/components/(string: *project*)/(string: *component*)/addons/

Tworzy nowy dodatek.

Parametry

- **project_slug** (*string*) – Końcówka adresu URL projektu
- **component_slug** (*string*) – Końcówka adresu URL komponentu

Żądanie obiektu JSON

- **name** (*string*) – nazwa dodatku
- **configuration** (*object*) – Opcjonalna konfiguracja dodatku

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

Edytuj częściowe informacje o dodatku.

Parametry

- **id** (*int*) – ID dodatku

Odpowiedź Obiekt JSON

- **configuration** (*object*) – Opcjonalna konfiguracja dodatku

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

Edytuj pełne informacje o dodatku.

Parametry

- **id** (*int*) – ID dodatku

Odpowiedź Obiekt JSON

- **configuration** (*object*) – Opcjonalna konfiguracja dodatku

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

Usuń dodatek.

Parametry

- **id** (*int*) – ID dodatku

1.12.14 Listy komponentów

Nowe w wersji 4.0.

GET /api/component-lists/

Returns a list of component lists.

Zobacz także:

Component list object attributes are documented at *GET /api/component-lists/(str:slug)/*.

GET /api/component-lists/(str: *slug*)/

Zwraca informacje o liście komponentów.

Parametry

- **slug** (*string*) – Końcówka adresu URL listy komponentów

Odpowiedź Obiekt JSON

- **name** (*string*) – nazwa listy komponentów
- **slug** (*string*) – końcówka adresu URL listy komponentów
- **show_dashboard** (*boolean*) – whether to show it on a dashboard
- **components** (*array*) – link to associated components; see *GET /api/components/(string:project)/(string:component)/*
- **auto_assign** (*array*) – zasady automatycznego przydzielania

PUT /api/component-lists/(str: *slug*)/

Zmienia parametry listy komponentów.

Parametry

- **slug** (*string*) – Końcówka adresu URL listy komponentów

Żądanie obiektu JSON

- **name** (*string*) – nazwa listy komponentów
- **slug** (*string*) – końcówka adresu URL listy komponentów
- **show_dashboard** (*boolean*) – whether to show it on a dashboard

PATCH /api/component-lists/(str: *slug*)/

Zmienia parametry listy komponentów.

Parametry

- **slug** (*string*) – Końcówka adresu URL listy komponentów

Żądanie obiektu JSON

- **name** (*string*) – nazwa listy komponentów
- **slug** (*string*) – końcówka adresu URL listy komponentów
- **show_dashboard** (*boolean*) – whether to show it on a dashboard

DELETE /api/component-lists/(str: *slug*)/

Usuwa listę komponentów.

Parametry

- **slug** (*string*) – Końcówka adresu URL listy komponentów

POST /api/component-lists/(str: *slug*)/components/

Associate component with a component list.

Parametry

- **slug** (*string*) – Końcówka adresu URL listy komponentów

Parametry formularza

- **string component_id** – Identyfikator komponentu

DELETE /api/component-lists/(str: *slug*)/components/(str: *component_slug*)

Disassociate a component from the component list.

Parametry

- **slug** (*string*) – Końcówka adresu URL listy komponentów
- **component_slug** (*string*) – Końcówka adresu URL komponentu

1.12.15 Słownik

Zmienione w wersji 4.5: Glossaries are now stored as regular components, translations and strings, please use respective API instead.

1.12.16 Zadania

Nowe w wersji 4.4.

GET /api/tasks/

Lista zadań jest obecnie niedostępna.

GET /api/tasks/(str: *uuid*)/

Zwraca informacje o zadaniu

Parametry

- **uuid** (*string*) – Identyfikator UUID zadania

Odpowiedź Obiekt JSON

- **completed** (*boolean*) – Określa, czy zadanie zostało ukończone
- **progress** (*int*) – Task progress in percent
- **result** (*object*) – Task result or progress details
- **log** (*string*) – Dziennik zadań

1.12.17 Metrics

GET /api/metrics/

Returns server metrics.

Odpowiedź Obiekt JSON

- **units** (*int*) – Liczba jednostek
- **units_translated** (*int*) – Liczba przetłumaczonych jednostek
- **users** (*int*) – Liczba użytkowników
- **changes** (*int*) – Liczba zmian
- **projects** (*int*) – Liczba projektów
- **components** (*int*) – Liczba komponentów
- **translations** (*int*) – Liczba tłumaczeń
- **languages** (*int*) – Liczba używanych języków
- **checks** (*int*) – Liczba uruchomionych kontroli jakości
- **configuration_errors** (*int*) – Liczba błędów konfiguracji
- **suggestions** (*int*) – Liczba oczekujących sugestii
- **celery_queues** (*object*) – Długości kolejek Celery, patrz *Zadania w tle korzystające z Celery*
- **name** (*string*) – Skonfigurowana nazwa serwera

1.12.18 Hooki powiadomień

Notification hooks allow external applications to notify Weblate that the VCS repository has been updated.

You can use repository endpoints for projects, components and translations to update individual repositories; see `POST /api/projects/(string:project)/repository/` for documentation.

GET /hooks/update/(string: *project*)/(string: *component*)/

Niezalecane od wersji 2.6: Please use `POST /api/components/(string:project)/(string:component)/repository/` instead which works properly with authentication for ACL limited projects.

Triggers update of a component (pulling from VCS and scanning for translation changes).

GET /hooks/update/(string: *project*)/

Niezalecane od wersji 2.6: Please use `POST /api/projects/(string:project)/repository/` instead which works properly with authentication for ACL limited projects.

Triggers update of all components in a project (pulling from VCS and scanning for translation changes).

POST /hooks/github/

Special hook for handling GitHub notifications and automatically updating matching components.

Informacja: GitHub includes direct support for notifying Weblate: enable Weblate service hook in repository settings and set the URL to the URL of your Weblate installation.

Zobacz także:

Automatyczne otrzymywanie zmian z GitHub
For instruction on setting up GitHub integration

https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks
Ogólne informacje na temat Webhooków GitHub

ENABLE_HOOKS
For enabling hooks for whole Weblate

POST /hooks/gitlab/

Special hook for handling GitLab notifications and automatically updating matching components.

Zobacz także:

Automatyczne otrzymywanie zmian z GitHub
For instruction on setting up GitLab integration

https://docs.gitlab.com/ee/user/project/integrations/webhooks.html
Ogólne informacje na temat Webhooków GitLab

ENABLE_HOOKS
For enabling hooks for whole Weblate

POST /hooks/bitbucket/

Special hook for handling Bitbucket notifications and automatically updating matching components.

Zobacz także:

Automatyczne otrzymywanie zmian z Bitbucket
For instruction on setting up Bitbucket integration

https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/
Ogólne informacje na temat WebhookówBitbucket

ENABLE_HOOKS
For enabling hooks for whole Weblate

POST /hooks/pagure/

Nowe w wersji 3.3.

Special hook for handling Pagure notifications and automatically updating matching components.

Zobacz także:

Automatyczne odbieranie zmian Pagure
For instruction on setting up Pagure integration

https://docs.pagure.org/pagure/usage/using_webhooks.html
Ogólne informacje na temat Webhooków

ENABLE_HOOKS
For enabling hooks for whole Weblate

POST /hooks/azure/

Nowe w wersji 3.8.

Special hook for handling Azure DevOps notifications and automatically updating matching components.

Informacja: Please make sure that *Resource details to send* is set to *All*, otherwise Weblate will not be able to match your Azure repository.

Zobacz także:

Automatically receiving changes from Azure Repos
For instruction on setting up Azure integration

https://docs.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops
Ogólne informacje na temat Azure DevOps Web Hooks

ENABLE_HOOKS
For enabling hooks for whole Weblate

POST /hooks/gitea/

Nowe w wersji 3.9.

Special hook for handling Gitea Webhook notifications and automatically updating matching components.

Zobacz także:

Automatically receiving changes from Gitea Repos
For instruction on setting up Gitea integration

https://docs.gitea.io/en-us/webhooks/
Ogólne informacje na temat Webhooków Gitea

ENABLE_HOOKS
For enabling hooks for whole Weblate

POST /hooks/gitee/

Nowe w wersji 3.9.

Special hook for handling Gitee Webhook notifications and automatically updating matching components.

Zobacz także:

Automatically receiving changes from Gitee Repos
For instruction on setting up Gitee integration

https://gitee.com/help/categories/40
Ogólne informacje na temat Webhooków Gitee

ENABLE_HOOKS
For enabling hooks for whole Weblate

1.12.19 Eksport

Weblate provides various exports to allow you to further process the data.

GET /exports/stats/(string: *project*)/
string: *component*/

Parametry zapytania

- **format** (*string*) – Output format: either `json` or `csv`

Niezalecane od wersji 2.6: Please use *GET /api/components/(string:project)/(string:component)/statistics/* and *GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/* instead; it allows access to ACL controlled projects as well.

Retrieves statistics for given component in given format.

Przykładowe żądanie:

```
GET /exports/stats/weblate/main/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Przykładowa odpowiedź:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

[
    {
        "code": "cs",
        "failing": 0,
        "failing_percent": 0.0,
        "fuzzy": 0,
        "fuzzy_percent": 0.0,
        "last_author": "Michal Čihař",
        "last_change": "2012-03-28T15:07:38+00:00",
        "name": "Czech",
        "total": 436,
        "total_words": 15271,
        "translated": 436,
        "translated_percent": 100.0,
        "translated_words": 3201,
        "url": "http://hosted.weblate.org/engage/weblate/cs/",
        "url_translate": "http://hosted.weblate.org/projects/weblate/main/cs/"
    },
    {
        "code": "nl",
        "failing": 21,
        "failing_percent": 4.8,
        "fuzzy": 11,
        "fuzzy_percent": 2.5,
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)


```
        "last_author": null,
        "last_change": null,
        "name": "Dutch",
        "total": 436,
        "total_words": 15271,
        "translated": 319,
        "translated_percent": 73.2,
        "translated_words": 3201,
        "url": "http://hosted.weblate.org/engage/weblate/nl/",
        "url_translate": "http://hosted.weblate.org/projects/weblate/main/nl/"
    },
    {
        "code": "el",
        "failing": 11,
        "failing_percent": 2.5,
        "fuzzy": 21,
        "fuzzy_percent": 4.8,
        "last_author": null,
        "last_change": null,
        "name": "Greek",
        "total": 436,
        "total_words": 15271,
        "translated": 312,
        "translated_percent": 71.6,
        "translated_words": 3201,
        "url": "http://hosted.weblate.org/engage/weblate/el/",
        "url_translate": "http://hosted.weblate.org/projects/weblate/main/el/"
    }
]
```


1.12.20 Kanały RSS

Changes in translations are exported in RSS feeds.

GET /exports/rss/(string: *project*)/(string: *component*)/(string: *language*)/

Retrieves RSS feed with recent changes for a translation.

GET /exports/rss/(string: *project*)/(string: *component*)/

Retrieves RSS feed with recent changes for a component.

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

Retrieves RSS feed with recent changes for a project.

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

Retrieves RSS feed with recent changes for a language.

GET /exports/rss/

Retrieves RSS feed with recent changes for Weblate instance.

Zobacz także:

RSS w Wikipedii

1.13 Klient Weblate

Nowe w wersji 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 Instalacja

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 Użycie platformy Docker

The Weblate Client is also available as a Docker image.

The image is published on Docker Hub: https://hub.docker.com/r/weblate/wlc

Instalowanie:

```
docker pull weblate/wlc
```

The Docker container uses Weblate's default settings and connects to the API deployed in localhost. The API URL and API_KEY can be configured through the arguments accepted by Weblate.

The command to launch the container uses the following syntax:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Przykład:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

You might want to pass your *Plik konfiguracyjny* to the Docker container, the easiest approach is to add your current directory as `/home/weblate` volume:

```
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

1.13.3 Pierwsze kroki

The wlc configuration is stored in `~/.config/weblate` (see *Plik konfiguracyjny* for other locations), please create it to match your environment:

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

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

You can then invoke commands on the default server:

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

Zobacz także:

Plik konfiguracyjny

1.13.4 Streszczenie

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

Commands actually indicate which operation should be performed.

1.13.5 Opis

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

Argumenty

The program accepts the following arguments which define output format or which Weblate instance to use. These must be entered before any command.

--format `{csv,json,text,html}`

Określ format danych wyjściowych.

--url `URL`

Specify the API URL. Overrides any value found in the configuration file, see *Plik konfiguracyjny*. 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 *Plik konfiguracyjny*. You can find your key in your profile on Weblate.

--config `PATH`

Overrides the configuration file path, see *Plik konfiguracyjny*.

--config-section `SECTION`

Overrides configuration file section in use, see *Plik konfiguracyjny*.

Polecenia

Te polecenia są dostępne:

version

Wydrukuj bieżącą wersję.

list-languages

Lista języków używanych w Weblate.

list-projects

Lista projektów w Weblate.

list-components

Lista komponentów w Weblate.

list-translations

Lista tłumaczeń w Weblate.

show

Shows Weblate object (translation, component or project).

ls

Lists Weblate object (translation, component or project).

commit

Commits changes made in a Weblate object (translation, component or project).

pull

Pulls remote repository changes into Weblate object (translation, component or project).

push

Pushes Weblate object changes into remote repository (translation, component or project).

reset

Nowe w wersji 0.7: Obsługiwane od wersji wlc 0.7.

Resets changes in Weblate object to match remote repository (translation, component or project).

cleanup

Nowe w wersji 0.9: Obsługiwane od wersji wlc 0.9.

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

repo

Displays repository status for a given Weblate object (translation, component or project).

statistics

Displays detailed statistics for a given Weblate object (translation, component or project).

lock-status

Nowe w wersji 0.5: Obsługiwane od wersji wlc 0.5.

Wyświetla stan blokady.

lock

Nowe w wersji 0.5: Obsługiwane od wersji wlc 0.5.

Locks component from further translation in Weblate.

unlock

Nowe w wersji 0.5: Obsługiwane od wersji wlc 0.5.

Unlocks translation of Weblate component.

changes

Nowe w wersji 0.7: Supported since wlc 0.7 and Weblate 2.10.

Displays changes for a given object.

download

Nowe w wersji 0.7: Obsługiwane od wersji wlc 0.7.

Pobierz plik z tłumaczeniami.

--convert

Converts file format, if unspecified no conversion happens on the server and the file is downloaded as is to the repository.

--output

Specifies file to save output in, if left unspecified it is printed to stdout.

upload

Nowe w wersji 0.9: Obsługiwane od wersji wlc 0.9.

Przesyła plik tłumaczenia.

--overwrite

Overwrite existing translations upon uploading.

--input
File from which content is read, if left unspecified it is read from stdin.

--method
Upload method to use, see *Metody importu*.

--fuzzy
Fuzzy (marked for edit) strings processing (*empty*, `process`, `approve`)

--author-name
Author name, to override currently authenticated user

--author-email
Author e-mail, to override currently authenticated user

Podpowiedź: You can get more detailed information on invoking individual commands by passing `--help`, for example: `wlc ls --help`.

1.13.6 Plik konfiguracyjny

.weblate, .weblate.ini, weblate.ini
Zmienione w wersji 1.6: The files with *.ini* extension are accepted as well.

Per project configuration file

C:\Users\NAME\AppData\weblate.ini
Nowe w wersji 1.6.

Plik konfiguracyjny użytkownika dla Windows.

~/.config/weblate
Plik konfiguracyjny użytkownika

/etc/xdg/weblate
Plik konfiguracyjny dla całego systemu

The program follows the XDG specification, so you can adjust placement of config files by environment variables `XDG_CONFIG_HOME` or `XDG_CONFIG_DIRS`. On Windows `APPDATA` directory is preferred location for the configuration file.

Following settings can be configured in the `[weblate]` section (you can customize this by *`--config-section`*):

key
API KEY to access Weblate.

url
API server URL, defaults to `http://127.0.0.1:8000/api/`.

translation
Path to the default translation - component or project.

The configuration file is an INI file, for example:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

Additionally API keys can be stored in the `[keys]` section:

```
[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.

1.13.7 Przykłady

Wydrukuj aktualną wersję programu:

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

Lista wszystkich projektów:

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

Prześlij plik tłumaczenia:

```
$ wlc upload project/component/language --input /tmp/hello.po
```

You can also designate what project wlc should work on:

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

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

With this setup it is easy to commit pending changes in the current project:

```
$ wlc commit
```

1.14 Weblate Python API

1.14.1 Instalacja

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

```
pip install wlc
```

1.14.2 wlc

WeblateException

exception wlc.**WeblateException**

Base class for all exceptions.

Weblate

class wlc.**Weblate**(*key=''*, *url=None*, *config=None*)

Parametry

- **key** (*str*) – Klucz użytkownika
- **url** (*str*) – API server URL, if not specified default is used
- **config** (wlc.config.WeblateConfig) – Configuration object, overrides any other parameters.

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

get(*path*)

Parametry

path (*str*) – Ścieżka żądania

Typ zwracany

object

Performs a single API GET call.

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

Parametry

path (*str*) – Ścieżka żądania

Typ zwracany

object

Performs a single API GET call.

1.14.3 wlc.config

WeblateConfig

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

Parametry
section (*str*) – Sekcja konfiguracji do użycia

Configuration file parser following XDG specification.

load(*path=None*)

Parametry
path (*str*) – Path from which to load configuration.

Loads configuration from a file, if none is specified, it loads from the *wlc* configuration file (`~/.config/wlc`) placed in your XDG configuration path (`/etc/xdg/wlc`).

1.14.4 wlc.main

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

Parametry

- **settings** (*list*) – Settings to override as list of tuples
- **stdout** (*object*) – stdout file object for printing output, uses `sys.stdout` as default
- **args** (*list*) – Command-line arguments to process, uses `sys.args` as default

Main entry point for command-line interface.

@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*)

Main class for invoking commands.

ROZDZIAŁ 2

Dokumenty administratora

2.1 Instrukcje konfiguracji

2.1.1 Instalacja Weblate

Instalowanie za pomocą Dockera

Dzięki zadokeryzowanemu wdrożeniu Weblate możesz uruchomić swoją osobistą instancję Weblate w kilka sekund. Wszystkie zależności Weblate są już uwzględnione. PostgreSQL jest skonfigurowany jako domyślna baza danych.

Wymagania sprzętowe

Weblate powinien działać bez problemów na każdym współczesnym sprzęcie, poniżej przedstawiono minimalną konfigurację wymaganą do uruchomienia Weblate na jednym hoście (Weblate, baza danych i serwer sieciowy):

- 2 GB pamięci RAM
- 2 rdzenie procesora
- 1 GB miejsca

Im więcej pamięci, tym lepiej - wykorzystywana jest do buforowania na wszystkich poziomach (system plików, baza danych i Weblate).

Wiele użytkowników jednocześnie zwiększa liczbę potrzebnych rdzeni procesora. W przypadku setek elementów tłumaczenia zalecane jest co najmniej 4 GB pamięci RAM.

Typowe użycie magazynu bazy danych wynosi około 300 MB na 1 milion hostowanych słów. Miejsce do magazynowania potrzebne do sklonowanych repozytoriów jest różna, ale Weblate stara się zachować ich minimalny rozmiar, wykonując płytkie klonowanie.

Informacja: Rzeczywiste wymagania dotyczące instalacji Weblate różnią się znacznie w zależności od rozmiaru zarządzanych tłumaczeń.

Instalacja

Poniższe przykłady zakładają, że masz działające środowisko Docker z zainstalowanym `docker-compose`. Instrukcje można znaleźć w dokumentacji platformy Docker.

1. Klonowanie repozytorium weblate-docker:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Utwórz plik: *docker-compose.override.yml* z ustawieniami. Zobacz *Zmienne środowiskowe platformy Docker* aby zapoznać się z pełną listą zmiennych środowiskowych.

```
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
```

Informacja: Jeżeli *WEBLATE_ADMIN_PASSWORD* nie jest ustawione, administrator jest tworzony z losowym hasłem wyświetlanym przy pierwszym uruchomieniu.

Podany przykład sprawia, że Weblate nasłuchuje na porcie 80, edytuj mapowanie portów w pliku `docker-compose.override.yml` aby to zmienić.

3. Uruchamianie kontenera Weblate:

```
docker-compose up
```

Ciesz się wdrożeniem, Weblate jest teraz dostępne na porcie 80 kontenera `` weblate``.

Zmienione w wersji 2.15-2: Konfiguracja zmieniła się niedawno, wcześniej istniał oddzielny kontener serwera WWW, od wersji 2.15-2 serwer WWW jest osadzony w kontenerze Weblate.

Zmienione w wersji 3.7.1-6: W lipcu 2019 r. (Począwszy od tagu 3.7.1-6) kontenery nie działają jako użytkownik root. Zmieniło to odsłonięty port 80 na 8080.

Zobacz także:

Wywoływanie poleceń zarządzania

Choosing Docker hub tag

You can use following tags on Docker hub, see https://hub.docker.com/r/weblate/weblate/tags/ for full list of available ones.

Nazwa znacznika	Opis	Przypadek użycia
`latest`	Weblate stable release, matches latest tagged release	Rolling updates in a production environment
`<VERSION>-<PATCH>`	Wersja stabilna Weblate	Well defined deploy in a production environment
`edge`	Weblate stable release with development changes in the Docker container (for example updated dependencies)	Rolling updates in a staging environment
`edge-<DATE>-<SHA>`	Weblate stable release with development changes in the Docker container (for example updated dependencies)	Well defined deploy in a staging environment
`bleeding`	Development version Weblate from Git	Rollling updates to test upcoming Weblate features
`bleeding-<DATE>-<SHA>`	Development version Weblate from Git	Well defined deploy to test upcoming Weblate features

Every image is tested by our CI before it gets published, so even the *bleeding* version should be quite safe to use.

Kontener platformy Docker z obsługą protokołu HTTPS

Zobacz *Instalacja*, aby zapoznać się z ogólnymi instrukcjami dotyczącymi wdrażania, ta sekcja wspomina tylko o różnicach w porównaniu z ogólną instrukcją.

Używanie własnych certyfikatów SSL

Nowe w wersji 3.8-3.

Jeśli masz własny certyfikat SSL, którego chcesz użyć, po prostu umieść pliki w woluminie danych Weblate (zobacz *Woluminy kontenerów platformy Docker*):

- : file:*ssl / fullchain.pem* certyfikat zawierający wszystkie potrzebne certyfikaty CA
- `ssl/privkey.pem` zawierający klucz prywatny

Oba te pliki muszą być własnością tego samego użytkownika, który uruchamia kontener docker i mieć maskę pliku ustawioną na «»600»» (czytelną i zapisywalną tylko przez użytkownika będącego właścicielem).

Dodatkowo, kontener Weblate będzie teraz akceptował połączenia SSL na porcie 4443, będziesz musiał uwzględnić przekierowanie portu dla HTTPS w docker-compose.override.yml:

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
      - 443:4443
```

Jeśli masz już inne strony na tym samym serwerze, prawdopodobnie porty `80` i `443` są używane przez odwrotny serwer proxy, taki jak NGINX. Aby przekazać połączenie HTTPS z NGINX do kontenera dockerowego, można użyć następującej konfiguracji:

```
server {
    listen 443;
    listen [::]:443;
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    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>;
    }
}
```

Zastąp `<SITE_URL>`, `<SITE>` i `<EXPOSED_DOCKER_PORT>` rzeczywistymi wartościami z twojego środowiska.

Automatyczne certyfikaty SSL z wykorzystaniem Let's Encrypt

In case you want to use Let's Encrypt automatically generated SSL certificates on public installation, you need to add a reverse HTTPS proxy an additional Docker container, https-portal will be used for that. This is made use of in the `docker-compose-https.yml` file. Then create a `docker-compose-https.override.yml` file with your settings:

```
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'
```

Whenever invoking **`docker-compose`** you need to pass both files to it, and then do:

```
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
```

Uaktualnianie kontenera platformy Docker

Usually it is good idea to only update the Weblate container and keep the PostgreSQL container at the version you have, as upgrading PostgreSQL is quite painful and in most cases does not bring many benefits.

Zmienione w wersji 4.10-1: Since Weblate 4.10-1, the Docker container uses Django 4.0 what requires PostgreSQL 10 or newer, please upgrade it prior to upgrading Weblate. See *Uaktualnienie z 4.9 do 4.10* and *Uaktualnianie kontenera PostgreSQL*.

You can do this by sticking with the existing docker-compose and just pull the latest images and then restart:

```
# Fetch latest versions of the images
docker-compose pull
# Stop and destroy the containers
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
docker-compose down
# Spawn new containers in the background
docker-compose up -d
# Follow the logs during upgrade
docker-compose logs -f
```

The Weblate database should be automatically migrated on first startup, and there should be no need for additional manual actions.

Informacja: Upgrades across major versions are not supported by Weblate. For example, if you are on 3.x series and want to upgrade to 4.x, first upgrade to the latest 4.0.x-y image (at time of writing this it is the `4.0.4-5`), which will do the migration and then continue upgrading to newer versions.

You might also want to update the `docker-compose` repository, though it's not needed in most case. See *Uaktualnianie kontenera PostgreSQL* for upgrading the PostgreSQL server.

Uaktualnianie kontenera PostgreSQL

PostgreSQL containers do not support automatic upgrading between version, you need to perform the upgrade manually. Following steps show one of the options of upgrading.

Zobacz także:

https://github.com/docker-library/postgres/issues/37

1. Zatrzymaj kontener Weblate:

   ```
   docker-compose stop weblate cache
   ```

2. Utwórz kopię zapasową bazy danych:

   ```
   docker-compose exec database pg_dumpall --clean --username weblate > backup.sql
   ```

3. Zatrzymaj kontener bazy danych:

   ```
   docker-compose stop database
   ```

4. Usuń wolumin PostgreSQL:

   ```
   docker-compose rm -v database
   docker volume remove weblate_postgres-data
   ```

5. Adjust `docker-compose.yml` to use new PostgreSQL version.
6. Uruchom kontener bazy danych:

   ```
   docker-compose up -d database
   ```

7. Przywróć bazę danych z kopii zapasowej:

   ```
   cat backup.sql | docker-compose exec -T database psql --username weblate --
   ↪dbname postgres
   ```

8. Uruchom wszystkie pozostałe kontenery:

   ```
   docker-compose up -d
   ```

Zaloguj się jako administrator

After container setup, you can sign in as *admin* user with password provided in *WEBLATE_ADMIN_PASSWORD*, or a random password generated on first start if that was not set.

To reset *admin* password, restart the container with *WEBLATE_ADMIN_PASSWORD* set to new password.

Zobacz także:

WEBLATE_ADMIN_PASSWORD, *WEBLATE_ADMIN_NAME*, *WEBLATE_ADMIN_EMAIL*

Number of processes and memory consumption

The number of worker processes for both uWSGI and Celery is determined automatically based on number of CPUs. This works well for most cloud virtual machines as these typically have few CPUs and good amount of memory.

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

```
environment:
  WEBLATE_WORKERS: 2
```

You can also fine-tune individual worker categories:

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

Zobacz także:

WEBLATE_WORKERS, *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*, *CELERY_MEMORY_OPTIONS*, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*, *CELERY_BEAT_OPTIONS*, *WEB_WORKERS*

Scaling horizontally

Nowe w wersji 4.6.

You can run multiple Weblate containers to scale the service horizontally. The `/app/data` volume has to be shared by all containers, it is recommended to use cluster filesystem such as GlusterFS for this. The `/app/cache` volume should be separate for each container.

Each Weblate container has defined role using *WEBLATE_SERVICE* environment variable. Please follow carefully the documentation as some of the services should be running just once in the cluster and the ordering of the services matters as well.

You can find example setup in the `docker-compose` repo as docker-compose-split.yml.

Zmienne środowiskowe platformy Docker

Many of Weblate's *Konfiguracja* can be set in the Docker container using environment variables:

Ustawienia ogólne

WEBLATE_DEBUG

Configures Django debug mode using `DEBUG`.

Przykład:

```
environment:
  WEBLATE_DEBUG: 1
```

Zobacz także:

Wyłączenie trybu debugowania

WEBLATE_LOGLEVEL

Konfiguruje szczegółowość rejestrowania.

WEBLATE_LOGLEVEL_DATABASE

Konfiguruje rejestrowanie szczegółowości zapytań bazy danych.

WEBLATE_SITE_TITLE

Changes the site-title shown in the header of all pages.

WEBLATE_SITE_DOMAIN

Konfiguruje domenę strony. Ten parametr jest wymagany.

Zobacz także:

Ustaw prawidłową domenę strony, `SITE_DOMAIN`

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Configures the site-admin's name and e-mail. It is used for both `ADMINS` setting and creating *admin* user (see `WEBLATE_ADMIN_PASSWORD` for more info on that).

Przykład:

```
environment:
  WEBLATE_ADMIN_NAME: Weblate admin
  WEBLATE_ADMIN_EMAIL: noreply@example.com
```

Zobacz także:

Zaloguj się jako administrator, *Prawidłowe konfigurowanie administratorów*, `ADMINS`

WEBLATE_ADMIN_PASSWORD

Sets the password for the *admin* user.

- If not set and *admin* user does not exist, it is created with a random password shown on first container startup.
- If not set and *admin* user exists, no action is performed.
- If set the *admin* user is adjusted on every container startup to match `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME` and `WEBLATE_ADMIN_EMAIL`.

> **Ostrzeżenie:** It might be a security risk to store password in the configuration file. Consider using this variable only for initial setup (or let Weblate generate random password on initial startup) or for password recovery.

Zobacz także:

Zaloguj się jako administrator, *WEBLATE_ADMIN_PASSWORD*, *WEBLATE_ADMIN_PASSWORD_FILE*, *WEBLATE_ADMIN_NAME*, *WEBLATE_ADMIN_EMAIL*

WEBLATE_ADMIN_PASSWORD_FILE

Sets the path to a file containing the password for the *admin* user.

Zobacz także:

WEBLATE_ADMIN_PASSWORD

WEBLATE_SERVER_EMAIL

The email address that error messages are sent from.

Zobacz także:

SERVER_EMAIL, *Konfigurowanie wysyłania wiadomości e-mail*

WEBLATE_DEFAULT_FROM_EMAIL

Configures the address for outgoing e-mails.

Zobacz także:

DEFAULT_FROM_EMAIL, *Konfigurowanie wysyłania wiadomości e-mail*

WEBLATE_CONTACT_FORM

Configures contact form behavior, see *CONTACT_FORM*.

WEBLATE_ALLOWED_HOSTS

Configures allowed HTTP hostnames using *ALLOWED_HOSTS*.

Defaults to `*` which allows all hostnames.

Przykład:

```
environment:
  WEBLATE_ALLOWED_HOSTS: weblate.example.com,example.com
```

Zobacz także:

ALLOWED_HOSTS, *Konfiguracja dozwolonych hostów*, *Ustaw prawidłową domenę strony*

WEBLATE_REGISTRATION_OPEN

Configures whether registrations are open by toggling *REGISTRATION_OPEN*.

Przykład:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Configure which authentication methods can be used to create new account via *REGISTRATION_ALLOW_BACKENDS*.

Przykład:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
  WEBLATE_REGISTRATION_ALLOW_BACKENDS: azuread-oauth2,azuread-tenant-
↪oauth2
```

WEBLATE_TIME_ZONE

Configures the used time zone in Weblate, see TIME_ZONE.

Informacja: To change the time zone of the Docker container itself, use the `TZ` environment variable.

Przykład:

```
environment:
  WEBLATE_TIME_ZONE: Europe/Prague
```

WEBLATE_ENABLE_HTTPS

Makes Weblate assume it is operated behind a reverse HTTPS proxy, it makes Weblate use HTTPS in e-mail and API links or set secure flags on cookies.

Podpowiedź: Please see *ENABLE_HTTPS* documentation for possible caveats.

Informacja: This does not make the Weblate container accept HTTPS connections, you need to configure that as well, see *Kontener platformy Docker z obsługą protokołu HTTPS* for examples.

Przykład:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Zobacz także:

ENABLE_HTTPS *Ustaw prawidłową domenę strony*, *WEBLATE_SECURE_PROXY_SSL_HEADER*

WEBLATE_INTERLEDGER_PAYMENT_POINTERS

Nowe w wersji 4.12.1.

Lets Weblate set the *meta[name=monetization]* field in the head of the document. If multiple are specified, chooses one randomly.

Zobacz także:

INTERLEDGER_PAYMENT_POINTERS

WEBLATE_IP_PROXY_HEADER

Lets Weblate fetch the IP address from any given HTTP header. Use this when using a reverse proxy in front of the Weblate container.

Enables *IP_BEHIND_REVERSE_PROXY* and sets *IP_PROXY_HEADER*.

Informacja: The format must conform to Django's expectations. Django transforms raw HTTP header names as follows:

- konwertuje wszystkie znaki na wielkie litery
- zastępuje wszystkie łączniki podkreśleniami
- poprzedza `HTTP_` prefiks

So `X-Forwarded-For` would be mapped to `HTTP_X_FORWARDED_FOR`.

Przykład:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

A tuple representing a HTTP header/value combination that signifies a request is secure. This is needed when Weblate is running behind a reverse proxy doing SSL termination which does not pass standard HTTPS headers.

Przykład:

```
environment:
  WEBLATE_SECURE_PROXY_SSL_HEADER: HTTP_X_FORWARDED_PROTO,https
```

Zobacz także:

SECURE_PROXY_SSL_HEADER

WEBLATE_REQUIRE_LOGIN

Enables *REQUIRE_LOGIN* to enforce authentication on whole Weblate.

Przykład:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

Adds URL exceptions for authentication required for the whole Weblate installation using *LOGIN_REQUIRED_URLS_EXCEPTIONS*.

You can either replace whole settings, or modify default value using ADD and REMOVE variables.

WEBLATE_GOOGLE_ANALYTICS_ID

Configures ID for Google Analytics by changing *GOOGLE_ANALYTICS_ID*.

WEBLATE_GITHUB_USERNAME

Configures GitHub username for GitHub pull-requests by changing *GITHUB_USERNAME*.

Zobacz także:

GitHub pull requests

WEBLATE_GITHUB_TOKEN

Nowe w wersji 4.3.

Configures GitHub personal access token for GitHub pull-requests via API by changing *GITHUB_TOKEN*.

Zobacz także:

GitHub pull requests

WEBLATE_GITLAB_USERNAME

Configures GitLab username for GitLab merge-requests by changing *GITLAB_USERNAME*

Zobacz także:

GitLab merge requests

WEBLATE_GITLAB_TOKEN

Configures GitLab personal access token for GitLab merge-requests via API by changing *GITLAB_TOKEN*

Zobacz także:

GitLab merge requests

WEBLATE_PAGURE_USERNAME

Configures Pagure username for Pagure merge-requests by changing *PAGURE_USERNAME*

Zobacz także:

Pagure merge requests

WEBLATE_PAGURE_TOKEN

Configures Pagure personal access token for Pagure merge-requests via API by changing *PAGURE_TOKEN*

Zobacz także:

Pagure merge requests

WEBLATE_DEFAULT_PULL_MESSAGE

Configures the default title and message for pull requests via API by changing *DEFAULT_PULL_MESSAGE*

Zobacz także:

DEFAULT_PULL_MESSAGE

WEBLATE_SIMPLIFY_LANGUAGES

Configures the language simplification policy, see *SIMPLIFY_LANGUAGES*.

WEBLATE_DEFAULT_ACCESS_CONTROL

Configures the default *Kontrola dostępu* for new projects, see *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Configures the default value for *Ograniczony dostęp* for new components, see *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Configures the default value for *Zezwól na propagację tłumaczenia* for new components, see *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

Konfiguruje *DEFAULT_COMMITER_EMAIL*.

WEBLATE_DEFAULT_COMMITER_NAME

Konfiguruje *DEFAULT_COMMITER_NAME*.

WEBLATE_DEFAULT_SHARED_TM

Konfiguruje *DEFAULT_SHARED_TM*.

WEBLATE_AKISMET_API_KEY

Configures the Akismet API key, see *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

Configures GPG signing of commits, see *WEBLATE_GPG_IDENTITY*.

Zobacz także:

Signing Git commits with GnuPG

WEBLATE_URL_PREFIX

Configures URL prefix where Weblate is running, see *URL_PREFIX*.

WEBLATE_SILENCED_SYSTEM_CHECKS

Configures checks which you do not want to be displayed, see SILENCED_SYSTEM_CHECKS.

WEBLATE_CSP_SCRIPT_SRC

WEBLATE_CSP_IMG_SRC

WEBLATE_CSP_CONNECT_SRC

WEBLATE_CSP_STYLE_SRC

WEBLATE_CSP_FONT_SRC

Allows to customize `Content-Security-Policy` HTTP header.

Zobacz także:

Polityka bezpieczeństwa treści, *CSP_SCRIPT_SRC*, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

WEBLATE_LICENSE_FILTER

Konfiguruje *LICENSE_FILTER*.

WEBLATE_LICENSE_REQUIRED

Konfiguruje *LICENSE_REQUIRED*

WEBLATE_WEBSITE_REQUIRED

Konfiguruje *WEBSITE_REQUIRED*

WEBLATE_HIDE_VERSION

Konfiguruje *HIDE_VERSION*.

WEBLATE_BASIC_LANGUAGES

Konfiguruje *BASIC_LANGUAGES*.

WEBLATE_DEFAULT_AUTO_WATCH

Konfiguruje *DEFAULT_AUTO_WATCH*.

WEBLATE_RATELIMIT_ATTEMPTS

WEBLATE_RATELIMIT_LOCKOUT

WEBLATE_RATELIMIT_WINDOW

Nowe w wersji 4.6.

Konfiguruje ogranicznik szybkości.

Podpowiedź: You can set configuration for any rate limiter scopes. To do that add `WEBLATE_` prefix to any of setting described in *Ograniczenie szybkości*.

Zobacz także:

Ograniczenie szybkości, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*, *RATELIMIT_LOCKOUT*

WEBLATE_API_RATELIMIT_ANON

WEBLATE_API_RATELIMIT_USER

Nowe w wersji 4.11.

Configures API rate limiting. Defaults to `100/day` for anonymous and `5000/hour` for authenticated users.

Zobacz także:

Ograniczanie szybkości interfejsu API

WEBLATE_ENABLE_HOOKS

Nowe w wersji 4.13.

Configures *ENABLE_HOOKS*.

WEBLATE_ENABLE_AVATARS

Nowe w wersji 4.6.1.

Konfiguruje *ENABLE_AVATARS*.

WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Nowe w wersji 4.9.

Konfiguruje *LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH*.

WEBLATE_SSH_EXTRA_ARGS

Nowe w wersji 4.9.

Konfiguruje *SSH_EXTRA_ARGS*.

WEBLATE_BORG_EXTRA_ARGS

Nowe w wersji 4.9.

Konfiguruje *BORG_EXTRA_ARGS*.

Ustawienia automatycznych sugestii

Zmienione w wersji 4.13: Automatic suggestion services are now configured in the user interface, see *Konfigurowanie automatycznych sugestii*.

The existing environment variables are imported during the migration to Weblate 4.13, but changing them will not have any further effect.

Ustawienia uwierzytelniania

LDAP

WEBLATE_AUTH_LDAP_SERVER_URI

WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE

WEBLATE_AUTH_LDAP_USER_ATTR_MAP

WEBLATE_AUTH_LDAP_BIND_DN

WEBLATE_AUTH_LDAP_BIND_PASSWORD

WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE

Ścieżka do pliku zawierającego hasło powiązania serwera LDAP.

Zobacz także:

WEBLATE_AUTH_LDAP_BIND_PASSWORD

WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS

WEBLATE_AUTH_LDAP_USER_SEARCH

WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER

WEBLATE_AUTH_LDAP_USER_SEARCH_UNION

WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER

Konfiguracja uwierzytelniania LDAP.

** Przykład bezpośredniego powiązania: **

```
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.
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
  # another example that can be used with OpenLDAP: 'full_name:cn,email:mail'
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
```

Example for search and bind:

```
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
```

Example for union search and bind:

```
environment:
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH_UNION: ou=users,dc=example,
↪dc=com|ou=otherusers,dc=example,dc=com
```

Example with search and bind against 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)
```

Zobacz także:

Uwierzytelnianie LDAP

GitHub

WEBLATE_SOCIAL_AUTH_GITHUB_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID

Włączanie *Uwierzytelnianie GitHub*.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET

Włączanie *Uwierzytelnianie Bitbucket*.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET

Włączanie *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

Włączanie *Google OAuth 2*.

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY

WEBLATE_SOCIAL_AUTH_GITLAB_SECRET

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL

Włączanie *GitLab OAuth 2*.

Azure Active Directory

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET

Enables Azure Active Directory authentication, see *Microsoft Azure Active Directory*.

Azure Active Directory with Tenant support

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID

Enables Azure Active Directory authentication with Tenant support, see *Microsoft Azure Active Directory*.

Keycloak

WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY

WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET

WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY

WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM

WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL

WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL

WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE

WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE

Enables Keycloak authentication, see documentation.

Dostawcy linuksa

You can enable authentication using Linux vendors authentication services by setting following variables to any value.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Włącza uwierzytelnianie Slack, zobacz *Slack*.

OpenID Connect

Nowe w wersji 4.13-1.

WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT

WEBLATE_SOCIAL_AUTH_OIDC_KEY

WEBLATE_SOCIAL_AUTH_OIDC_SECRET

WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY

Configures generic OpenID Connect intergration.

Zobacz także:

OIDC (OpenID Connect)

SAML

Self-signed SAML keys are automatically generated on first container startup. In case you want to use own keys, place the certificate and private key in `/app/data/ssl/saml.crt` and `/app/data/ssl/saml.key`.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

WEBLATE_SAML_IDP_IMAGE

WEBLATE_SAML_IDP_TITLE

SAML Identity Provider settings, see *Uwierzytelnianie SAML*.

Inne ustawienia uwierzytelniania

WEBLATE_NO_EMAIL_AUTH

Wyłącza uwierzytelnianie poczty e-mail, gdy jest ustawiona na dowolną wartość. Zobacz disable-e-mail-auth.

Konfiguracja bazy danych PostgreSQL

The database is created by `docker-compose.yml`, so these settings affect both Weblate and PostgreSQL containers.

Zobacz także:

Konfiguracja bazy danych dla Weblate

POSTGRES_PASSWORD

Hasło PostgreSQL.

POSTGRES_PASSWORD_FILE

Path to the file containing the PostgreSQL password. Use as an alternative to POSTGRES_PASSWORD.

POSTGRES_USER

Nazwa użytkownika PostgreSQL.

POSTGRES_DATABASE

Nazwa bazy danych PostgreSQL.

POSTGRES_HOST

PostgreSQL server hostname or IP address. Defaults to `database`.

POSTGRES_PORT

PostgreSQL server port. Defaults to none (uses the default value).

POSTGRES_SSL_MODE

Configure how PostgreSQL handles SSL in connection to the server, for possible choices see SSL Mode Descriptions

POSTGRES_ALTER_ROLE

Configures name of role to alter during migrations, see *Konfigurowanie Weblate do korzystania z PostgreSQL*.

POSTGRES_CONN_MAX_AGE

Nowe w wersji 4.8.1.

The lifetime of a database connection, as an integer of seconds. Use 0 to close database connections at the end of each request (this is the default behavior).

Enabling connection persistence will typically, cause more open connection to the database. Please adjust your database configuration prior enabling.

Przykładowa konfiguracja:

```
environment:
    POSTGRES_CONN_MAX_AGE: 3600
```

Zobacz także:

`CONN_MAX_AGE`, Persistent connections

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

Nowe w wersji 4.9.1.

Disable server side cursors in the database. This is necessary in some **pgbouncer** setups.

Przykładowa konfiguracja:

```
environment:
    POSTGRES_DISABLE_SERVER_SIDE_CURSORS: 1
```

Zobacz także:

`DISABLE_SERVER_SIDE_CURSORS`, Transaction pooling and server-side cursors

Ustawienia kopii zapasowej bazy danych

Zobacz także:

Zrzucone dane do kopii zapasowych

WEBLATE_DATABASE_BACKUP

Configures the daily database dump using *`DATABASE_BACKUP`*. Defaults to `plain`.

Konfiguracja serwera buforowania

Using Redis is strongly recommended by Weblate and you have to provide a Redis instance when running Weblate in Docker.

Zobacz także:

Włączanie buforowania

REDIS_HOST

The Redis server hostname or IP address. Defaults to `cache`.

REDIS_PORT

The Redis server port. Defaults to `6379`.

REDIS_DB

The Redis database number, defaults to `1`.

REDIS_PASSWORD

The Redis server password, not used by default.

REDIS_PASSWORD_FILE

Ścieżka do pliku zawierającego hasło serwera Redis.

Zobacz także:

REDIS_PASSWORD

REDIS_TLS

Enables using SSL for Redis connection.

REDIS_VERIFY_SSL

Can be used to disable SSL certificate verification for Redis connection.

Konfiguracja serwera poczty e-mail

To make outgoing e-mail work, you need to provide a mail server.

Przykładowa konfiguracja protokołu TLS:

```
environment:
    WEBLATE_EMAIL_HOST: smtp.example.com
    WEBLATE_EMAIL_HOST_USER: user
    WEBLATE_EMAIL_HOST_PASSWORD: pass
```

Przykładowa konfiguracja 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
```

Zobacz także:

Konfigurowanie poczty wychodzącej

WEBLATE_EMAIL_HOST

Mail server hostname or IP address.

Zobacz także:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_SSL*, *WEBLATE_EMAIL_USE_TLS*, EMAIL_HOST

WEBLATE_EMAIL_PORT

Mail server port, defaults to 25.

Zobacz także:

EMAIL_PORT

WEBLATE_EMAIL_HOST_USER

Uwierzytelnienie za pomocą e-maila.

Zobacz także:

EMAIL_HOST_USER

WEBLATE_EMAIL_HOST_PASSWORD

Hasło uwierzytelniania poczty e-mail.

Zobacz także:

EMAIL_HOST_PASSWORD

WEBLATE_EMAIL_HOST_PASSWORD_FILE

Ścieżka do pliku zawierającego hasło uwierzytelniające e-mail.

Zobacz także:

WEBLATE_EMAIL_HOST_PASSWORD

WEBLATE_EMAIL_USE_SSL

Whether to use an implicit TLS (secure) connection when talking to the SMTP server. In most e-mail documentation, this type of TLS connection is referred to as SSL. It is generally used on port 465. If you are experiencing problems, see the explicit TLS setting *WEBLATE_EMAIL_USE_TLS*.

Zmienione w wersji 4.11: The SSL/TLS support is automatically enabled based on the *WEBLATE_EMAIL_PORT*.

Zobacz także:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_TLS*, EMAIL_USE_SSL

WEBLATE_EMAIL_USE_TLS

Whether to use a TLS (secure) connection when talking to the SMTP server. This is used for explicit TLS connections, generally on port 587 or 25. If you are experiencing connections that hang, see the implicit TLS setting *WEBLATE_EMAIL_USE_SSL*.

Zmienione w wersji 4.11: The SSL/TLS support is automatically enabled based on the *WEBLATE_EMAIL_PORT*.

Zobacz także:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_SSL*, EMAIL_USE_TLS

WEBLATE_EMAIL_BACKEND

Configures Django back-end to use for sending e-mails.

Zobacz także:

Konfigurowanie wysyłania wiadomości e-mail, EMAIL_BACKEND

WEBLATE_AUTO_UPDATE

Konfiguruje, czy i jak Weblate powinno aktualizować repozytoria.

Zobacz także:

AUTO_UPDATE

Informacja: This is a Boolean setting (use `"true"` or `"false"`).

Integracja strony

WEBLATE_GET_HELP_URL

Konfiguruje *GET_HELP_URL*.

WEBLATE_STATUS_URL

Konfiguruje *STATUS_URL*.

WEBLATE_LEGAL_URL

Konfiguruje *LEGAL_URL*.

WEBLATE_PRIVACY_URL

Konfiguruje *PRIVACY_URL*.

Raportowanie błędów

It is recommended to collect errors from the installation systematically, see *Zbieranie raportów o błędach*.

To enable support for Rollbar, set the following:

ROLLBAR_KEY

Your Rollbar post server access token.

ROLLBAR_ENVIRONMENT

Your Rollbar environment, defaults to `production`.

To enable support for Sentry, set following:

SENTRY_DSN

Twój DSN Sentry.

SENTRY_ENVIRONMENT

Twoje środowisko Sentry (opcjonalnie).

Lokalizacja CDN

WEBLATE_LOCALIZE_CDN_URL

WEBLATE_LOCALIZE_CDN_PATH

Nowe w wersji 4.2.1.

Konfiguracja dla *Lokalizacja JavaScript CDN*.

The *WEBLATE_LOCALIZE_CDN_PATH* is path within the container. It should be stored on the persistent volume and not in the transient storage.

One of possibilities is storing that inside the Weblate data dir:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

Informacja: You are responsible for setting up serving of the files generated by Weblate, it only does stores the files in configured location.

Zobacz także:

weblate-cdn, *LOCALIZE_CDN_URL*, *LOCALIZE_CDN_PATH*

Changing enabled apps, checks, add-ons or autofixes

Nowe w wersji 3.8-5.

The built-in configuration of enabled checks, add-ons or autofixes can be adjusted by the following variables:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Przykład:

```
environment:
  WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
↪SameBookendingWhitespace
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon,customize.addons.OtherAddon
```

Zobacz także:

CHECK_LIST, *AUTOFIX_LIST*, *WEBLATE_ADDONS*, INSTALLED_APPS

Ustawienia kontenera

WEBLATE_WORKERS

Nowe w wersji 4.6.1.

Base number of worker processes running in the container. When not set it is determined automatically on container startup based on number of CPU cores available.

It is used to determine *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*, *CELERY_MEMORY_OPTIONS*, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*, *CELERY_BEAT_OPTIONS*, and *WEB_WORKERS*. You can use these settings to fine-tune.

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

These variables allow you to adjust Celery worker options. It can be useful to adjust concurrency (`--concurrency 16`) or use different pool implementation (`--pool=gevent`).

By default, the number of concurrent workers is based on *WEBLATE_WORKERS*.

Przykład:

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

Zobacz także:

Celery worker options, *Zadania w tle korzystające z Celery*

WEB_WORKERS

Configure how many uWSGI workers should be executed.

It defaults to `WEBLATE_WORKERS`.

Przykład:

```
environment:
  WEB_WORKERS: 32
```

WEBLATE_SERVICE

Defines which services should be executed inside the container. Use this for *Scaling horizontally*.

Zdefiniowane są następujące usługi:

celery-beat
Celery task scheduler, only one instance should be running. This container is also responsible for the database structure migrations and it should be started prior others.

celery-backup
Celery worker for backups, only one instance should be running.

celery-celery
Generic Celery worker.

celery-memory
Pamięć tłumaczeniowa workera Celery.

celery-notify
Powiadomienia workera Celery.

celery-translate
Tłumaczenie automatyczne workera Celery.

web
Serwer internetowy.

Woluminy kontenerów platformy Docker

There are two volumes (data and cache) exported by the Weblate container. The other service containers (PostgreSQL or Redis) have their data volumes as well, but those are not covered by this document.

The data volume is used to store Weblate persistent data such as cloned repositories or to customize Weblate installation.

The placement of the Docker volume on host system depends on your Docker configuration, but usually it is stored in `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/` (the path consist of name of your docker-compose directory, container, and volume names). In the container it is mounted as `/app/data`.

The cache volume is mounted as `/app/cache` and is used to store static files. Its content is recreated on container startup and the volume can be mounted using ephemeral filesystem such as *tmpfs*.

When creating the volumes manually, the directories should be owned by UID 1000 as that is user used inside the container.

Zobacz także:

Dokumentacja woluminów platformy Docker

Dalsze dostosowywanie konfiguracji

You can further customize Weblate installation in the data volume, see *Woluminy kontenerów platformy Docker*.

Niestandardowe pliki konfiguracyjne

You can additionally override the configuration in `/app/data/settings-override.py` (see *Woluminy kontenerów platformy Docker*). This is executed at the end of built-in settings, after all environment settings are loaded, and you can adjust or override them.

Replacing logo and other static files

Nowe w wersji 3.8-5.

The static files coming with Weblate can be overridden by placing into `/app/data/python/customize/static` (see *Woluminy kontenerów platformy Docker*). For example creating `/app/data/python/customize/static/favicon.ico` will replace the favicon.

Podpowiedź: The files are copied to the corresponding location upon container startup, so a restart of Weblate is needed after changing the content of the volume.

This approach can be also used to override Weblate templates. For example *Informacje prawne* documents can be placed into `/app/data/python/customize/templates/legal/documents`.

Alternatively you can also include own module (see *Dostosowywanie Weblate*) and add it as separate volume to the Docker container, for example:

```
weblate:
  volumes:
    - weblate-data:/app/data
    - ./weblate_customization/weblate_customization:/app/data/python/weblate_
↪customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

Dodawanie własnych modułów Pythona

Nowe w wersji 3.8-5.

You can place own Python modules in `/app/data/python/` (see *Woluminy kontenerów platformy Docker*) and they can be then loaded by Weblate, most likely by using *Niestandardowe pliki konfiguracyjne*.

Zobacz także:

Dostosowywanie Weblate

Konfigurowanie serwera PostgreSQL

The PostgtreSQL container uses default PostgreSQL configuration and it won't effectively utilize your CPU cores or memory. It is recommended to customize the configuration to improve the performance.

The configuration can be adjusted as described in *Database Configuration* at https://hub.docker.com/_/postgres. The configuration matching your environment can be generated using https://pgtune.leopard.in.ua/.

Instalacja na Debianie i Ubuntu

Wymagania sprzętowe

Weblate powinien działać bez problemów na każdym współczesnym sprzęcie, poniżej przedstawiono minimalną konfigurację wymaganą do uruchomienia Weblate na jednym hoście (Weblate, baza danych i serwer sieciowy):

- 2 GB pamięci RAM
- 2 rdzenie procesora
- 1 GB miejsca

Im więcej pamięci, tym lepiej - wykorzystywana jest do buforowania na wszystkich poziomach (system plików, baza danych i Weblate).

Wiele użytkowników jednocześnie zwiększa liczbę potrzebnych rdzeni procesora. W przypadku setek elementów tłumaczenia zalecane jest co najmniej 4 GB pamięci RAM.

Typowe użycie magazynu bazy danych wynosi około 300 MB na 1 milion hostowanych słów. Miejsce do magazynowania potrzebne do sklonowanych repozytoriów jest różna, ale Weblate stara się zachować ich minimalny rozmiar, wykonując płytkie klonowanie.

Informacja: Rzeczywiste wymagania dotyczące instalacji Weblate różnią się znacznie w zależności od rozmiaru zarządzanych tłumaczeń.

Instalacja

Wymagania systemowe

Install the dependencies needed to build the Python modules (see *Wymagania dotyczące oprogramowania*):

```
apt install \
   libxml2-dev libxslt-dev libfreetype6-dev libjpeg-dev libz-dev libyaml-dev \
   libffi-dev libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev \
   libacl1-dev libssl-dev libpq-dev libjpeg62-turbo-dev build-essential \
   python3-gdbm python3-dev python3-pip python3-virtualenv virtualenv git
```

Install wanted optional dependencies depending on features you intend to use (see *Opcjonalne zależności*):

```
apt install tesseract-ocr libtesseract-dev libleptonica-dev
apt install libldap2-dev libldap-common libsasl2-dev
apt install libxmlsec1-dev
```

Optionally install software for running production server, see *Uruchamianie serwera*, *Konfiguracja bazy danych dla Weblate*, *Zadania w tle korzystające z Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Instrukcja instalacji lokalnej:

```
# 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-py3

# Caching backend: Redis
apt install redis-server

# Database server: PostgreSQL
apt install postgresql postgresql-contrib

# SMTP server
apt install exim4
```

Moduły Pythona

Podpowiedź: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv User Guide.

1. Utwórz virtualenv dla Weblate:

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

2. Utwórz virtualenv dla Weblate:

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

3. Zainstaluj Weblate wraz ze wszystkimi opcjonalnymi zależnościami:

   ```
   # pkgconfig is needed to install borgbackup 1.2
   pip install pkgconfig
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

   Please check *Opcjonalne zależności* for fine-tuning of optional dependencies.

   Informacja: On some Linux distributions running Weblate fails with libffi error:

   ```
   ffi_prep_closure(): bad user_data (it seems that the version of the libffi␣
   ↪library seen at runtime is different from the 'ffi.h' file seen at compile-
   ↪time)
   ```

   This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

   ```
   pip install --force-reinstall --no-binary :all: cffi
   ```

Konfigurowanie Weblate

Informacja: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **`weblate`** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Dostosowywanie konfiguracji*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Konfiguracja bazy danych dla Weblate* for production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see *Uruchamianie serwera* and *Obsługa plików statycznych*):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see *Kompresowanie zasobów klienta*):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See *Zadania w tle korzystające z Celery* for more info:

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

8. Start the development server (see *Uruchamianie serwera* for production setup):

```
weblate runserver
```

Po instalacji

Congratulations, your Weblate server is now running and you can start using it.

- You can now access Weblate on `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **`weblate`** command when Weblate virtualenv is active, see *Polecenia zarządzania*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfejs zarządzania*) or using **`weblate check --deploy`**, see *Instalacja produkcyjna*.

Dodawanie tłumaczenia

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Konfiguracja projektu* for more details.

   All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Konfiguracja komponentu* for more details.

   The important fields here are: *Nazwa komponentu*, *Repozytorium kodu źródłowego*, and *Maska pliku* for finding translatable files. Weblate supports a wide range of formats including *GNU gettext*, *Android string resources*, *Ciągi Apple iOS*, *Właściwości Java*, *Format stringsdict* or *Format Fluent*, see *Obsługiwane formaty plików* for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Instalacja na SUSE i openSUSE

Wymagania sprzętowe

Weblate powinien działać bez problemów na każdym współczesnym sprzęcie, poniżej przedstawiono minimalną konfigurację wymaganą do uruchomienia Weblate na jednym hoście (Weblate, baza danych i serwer sieciowy):

- 2 GB pamięci RAM
- 2 rdzenie procesora
- 1 GB miejsca

Im więcej pamięci, tym lepiej - wykorzystywana jest do buforowania na wszystkich poziomach (system plików, baza danych i Weblate).

Wiele użytkowników jednocześnie zwiększa liczbę potrzebnych rdzeni procesora. W przypadku setek elementów tłumaczenia zalecane jest co najmniej 4 GB pamięci RAM.

Typowe użycie magazynu bazy danych wynosi około 300 MB na 1 milion hostowanych słów. Miejsce do magazynowania potrzebne do sklonowanych repozytoriów jest różna, ale Weblate stara się zachować ich minimalny rozmiar, wykonując płytkie klonowanie.

Informacja: Rzeczywiste wymagania dotyczące instalacji Weblate różnią się znacznie w zależności od rozmiaru zarządzanych tłumaczeń.

Instalacja

Wymagania systemowe

Install the dependencies needed to build the Python modules (see *Wymagania dotyczące oprogramowania*):

```
zypper install \
   libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
   libyaml-devel libffi-devel cairo-devel pango-devel \
   gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
   python3-devel git
```

Install wanted optional dependencies depending on features you intend to use (see *Opcjonalne zależności*):

```
zypper install tesseract-ocr tesseract-devel leptonica-devel
zypper install libldap2-devel libsasl2-devel
zypper install libxmlsec1-devel
```

Optionally install software for running production server, see *Uruchamianie serwera*, *Konfiguracja bazy danych dla Weblate*, *Zadania w tle korzystające z Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Instrukcja instalacji lokalnej:

```
# 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
```

Moduły Pythona

Podpowiedź: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv User Guide.

1. Utwórz virtualenv dla Weblate:

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

2. Utwórz virtualenv dla Weblate:

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

3. Zainstaluj Weblate wraz ze wszystkimi opcjonalnymi zależnościami:

   ```
   # pkgconfig is needed to install borgbackup 1.2
   pip install pkgconfig
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

   Please check *Opcjonalne zależności* for fine-tuning of optional dependencies.

   Informacja: On some Linux distributions running Weblate fails with libffi error:

   ```
   ffi_prep_closure(): bad user_data (it seems that the version of the libffi␣
   ↪library seen at runtime is different from the 'ffi.h' file seen at compile-
   ↪time)
   ```

   This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

   ```
   pip install --force-reinstall --no-binary :all: cffi
   ```

Konfigurowanie Weblate

Informacja: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **`weblate`** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Dostosowywanie konfiguracji*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Konfiguracja bazy danych dla Weblate* for production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see *Uruchamianie serwera* and *Obsługa plików statycznych*):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see *Kompresowanie zasobów klienta*):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See *Zadania w tle korzystające z Celery* for more info:

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

8. Start the development server (see *Uruchamianie serwera* for production setup):

```
weblate runserver
```

Po instalacji

Congratulations, your Weblate server is now running and you can start using it.

- You can now access Weblate on `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **`weblate`** command when Weblate virtualenv is active, see *Polecenia zarządzania*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfejs zarządzania*) or using **`weblate check --deploy`**, see *Instalacja produkcyjna*.

Dodawanie tłumaczenia

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Konfiguracja projektu* for more details.

   All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Konfiguracja komponentu* for more details.

   The important fields here are: *Nazwa komponentu*, *Repozytorium kodu źródłowego*, and *Maska pliku* for finding translatable files. Weblate supports a wide range of formats including *GNU gettext*, *Android string resources*, *Ciągi Apple iOS*, *Właściwości Java*, *Format stringsdict* or *Format Fluent*, see *Obsługiwane formaty plików* for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Installing on RedHat, Fedora and CentOS

Wymagania sprzętowe

Weblate powinien działać bez problemów na każdym współczesnym sprzęcie, poniżej przedstawiono minimalną konfigurację wymaganą do uruchomienia Weblate na jednym hoście (Weblate, baza danych i serwer sieciowy):

- 2 GB pamięci RAM
- 2 rdzenie procesora
- 1 GB miejsca

Im więcej pamięci, tym lepiej - wykorzystywana jest do buforowania na wszystkich poziomach (system plików, baza danych i Weblate).

Wiele użytkowników jednocześnie zwiększa liczbę potrzebnych rdzeni procesora. W przypadku setek elementów tłumaczenia zalecane jest co najmniej 4 GB pamięci RAM.

Typowe użycie magazynu bazy danych wynosi około 300 MB na 1 milion hostowanych słów. Miejsce do magazynowania potrzebne do sklonowanych repozytoriów jest różna, ale Weblate stara się zachować ich minimalny rozmiar, wykonując płytkie klonowanie.

Informacja: Rzeczywiste wymagania dotyczące instalacji Weblate różnią się znacznie w zależności od rozmiaru zarządzanych tłumaczeń.

Instalacja

Wymagania systemowe

Install the dependencies needed to build the Python modules (see *Wymagania dotyczące oprogramowania*):

```
dnf install \
   libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
   libyaml-devel libffi-devel cairo-devel pango-devel \
   gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
   python3-devel git
```

Install wanted optional dependencies depending on features you intend to use (see *Opcjonalne zależności*):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
dnf install libldap2-devel libsasl2-devel
dnf install libxmlsec1-devel
```

Optionally install software for running production server, see *Uruchamianie serwera*, *Konfiguracja bazy danych dla Weblate*, *Zadania w tle korzystające z Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Instrukcja instalacji lokalnej:

```
# 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
```

Moduły Pythona

Podpowiedź: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv User Guide.

1. Utwórz virtualenv dla Weblate:

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

2. Utwórz virtualenv dla Weblate:

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

3. Zainstaluj Weblate wraz ze wszystkimi opcjonalnymi zależnościami:

   ```
   # pkgconfig is needed to install borgbackup 1.2
   pip install pkgconfig
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

   Please check *Opcjonalne zależności* for fine-tuning of optional dependencies.

   Informacja: On some Linux distributions running Weblate fails with libffi error:

   ```
   ffi_prep_closure(): bad user_data (it seems that the version of the libffi␣
   ↪library seen at runtime is different from the 'ffi.h' file seen at compile-
   ↪time)
   ```

   This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

   ```
   pip install --force-reinstall --no-binary :all: cffi
   ```

Konfigurowanie Weblate

Informacja: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **`weblate`** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Dostosowywanie konfiguracji*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Konfiguracja bazy danych dla Weblate* for production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see *Uruchamianie serwera* and *Obsługa plików statycznych*):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see *Kompresowanie zasobów klienta*):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See *Zadania w tle korzystające z Celery* for more info:

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

8. Start the development server (see *Uruchamianie serwera* for production setup):

```
weblate runserver
```

Po instalacji

Congratulations, your Weblate server is now running and you can start using it.

- You can now access Weblate on `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **`weblate`** command when Weblate virtualenv is active, see *Polecenia zarządzania*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfejs zarządzania*) or using **`weblate check --deploy`**, see *Instalacja produkcyjna*.

Dodawanie tłumaczenia

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Konfiguracja projektu* for more details.

   All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Konfiguracja komponentu* for more details.

   The important fields here are: *Nazwa komponentu*, *Repozytorium kodu źródłowego*, and *Maska pliku* for finding translatable files. Weblate supports a wide range of formats including *GNU gettext*, *Android string resources*, *Ciągi Apple iOS*, *Właściwości Java*, *Format stringsdict* or *Format Fluent*, see *Obsługiwane formaty plików* for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Instalacja w systemie macOS

Wymagania sprzętowe

Weblate powinien działać bez problemów na każdym współczesnym sprzęcie, poniżej przedstawiono minimalną konfigurację wymaganą do uruchomienia Weblate na jednym hoście (Weblate, baza danych i serwer sieciowy):

- 2 GB pamięci RAM
- 2 rdzenie procesora
- 1 GB miejsca

Im więcej pamięci, tym lepiej - wykorzystywana jest do buforowania na wszystkich poziomach (system plików, baza danych i Weblate).

Wiele użytkowników jednocześnie zwiększa liczbę potrzebnych rdzeni procesora. W przypadku setek elementów tłumaczenia zalecane jest co najmniej 4 GB pamięci RAM.

Typowe użycie magazynu bazy danych wynosi około 300 MB na 1 milion hostowanych słów. Miejsce do magazynowania potrzebne do sklonowanych repozytoriów jest różna, ale Weblate stara się zachować ich minimalny rozmiar, wykonując płytkie klonowanie.

Informacja: Rzeczywiste wymagania dotyczące instalacji Weblate różnią się znacznie w zależności od rozmiaru zarządzanych tłumaczeń.

Instalacja

Wymagania systemowe

Install the dependencies needed to build the Python modules (see *Wymagania dotyczące oprogramowania*):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
pip3 install virtualenv
```

Make sure pip will be able to find the `libffi` version provided by homebrew — this will be needed during the installation build step.

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

Install wanted optional dependencies depending on features you intend to use (see *Opcjonalne zależności*):

```
brew install tesseract
```

Optionally install software for running production server, see *Uruchamianie serwera*, *Konfiguracja bazy danych dla Weblate*, *Zadania w tle korzystające z Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Instrukcja instalacji lokalnej:

```
# 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
```

Moduły Pythona

Podpowiedź: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv User Guide.

1. Utwórz virtualenv dla Weblate:

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

2. Utwórz virtualenv dla Weblate:

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

3. Zainstaluj Weblate wraz ze wszystkimi opcjonalnymi zależnościami:

   ```
   # pkgconfig is needed to install borgbackup 1.2
   pip install pkgconfig
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

   Please check *Opcjonalne zależności* for fine-tuning of optional dependencies.

   Informacja: On some Linux distributions running Weblate fails with libffi error:

   ```
   ffi_prep_closure(): bad user_data (it seems that the version of the libffi␣
   ↪library seen at runtime is different from the 'ffi.h' file seen at compile-
   ↪time)
   ```

   This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

   ```
   pip install --force-reinstall --no-binary :all: cffi
   ```

Konfigurowanie Weblate

Informacja: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **`weblate`** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Dostosowywanie konfiguracji*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Konfiguracja bazy danych dla Weblate* for production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see *Uruchamianie serwera* and *Obsługa plików statycznych*):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see *Kompresowanie zasobów klienta*):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See *Zadania w tle korzystające z Celery* for more info:

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

8. Start the development server (see *Uruchamianie serwera* for production setup):

```
weblate runserver
```

Po instalacji

Congratulations, your Weblate server is now running and you can start using it.

- You can now access Weblate on `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **`weblate`** command when Weblate virtualenv is active, see *Polecenia zarządzania*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfejs zarządzania*) or using **`weblate check --deploy`**, see *Instalacja produkcyjna*.

Dodawanie tłumaczenia

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Konfiguracja projektu* for more details.

   All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Konfiguracja komponentu* for more details.

   The important fields here are: *Nazwa komponentu*, *Repozytorium kodu źródłowego*, and *Maska pliku* for finding translatable files. Weblate supports a wide range of formats including *GNU gettext*, *Android string resources*, *Ciągi Apple iOS*, *Właściwości Java*, *Format stringsdict* or *Format Fluent*, see *Obsługiwane formaty plików* for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Instalacja ze źródeł

1. Please follow the installation instructions for your system first up to installing Weblate:

   - *Instalacja na Debianie i Ubuntu*
   - *Instalacja na SUSE i openSUSE*
   - *Installing on RedHat, Fedora and CentOS*

2. Grab the latest Weblate sources using Git (or download a tarball and unpack that):

   ```
   git clone https://github.com/WeblateOrg/weblate.git weblate-src
   ```

   Alternatively you can use released archives. You can download them from our website <https://weblate.org/>. Those downloads are cryptographically signed, please see *Weryfikowanie podpisów wersji*.

3. Install current Weblate code into the virtualenv:

   ```
   . ~/weblate-env/bin/activate
   pip install -e weblate-src
   ```

4. Skopiuj `weblate/settings_example.py` do `weblate/settings.py`.

5. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Dostosowywanie konfiguracji*.

6. Create the database used by Weblate, see *Konfiguracja bazy danych dla Weblate*.

7. Build Django tables, static files and initial data (see *Wypełnianie bazy danych* and *Obsługa plików statycznych*):

   ```
   weblate migrate
   weblate collectstatic
   weblate compress
   ```

   Informacja: This step should be repeated whenever you update the repository.

Instalacja za pomocą OpenShift

With the OpenShift Weblate template you can get your personal Weblate instance up and running in seconds. All of Weblate's dependencies are already included. PostgreSQL is set up as the default database and persistent volume claims are used.

You can find the template at <https://github.com/WeblateOrg/openshift/>.

Instalacja

The following examples assume you have a working OpenShift v3.x environment, with `oc` client tool installed. Please check the OpenShift documentation for instructions.

The `template.yml` is suited for running all components in OpenShift. There is also `template-external-postgresql.yml` which does not start a PostgreSQL server and allows you to configure external PostgreSQL server.

Konsola Web

Copy the raw content from template.yml and import them into your project, then use the `Create` button in the OpenShift web console to create your application. The web console will prompt you for the values for all of the parameters used by the template.

CLI

To upload the Weblate template to your current project's template library, pass the `template.yml` file with the following command:

```
$ oc create -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
   -n <PROJECT>
```

The template is now available for selection using the web console or the CLI.

Parametry

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

```
$ oc process --parameters -f https://raw.githubusercontent.com/WeblateOrg/
↪openshift/main/template.yml

# If the template is already uploaded
$ oc process --parameters -n <PROJECT> weblate
```

Obsługa administracyjna

You can also use the CLI to process templates and use the configuration that is generated to create objects immediately.

```
$ oc process -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
    -p APPLICATION_NAME=weblate \
    -p WEBLATE_VERSION=4.3.1-1 \
    -p WEBLATE_SITE_DOMAIN=weblate.app-openshift.example.com \
    -p POSTGRESQL_IMAGE=docker-registry.default.svc:5000/openshift/postgresql:9.6 \
    -p REDIS_IMAGE=docker-registry.default.svc:5000/openshift/redis:3.2 \
    | oc create -f
```

The Weblate instance should be available after successful migration and deployment at the specified `WEBLATE_SITE_DOMAIN` parameter.

After container setup, you can sign in as *admin* user with password provided in `WEBLATE_ADMIN_PASSWORD`, or a random password generated on first start if that was not set.

To reset *admin* password, restart the container with `WEBLATE_ADMIN_PASSWORD` set to new password in the respective `Secret`.

Eliminacja

```
$ oc delete all -l app=<APPLICATION_NAME>
$ oc delete configmap -l app= <APPLICATION_NAME>
$ oc delete secret -l app=<APPLICATION_NAME>
# ATTTENTION! The following command is only optional and will permanently delete␣
↪all of your data.
$ oc delete pvc -l app=<APPLICATION_NAME>

$ oc delete all -l app=weblate \
    && oc delete secret -l app=weblate \
    && oc delete configmap -l app=weblate \
    && oc delete pvc -l app=weblate
```

Konfiguracja

By processing the template a respective `ConfigMap` will be created and which can be used to customize the Weblate image. The `ConfigMap` is directly mounted as environment variables and triggers a new deployment every time it is changed. For further configuration options, see *Zmienne środowiskowe platformy Docker* for full list of environment variables.

Instalacja Kubernetes

Informacja: This guide is looking for contributors experienced with Kubernetes to cover the setup in more details.

With the Kubernetes Helm chart you can get your personal Weblate instance up and running in seconds. All of Weblate's dependencies are already included. PostgreSQL is set up as the default database and persistent volume claims are used.

You can find the chart at <https://github.com/WeblateOrg/helm/> and it can be displayed at <https://artifacthub.io/packages/helm/weblate/weblate>.

Instalacja

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

Konfiguracja

Aby uzyskać dalsze opcje konfiguracji, zobacz *Zmienne środowiskowe platformy Docker*, gdzie znajduje się pełna lista zmiennych środowiskowych.

Depending on your setup and experience, choose an appropriate installation method for you:

- *Instalowanie za pomocą Dockera*, recommended for production setups.
- Virtualenv installation, recommended for production setups:
  - *Instalacja na Debianie i Ubuntu*
  - *Instalacja na SUSE i openSUSE*
  - *Installing on RedHat, Fedora and CentOS*
  - *Instalacja w systemie macOS*
- *Instalacja ze źródeł*, recommended for development.
- *Instalacja za pomocą OpenShift*
- *Instalacja Kubernetes*

2.1.2 Wymagania dotyczące oprogramowania

System operacyjny

Weblate is known to work on Linux, FreeBSD and macOS. Other Unix like systems will most likely work too.

Weblate is not supported on Windows. But it may still work and patches are happily accepted.

Inne usługi

Weblate is using other services for its operation. You will need at least following services running:

- Serwer bazy danych PostgreSQL, zobacz *Konfiguracja bazy danych dla Weblate*.
- Redis server for cache and tasks queue, see *Zadania w tle korzystające z Celery*.
- SMTP server for outgoing e-mail, see *Konfigurowanie poczty wychodzącej*.

Zależności ythona

Weblate is written in Python and supports Python 3.6 or newer. You can install dependencies using pip or from your distribution packages, full list is available in `requirements.txt`.

Najbardziej znaczące zależności:

Django
https://www.djangoproject.com/

Celery
https://docs.celeryq.dev/

Translate Toolkit
https://toolkit.translatehouse.org/

translation-finder
https://github.com/WeblateOrg/translation-finder

Python Social Auth
https://python-social-auth.readthedocs.io/

Django REST Framework
https://www.django-rest-framework.org/

Opcjonalne zależności

Following modules are necessary for some Weblate features. You can find all of them in `requirements-optional.txt`.

`Mercurial` (opcjonalnie dla obsługi repozytoriów *Mercurial*)
https://www.mercurial-scm.org/

`phply` (opcjonalnie dla *Ciągi PHP*)
https://github.com/viraptor/phply

`tesserocr` (opcjonalnie dla OCR w *Kontekst wizualny dla ciągów*)
https://github.com/sirfz/tesserocr

`python-akismet` (opcjonalnie dla *Ochrona przed spamem*)
https://github.com/Nekmo/python-akismet

`ruamel.yaml` (opcjonalnie dla *Pliki YAML*)
https://pypi.org/project/ruamel.yaml/

`Zeep` (opcjonalnie dla *Microsoft Terminology*)
https://docs.python-zeep.org/

`aeidon` (opcjonalnie dla *Pliki napisów*)
https://pypi.org/project/aeidon/

`fluent.syntax` (opcjonalnie dla *Format Fluent*)
https://projectfluent.org/

Podpowiedź: When installing using pip, you can directly specify desired features when installing:

```
pip install "Weblate[PHP,Fluent]"
```

Lub możesz zainstalować Weblate ze wszystkimi opcjonalnymi funkcjami:

```
pip install "Weblate[all]"
```

Or you can install Weblate without any optional features:

```
pip install Weblate
```

Zależności backendu bazy danych

Weblate supports PostgreSQL, MySQL and MariaDB, see *Konfiguracja bazy danych dla Weblate* and backends documentation for more details.

Inne wymagania systemowe

The following dependencies have to be installed on the system:

`Git`
https://git-scm.com/

Pango, Cairo and related header files and GObject introspection data
https://cairographics.org/, https://pango.gnome.org/, patrz *Pango i Cairo*

`git-review` (opcjonalnie dla obsługi Gerrit)
https://pypi.org/project/git-review/

`git-svn` (opcjonalnie dla obsługi Subversion)
https://git-scm.com/docs/git-svn

`tesseract` and its data (optional for screenshots OCR)
https://github.com/tesseract-ocr/tesseract

`licensee` (optional for detecting license when creating component)
https://github.com/licensee/licensee

Zależność od czasu kompilacji

To build some of the *Zależności ythona* you might need to install their dependencies. This depends on how you install them, so please consult individual packages for documentation. You won't need those if using prebuilt `Wheels` while installing using `pip` or when you use distribution packages.

Pango i Cairo

Zmienione w wersji 3.7.

Weblate uses Pango and Cairo for rendering bitmap widgets (see promotion) and rendering checks (see *Zarządzanie czcionkami*). To properly install Python bindings for those you need to install system libraries first - you need both Cairo and Pango, which in turn need GLib. All those should be installed with development files and GObject introspection data.

2.1.3 Weryfikowanie podpisów wersji

Weblate release are cryptographically signed by the releasing developer. Currently this is Michal Čihař. Fingerprint of his PGP key is:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

and you can get more identification information from <https://keybase.io/nijel>.

You should verify that the signature matches the archive you have downloaded. This way you can be sure that you are using the same code that was released. You should also verify the date of the signature to make sure that you downloaded the latest version.

Each archive is accompanied with `.asc` files which contain the PGP signature for it. Once you have both of them in the same folder, you can verify the signature:

```
$ 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
```

As you can see GPG complains that it does not know the public key. At this point you should do one of the following steps:

- Use *wkd* to download the key:

```
$ 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 (Braiins) <michal.cihar@braiins.cz>
sub   rsa4096 2009-06-17 [E]
sub   rsa4096 2015-09-09 [S]
```

- Download the keyring from Michal's server, then import it with:

```
$ gpg --import wmxth3chu9jfxdxywj1skpmhsj311mzm
```

- Download and import the key from one of the key servers:

```
$ 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
```

This will improve the situation a bit - at this point you can verify that the signature from the given key is correct but you still can not trust the name used in the key:

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

The problem here is that anybody could issue the key with this name. You need to ensure that the key is actually owned by the mentioned person. The GNU Privacy Handbook covers this topic in the chapter Validating other keys on your public keyring. The most reliable method is to meet the developer in person and exchange key fingerprints, however you can also rely on the web of trust. This way you can trust the key transitively through signatures of others, who have met the developer in person.

Once the key is trusted, the warning will not occur:

```
$ 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]
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
gpg:                 aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:                 aka "[jpeg image of size 8848]" [ultimate]
gpg:                 aka "Michal Čihař (Braiins) <michal.cihar@braiins.cz>"␣
↪[ultimate]
```

Should the signature be invalid (the archive has been changed), you would get a clear error regardless of the fact that the key is trusted or not:

```
$ 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 Uprawnienia systemu plików

The Weblate process needs to be able to read and write to the directory where it keeps data - `DATA_DIR`. All files within this directory should be owned and writable by the user running all Weblate processes (typically WSGI and Celery, see *Uruchamianie serwera* and *Zadania w tle korzystające z Celery*).

The default configuration places them in the same tree as the Weblate sources, however you might prefer to move these to a better location such as: `/var/lib/weblate`.

Weblate tries to create these directories automatically, but it will fail when it does not have permissions to do so.

You should also take care when running *Polecenia zarządzania*, as they should be ran under the same user as Weblate itself is running, otherwise permissions on some files might be wrong.

In the Docker container, all files in the `/app/data` volume have to be owned by the `weblate` user inside the container (UID 1000).

Zobacz także:

Obsługa plików statycznych

2.1.5 Konfiguracja bazy danych dla Weblate

It is recommended to run Weblate with a PostgreSQL database server.

Zobacz także:

Użyj potężnego silnika bazy danych, Databases, *Migrating from other databases to PostgreSQL*

PostgreSQL

PostgreSQL is usually the best choice for Django-based sites. It's the reference database used for implementing Django database layer.

Informacja: Weblate uses trigram extension which has to be installed separately in some cases. Look for `postgresql-contrib` or a similarly named package.

Zobacz także:

PostgreSQL notes

Tworzenie bazy danych w PostgreSQL

It is usually a good idea to run Weblate in a separate database, and separate user account:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser --superuser --pwprompt weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Podpowiedź: If you don't want to make the Weblate user a superuser in PostgreSQL, you can omit that. In that case you will have to perform some of the migration steps manually as a PostgreSQL superuser in schema Weblate will use:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
```

Konfigurowanie Weblate do korzystania z PostgreSQL

The `settings.py` snippet for PostgreSQL:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    }
}
```

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

MySQL i MariaDB

Podpowiedź: Some Weblate features will perform better with *PostgreSQL*. This includes searching and translation memory, which both utilize full-text features in the database and PostgreSQL implementation is superior.

Weblate can be also used with MySQL or MariaDB, please see MySQL notes and MariaDB notes for caveats using Django with those. Because of the limitations it is recommended to use *PostgreSQL* for new installations.

Weblate requires MySQL at least 5.7.8 or MariaDB at least 10.2.7.

Following configuration is recommended for Weblate:

- Use the `utf8mb4` charset to allow representation of higher Unicode planes (for example emojis).
- Configure the server with `innodb_large_prefix` to allow longer indices on text fields.
- Set the isolation level to `READ COMMITTED`.
- The SQL mode should be set to `STRICT_TRANS_TABLES`.

MySQL 8.x, MariaDB 10.5.x or newer have reasonable default configuration so that no server tweaking should be necessary and all what is needed can be configured on the client side.

Below is an example `/etc/my.cnf.d/server.cnf` for a server with 8 GB of RAM. These settings should be sufficient for most installs. MySQL and MariaDB have tunables that will increase the performance of your server that are considered not necessary unless you are planning on having large numbers of concurrent users accessing the system. See the various vendors documentation on those details.

It is absolutely critical to reduce issues when installing that the setting `innodb_file_per_table` is set properly and MySQL/MariaDB restarted before you start your Weblate install.

```
[mysqld]
character-set-server = utf8mb4
character-set-client = utf8mb4
collation-server = utf8mb4_unicode_ci

datadir=/var/lib/mysql

log-error=/var/log/mariadb/mariadb.log

innodb_large_prefix=1
innodb_file_format=Barracuda
innodb_file_per_table=1
innodb_buffer_pool_size=2G
sql_mode=STRICT_TRANS_TABLES
```

Podpowiedź: In case you are getting `#1071 - Specified key was too long; max key length is 767 bytes` error, please update your configuration to include the `innodb` settings above and restart your install.

Podpowiedź: In case you are getting `#2006 - MySQL server has gone away` error, configuring `CONN_MAX_AGE` might help.

Konfigurowanie Weblate przy użyciu MySQL/MariaDB

The `settings.py` snippet for MySQL and MariaDB:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "127.0.0.1",
        # Set to empty string for default
        "PORT": "3306",
        # In case you wish to use additional
        # connection options
        "OPTIONS": {},
    }
}
```

You should also create the `weblate` user account in MySQL or MariaDB before you begin the install. Use the commands below to achieve that:

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

2.1.6 Inne konfiguracje

Konfigurowanie poczty wychodzącej

Weblate sends out e-mails on various occasions - for account activation and on various notifications configured by users. For this it needs access to an SMTP server.

The mail server setup is configured using these settings: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_TLS`, `EMAIL_USE_SSL`, `EMAIL_HOST_USER` and `EMAIL_PORT`. Their names are quite self-explanatory, but you can find more info in the Django documentation.

Podpowiedź: In case you get error about not supported authentication (for example `SMTP AUTH extension not supported by server`), it is most likely caused by using insecure connection and server refuses to authenticate this way. Try enabling `EMAIL_USE_TLS` in such case.

Zobacz także:

Nie otrzymuję e-maili od Weblate, *Configuring outgoing e-mail in Docker container*

Działanie za odwrotnym proxy

Several features in Weblate rely on being able to get client IP address. This includes *Ograniczenie szybkości*, *Ochrona przed spamem* or *Dziennik aktywności*.

In default configuration Weblate parses IP address from `REMOTE_ADDR` which is set by the WSGI handler.

In case you are running a reverse proxy, this field will most likely contain its address. You need to configure Weblate to trust additional HTTP headers and parse the IP address from these. This can not be enabled by default as it would allow IP address spoofing for installations not using a reverse proxy. Enabling `IP_BEHIND_REVERSE_PROXY` might be enough for the most usual setups, but you might need to adjust `IP_PROXY_HEADER` and `IP_PROXY_OFFSET` as well.

Another thing to take care of is the Host header. It should match to whatever is configured as `SITE_DOMAIN`. Additional configuration might be needed in your reverse proxy (for example use `ProxyPreserveHost On` for Apache or `proxy_set_header Host $host;` with nginx).

Zobacz także:

Ochrona przed spamem, *Ograniczenie szybkości*, *Dziennik aktywności*, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`, `SECURE_PROXY_SSL_HEADER`

HTTP proxy

Weblate does execute VCS commands and those accept proxy configuration from environment. The recommended approach is to define proxy settings in `settings.py`:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Zobacz także:

Zmienne środowiska proxy

2.1.7 Dostosowywanie konfiguracji

Zobacz także:

Przykładowa konfiguracja

Copy `weblate/settings_example.py` to `weblate/settings.py` and adjust it to match your setup. You will probably want to adjust the following options: `ADMINS`

List of site administrators to receive notifications when something goes wrong, for example notifications on failed merges, or Django errors.

Zobacz także:

`ADMINS`, *Prawidłowe konfigurowanie administratorów*

`ALLOWED_HOSTS`

You need to set this to list the hosts your site is supposed to serve. For example:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Alternatywnie można dołączyć symbol wieloznaczny:

```
ALLOWED_HOSTS = ["*"]
```

Zobacz także:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Konfiguracja dozwolonych hostów*

SESSION_ENGINE

Configure how your sessions will be stored. In case you keep the default database backend engine, you should schedule: **weblate clearsessions** to remove stale session data from the database.

If you are using Redis as cache (see *Włączanie buforowania*) it is recommended to use it for sessions as well:

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

Zobacz także:

Configuring the session engine, SESSION_ENGINE

DATABASES

Connectivity to database server, please check Django's documentation for more details.

Zobacz także:

Konfiguracja bazy danych dla Weblate, DATABASES, Databases

DEBUG

Disable this for any production server. With debug mode enabled, Django will show backtraces in case of error to users, when you disable it, errors will be sent per e-mail to ADMINS (see above).

Debug mode also slows down Weblate, as Django stores much more info internally in this case.

Zobacz także:

DEBUG, *Wyłączenie trybu debugowania*

DEFAULT_FROM_EMAIL

E-mail sender address for outgoing e-mail, for example registration e-mails.

Zobacz także:

DEFAULT_FROM_EMAIL

SECRET_KEY

Key used by Django to sign some info in cookies, see *Tajny klucz Django* for more info.

Zobacz także:

SECRET_KEY

SERVER_EMAIL

E-mail used as sender address for sending e-mails to the administrator, for example notifications on failed merges.

Zobacz także:

SERVER_EMAIL

2.1.8 Wypełnianie bazy danych

After your configuration is ready, you can run `weblate migrate` to create the database structure. Now you should be able to create translation projects using the admin interface.

In case you want to run an installation non interactively, you can use `weblate migrate --noinput`, and then create an admin user using `createadmin` command.

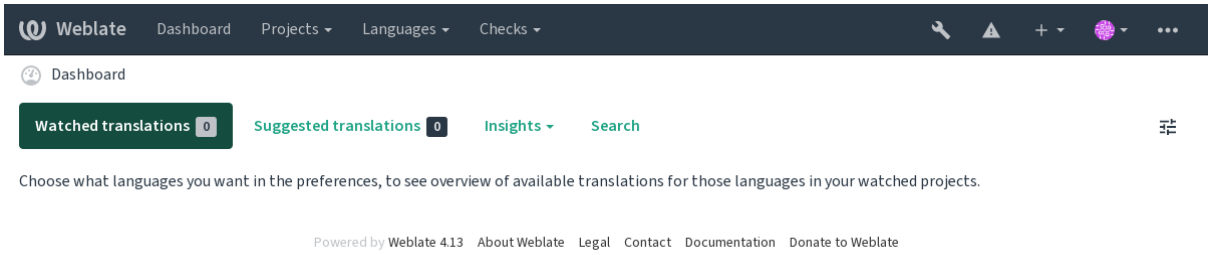
Once you are done, you should also check the *Performance report* in the admin interface, which will give you hints of potential non optimal configuration on your site.

Zobacz także:

Konfiguracja, *Lista uprawnień i wbudowanych ról*

2.1.9 Instalacja produkcyjna

For a production setup you should carry out adjustments described in the following sections. The most critical settings will trigger a warning, which is indicated by an exclamation mark in the top bar if signed in as a superuser:



It is also recommended to inspect checks triggered by Django (though you might not need to fix all of them):

```
weblate check --deploy
```

You can also review the very same checklist from the *Interfejs zarządzania*.

Zobacz także:

Wdrożeniowa lista kontrolna

Wyłączenie trybu debugowania

Disable Django's debug mode (`DEBUG`) by:

```
DEBUG = False
```

With debug mode on, Django stores all executed queries and shows users backtraces of errors, which is not desired in a production setup.

Zobacz także:

Dostosowywanie konfiguracji

Prawidłowe konfigurowanie administratorów

Set the correct admin addresses to the `ADMINS` setting to defining who will receive e-mails in case something goes wrong on the server, for example:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Zobacz także:

Dostosowywanie konfiguracji

Ustaw prawidłową domenę strony

Adjust site name and domain in the admin interface, otherwise links in RSS or registration e-mails will not work. This is configured using `SITE_DOMAIN` which should contain site domain name.

Zmienione w wersji 4.2: Prior to the 4.2 release the Django sites framework was used instead, please see The „sites" framework.

Zobacz także:

Konfiguracja dozwolonych hostów, *Prawidłowa konfiguracja HTTPS* `SITE_DOMAIN`, `WEBLATE_SITE_DOMAIN`, `ENABLE_HTTPS`

Prawidłowa konfiguracja HTTPS

It is strongly recommended to run Weblate using the encrypted HTTPS protocol. After enabling it, you should set `ENABLE_HTTPS` in the settings:

```
ENABLE_HTTPS = True
```

Podpowiedź: You might want to set up HSTS as well, see SSL/HTTPS for more details.

Zobacz także:

`ENABLE_HTTPS`, *Konfiguracja dozwolonych hostów*, *Ustaw prawidłową domenę strony*

Ustaw prawidłowo SECURE_HSTS_SECONDS

If your site is served over SSL, you have to consider setting a value for `SECURE_HSTS_SECONDS` in the `settings.py` to enable HTTP Strict Transport Security. By default it's set to 0 as shown below.

```
SECURE_HSTS_SECONDS = 0
```

If set to a non-zero integer value, the `django.middleware.security.SecurityMiddleware` sets the HTTP Strict Transport Security header on all responses that do not already have it.

Ostrzeżenie: Setting this incorrectly can irreversibly (for some time) break your site. Read the HTTP Strict Transport Security documentation first.

Użyj potężnego silnika bazy danych

- Please use PostgreSQL for a production environment, see *Konfiguracja bazy danych dla Weblate* for more info.
- Use adjacent location for running the database server, otherwise the networking performance or reliability might ruin your Weblate experience.
- Check the database server performance or tweak its configuration, for example using PGTune.

Zobacz także:

Konfiguracja bazy danych dla Weblate, *Migrating from other databases to PostgreSQL*, *Dostosowywanie konfiguracji*, Databases

Włączanie buforowania

If possible, use Redis from Django by adjusting the `CACHES` configuration variable, for example:

```
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",
        },
    }
}
```

Podpowiedź: In case you change Redis settings for the cache, you might need to adjust them for Celery as well, see *Zadania w tle korzystające z Celery*.

Zobacz także:

Pamięć podręczna awatarów, Django's cache framework

Pamięć podręczna awatarów

In addition to caching of Django, Weblate performs caching of avatars. It is recommended to use a separate, file-backed cache for this purpose:

```
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,
        },
    },
}
```

Zobacz także:

`ENABLE_AVATARS`, `AVATAR_URL_PREFIX`, *Awatary*, *Włączanie buforowania*, Django's cache framework

Konfigurowanie wysyłania wiadomości e-mail

Weblate needs to send out e-mails on several occasions, and these e-mails should have a correct sender address, please configure `SERVER_EMAIL` and `DEFAULT_FROM_EMAIL` to match your environment, for example:

```
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"
```

Informacja: To disable sending e-mails by Weblate set `EMAIL_BACKEND` to `django.core.mail.backends.dummy.EmailBackend`.

This will disable *all* e-mail delivery including registration or password reset e-mails.

Zobacz także:

Dostosowywanie konfiguracji, *Konfigurowanie poczty wychodzącej*, `EMAIL_BACKEND`, `DEFAULT_FROM_EMAIL`, `SERVER_EMAIL`

Konfiguracja dozwolonych hostów

Django requires *ALLOWED_HOSTS* to hold a list of domain names your site is allowed to serve, leaving it empty will block any requests.

In case this is not configured to match your HTTP server, you will get errors like `Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS.`

Podpowiedź: On Docker container, this is available as *WEBLATE_ALLOWED_HOSTS*.

Zobacz także:

ALLOWED_HOSTS, *WEBLATE_ALLOWED_HOSTS*, *Ustaw prawidłową domenę strony*

Tajny klucz Django

The *SECRET_KEY* setting is used by Django to sign cookies, and you should really generate your own value rather than using the one from the example setup.

You can generate a new key using `weblate/examples/generate-secret-key` shipped with Weblate.

Zobacz także:

SECRET_KEY

Katalog domowy

Zmienione w wersji 2.1: This is no longer required, Weblate now stores all its data in *DATA_DIR*.

The home directory for the user running Weblate should exist and be writable by this user. This is especially needed if you want to use SSH to access private repositories, but Git might need to access this directory as well (depending on the Git version you use).

You can change the directory used by Weblate in `settings.py`, for example to set it to `configuration` directory under the Weblate tree:

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

Informacja: On Linux, and other UNIX like systems, the path to user's home directory is defined in `/etc/passwd`. Many distributions default to a non-writable directory for users used for serving web content (such as `apache`, `www-data` or `wwwrun`), so you either have to run Weblate under a different user, or change this setting.

Zobacz także:

Dostęp do repozytoriów

Ładowanie szablonu

It is recommended to use a cached template loader for Django. It caches parsed templates and avoids the need to do parsing with every single request. You can configure it using the following snippet (the `loaders` setting is important here):

```
TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [
            os.path.join(BASE_DIR, "templates"),
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
        ],
        "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",
                    ],
                ),
            ],
        },
    },
]
```

Zobacz także:

`django.template.loaders.cached.Loader`

Uruchamianie zadań konserwacyjnych

For optimal performance, it is good idea to run some maintenance tasks in the background. This is now automatically done by *Zadania w tle korzystające z Celery* and covers following tasks:

- Sprawdzanie kondycji konfiguracji (co godzinę).
- Committing pending changes (hourly), see *Leniwe zatwierdzenia* and `commit_pending`.
- Aktualizacja alertów komponentów (codziennie).
- Update remote branches (nightly), see `AUTO_UPDATE`.
- Translation memory backup to JSON (daily), see `dump_memory`.
- Fulltext and database maintenance tasks (daily and weekly tasks), see `cleanuptrans`.

Zmienione w wersji 3.2: Since version 3.2, the default way of executing these tasks is using Celery and Weblate already comes with proper configuration, see *Zadania w tle korzystające z Celery*.

System locales and encoding

The system locales should be configured to UTF-8 capable ones. On most Linux distributions this is the default setting. In case it is not the case on your system, please change locales to UTF-8 variant.

For example by editing `/etc/default/locale` and setting there `LANG="C.UTF-8"`.

In some cases the individual services have separate configuration for locales. This varies between distribution and web servers, so check documentation of your web server packages for that.

Apache on Ubuntu uses `/etc/apache2/envvars`:

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

Apache on CentOS uses `/etc/sysconfig/httpd` (or `/opt/rh/httpd24/root/etc/sysconfig/httpd`):

```
LANG='en_US.UTF-8'
```

Korzystanie z niestandardowego urzędu certyfikacji

Weblate does verify SSL certificates during HTTP requests. In case you are using custom certificate authority which is not trusted in default bundles, you will have to add its certificate as trusted.

The preferred approach is to do this at system level, please check your distro documentation for more details (for example on debian this can be done by placing the CA certificate into `/usr/local/share/ca-certificates/` and running **`update-ca-certificates`**).

Once this is done, system tools will trust the certificate and this includes Git.

For Python code, you will need to configure requests to use system CA bundle instead of the one shipped with it. This can be achieved by placing following snippet to `settings.py` (the path is Debian specific):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Kompresowanie zasobów klienta

Weblate comes with a bunch of JavaScript and CSS files. For performance reasons it is good to compress them before sending to a client. In default configuration this is done on the fly at cost of little overhead. On big installations, it is recommended to enable offline compression mode. This needs to be done in the configuration and the compression has to be triggered on every Weblate upgrade.

The configuration switch is simple by enabling `django.conf.settings.COMPRESS_OFFLINE` and configuring `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` (the latter is already included in the example configuration):

```
COMPRESS_OFFLINE = True
```

On each deploy you need to compress the files to match current version:

```
weblate compress
```

Podpowiedź: The official Docker image has this feature already enabled.

Zobacz także:

Common Deployment Scenarios, *Obsługa plików statycznych*

2.1.10 Uruchamianie serwera

Podpowiedź: In case you are not experienced with services described below, you might want to try *Instalowanie za pomocą Dockera*.

You will need several services to run Weblate, the recommended setup consists of:

- Serwer bazy danych (patrz *Konfiguracja bazy danych dla Weblate*)
- Serwer pamięci podręcznej (patrz *Włączanie buforowania*)

- Frontend web server for static files and SSL termination (see *Obsługa plików statycznych*)
- WSGI server for dynamic content (see *Sample configuration for NGINX and uWSGI*)
- Celery for executing background tasks (see *Zadania w tle korzystające z Celery*)

Informacja: There are some dependencies between the services, for example cache and database should be running when starting up Celery or uwsgi processes.

In most cases, you will run all services on single (virtual) server, but in case your installation is heavy loaded, you can split up the services. The only limitation on this is that Celery and Wsgi servers need access to `DATA_DIR`.

Informacja: The WSGI process has to be executed under the same user the Celery process, otherwise files in the `DATA_DIR` will be stored with mixed ownership, leading to runtime issues.

See also *Uprawnienia systemu plików* and *Zadania w tle korzystające z Celery*.

Uruchamianie serwer WWW

Running Weblate is not different from running any other Django based program. Django is usually executed as uWSGI or fcgi (see examples for different webservers below).

For testing purposes, you can use the built-in web server in Django:

```
weblate runserver
```

Ostrzeżenie: DO NOT USE THIS SERVER IN A PRODUCTION SETTING. It has not gone through security audits or performance tests. See also Django documentation on `runserver`.

Podpowiedź: The Django built-in server serves static files only with `DEBUG` enabled as it is intended for development only. For production use, please see wsgi setups in *Sample configuration for NGINX and uWSGI*, *Przykładowa konfiguracja dla Apache*, *Sample configuration for Apache and Gunicorn*, and *Obsługa plików statycznych*.

Obsługa plików statycznych

Zmienione w wersji 2.4: Prior to version 2.4, Weblate didn't properly use the Django static files framework and the setup was more complex.

Django needs to collect its static files in a single directory. To do so, execute `weblate collectstatic --noinput`. This will copy the static files into a directory specified by the `STATIC_ROOT` setting (this defaults to a `static` directory inside `DATA_DIR`).

It is recommended to serve static files directly from your web server, you should use that for the following paths:

`/static/`
Serves static files for Weblate and the admin interface (from defined by `STATIC_ROOT`).

`/media/`
Used for user media uploads (e.g. screenshots).

`/favicon.ico`
Should be rewritten to rewrite a rule to serve `/static/favicon.ico`.

Zobacz także:

Sample configuration for NGINX and uWSGI, *Przykładowa konfiguracja dla Apache*, *Sample configuration for Apache and Gunicorn*, *Kompresowanie zasobów klienta*, Jak wdrażać Django, How to deploy static files

Polityka bezpieczeństwa treści

The default Weblate configuration enables `weblate.middleware.SecurityMiddleware` middleware which sets security related HTTP headers like `Content-Security-Policy` or `X-XSS-Protection`. These are by default set up to work with Weblate and its configuration, but this might need customization for your environment.

Zobacz także:

CSP_SCRIPT_SRC, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

Sample configuration for NGINX and uWSGI

To run production webserver, use the wsgi wrapper installed with Weblate (in virtual env case it is installed as `~/weblate-env/lib/python3.7/site-packages/weblate/wsgi.py`). Don't forget to set the Python search path to your virtualenv as well (for example using `virtualenv = /home/user/weblate-env` in uWSGI).

The following configuration runs Weblate as uWSGI under the NGINX webserver.

Configuration for NGINX (also available as `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 {
    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;
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
        # uwsgi_pass 127.0.0.1:8080;
    }
}
```

Configuration for uWSGI (also available as `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/python3.9/site-packages/weblate/wsgi.
↪py

# Add path to Weblate checkout if you did not install
# Weblate by pip
# python-path   = /path/to/weblate

# In case you're using virtualenv uncomment this:
virtualenv = /home/weblate/weblate-env

# Needed for OAuth/OpenID
buffer-size   = 8192

# Reload when consuming too much of memory
reload-on-rss = 250

# Increase number of workers for heavily loaded sites
workers       = 8

# Enable threads for Sentry error submission
enable-threads = true

# Child processes do not need file descriptors
close-on-exec = true

# Avoid default 0000 umask
umask = 0022

# Run as weblate user
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
```

Zobacz także:

Jak używać Django z uWSGI

Przykładowa konfiguracja dla Apache

It is recommended to use prefork MPM when using WSGI with Weblate.

The following configuration runs Weblate as WSGI, you need to have enabled `mod_wsgi` (available as `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/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate␣
↪request-timeout=600
    WSGIProcessGroup weblate
    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>
```

Informacja: Weblate requires Python 3, so please make sure you are running Python 3 variant of the modwsgi. Usually it is available as a separate package, for example `libapache2-mod-wsgi-py3`.

Zobacz także:

System locales and encoding, Jak używać Django z Apache i mod_wsgi

Sample configuration for Apache and Gunicorn

The following configuration runs Weblate in Gunicorn and Apache 2.4 (available as `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/>
        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>
```

Zobacz także:

Jak używać Django z Gunicornem

Uruchamianie Weblate pod ścieżką

Nowe w wersji 1.3.

It is recommended to use prefork MPM when using WSGI with Weblate.

A sample Apache configuration to serve Weblate under `/weblate`. Again using `mod_wsgi` (also available as `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 user=weblate␣
↪request-timeout=600
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

    WSGIScriptAlias /weblate /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>
```

Additionally, you will have to adjust `weblate/settings.py`:

```
URL_PREFIX = "/weblate"
```

2.1.11 Zadania w tle korzystające z Celery

Nowe w wersji 3.2.

Weblate uses Celery to execute regular and background tasks. You are supposed to run a Celery service that will execute these. For example, it is responsible for handling following operations (this list is not complete):

- Receiving webhooks from external services (see *Hooki powiadomień*).
- Running regular maintenance tasks such as backups, cleanups, daily add-ons, or updates (see *Tworzenie kopii zapasowych i przenoszenie weblate*, `BACKGROUND_TASKS`, *Dodatki*).
- Uruchamianie *Tłumaczenie automatyczne*.
- Wysyłanie powiadomień podsumowujących.
- Offloading expensive operations from the wsgi process.
- Commitowanie oczekujących zmian (patrz *Leniwe zatwierdzenia*).

A typical setup using Redis as a backend looks like this:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

Zobacz także:

Redis broker configuration in Celery

You should also start the Celery worker to process the tasks and start scheduled tasks, this can be done directly on the command-line (which is mostly useful when debugging or developing):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Informacja: The Celery process has to be executed under the same user as the WSGI process, otherwise files in the `DATA_DIR` will be stored with mixed ownership, leading to runtime issues.

See also *Uprawnienia systemu plików* and *Uruchamianie serwera*.

Executing Celery tasks in the wsgi using eager mode

Informacja: This will have severe performance impact on the web interface, and will break features depending on regular trigger (for example committing pending changes, digest notifications, or backups).

For development, you might want to use eager configuration, which does process all tasks in place:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

Uruchamianie Celery jako usługi systemowej

Most likely you will want to run Celery as a daemon and that is covered by Daemonization. For the most common Linux setup using systemd, you can use the example files shipped in the `examples` folder listed below.

Systemd unit to be placed as `/etc/systemd/system/celery-weblate.service`:

```
[Unit]
Description=Celery Service (Weblate)
After=network.target

[Service]
Type=forking
User=weblate
Group=weblate
EnvironmentFile=/etc/default/celery-weblate
WorkingDirectory=/home/weblate
RuntimeDirectory=celery
RuntimeDirectoryPreserve=restart
LogsDirectory=celery
ExecStart=/bin/sh -c '${CELERY_BIN} multi start ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'
ExecStop=/bin/sh -c '${CELERY_BIN} multi stopwait ${CELERYD_NODES} \
  --pidfile=${CELERYD_PID_FILE}'
ExecReload=/bin/sh -c '${CELERY_BIN} multi restart ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'

[Install]
WantedBy=multi-user.target
```

Environment configuration to be placed as `/etc/default/celery-weblate`:

```
# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
CELERY_BIN="/home/weblate/weblate-env/bin/celery"

# App instance to use
# comment out this line if you don't use an app
CELERY_APP="weblate.utils"

# Extra command-line arguments to the worker,
# increase concurrency if you get weblate.E019
CELERYD_OPTS="--beat:celery --queues:celery=celery --prefetch-multiplier:celery=4 \
    --queues:notify=notify --prefetch-multiplier:notify=10 \
    --queues:memory=memory --prefetch-multiplier:memory=10 \
    --queues:translate=translate --prefetch-multiplier:translate=4 \
    --concurrency:backup=1 --queues:backup=backup  --prefetch-multiplier:backup=2"

# Logging configuration
# - %n will be replaced with the first part of the nodename.
# - %I will be replaced with the current child process index
#   and is important when using the prefork pool to avoid race conditions.
CELERYD_PID_FILE="/run/celery/weblate-%n.pid"
CELERYD_LOG_FILE="/var/log/celery/weblate-%n%I.log"
CELERYD_LOG_LEVEL="INFO"
```

Additional configuration to rotate Celery logs using **`logrotate`** to be placed as `/etc/logrotate.d/celery`:

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

Periodic tasks using Celery beat

Weblate comes with built-in setup for scheduled tasks. You can however define additional tasks in `settings.py`, for example see *Leniwe zatwierdzenia*.

The tasks are supposed to be executed by Celery beats daemon. In case it is not working properly, it might not be running or its database was corrupted. Check the Celery startup logs in such case to figure out root cause.

Monitorowanie stanu Celery

You can find current length of the Celery task queues in the *Interfejs zarządzania* or you can use `celery_queues` on the command-line. In case the queue will get too long, you will also get configuration error in the admin interface.

> **Ostrzeżenie:** The Celery errors are by default only logged into Celery log and are not visible to user. In case you want to have overview on such failures, it is recommended to configure *Zbieranie raportów o błędach*.

Zobacz także:

Monitorowanie eblate, *How can I check whether my Weblate is set up properly?*, Configuration and defaults, Workers Guide, Daemonization, Monitoring and Management Guide, `celery_queues`

2.1.12 Monitorowanie eblate

Weblate provides the `/healthz/` URL to be used in simple health checks, for example using Kubernetes. The Docker container has built-in health check using this URL.

For monitoring metrics of Weblate you can use `GET /api/metrics/` API endpoint.

Zobacz także:

How can I check whether my Weblate is set up properly?, *Monitorowanie stanu Celery*, Wtyczka Weblate dla Munina

2.1.13 Zbieranie raportów o błędach

Weblate, as any other software, can fail. In order to collect useful failure states we recommend to use third party services to collect such information. This is especially useful in case of failing Celery tasks, which would otherwise only report error to the logs and you won't get notified on them. Weblate has support for the following services:

Sentry

Weblate has built-in support for Sentry. To use it, it's enough to set *SENTRY_DSN* in the `settings.py`:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

Weblate has built-in support for Rollbar. To use it, it's enough to follow instructions for Rollbar notifier for Python.

In short, you need to adjust `settings.py`:

```
# Add rollbar as last middleware:
MIDDLEWARE = [
    # … other middleware classes …
    "rollbar.contrib.django.middleware.RollbarNotifierMiddleware",
]

# Configure client access
ROLLBAR = {
    "access_token": "POST_SERVER_ITEM_ACCESS_TOKEN",
    "client_token": "POST_CLIENT_ITEM_ACCESS_TOKEN",
    "environment": "development" if DEBUG else "production",
    "branch": "main",
    "root": "/absolute/path/to/code/root",
}
```

Everything else is integrated automatically, you will now collect both server and client side errors.

2.1.14 Migracja Weblate na inny serwer

Migrating Weblate to another server should be pretty easy, however it stores data in few locations which you should migrate carefully. The best approach is to stop Weblate for the migration.

Migrowanie bazy danych

Depending on your database backend, you might have several options to migrate the database. The most straightforward one is to dump the database on one server and import it on the new one. Alternatively you can use replication in case your database supports it.

The best approach is to use database native tools, as they are usually the most effective (e.g. **mysqldump** or **pg_dump**). If you want to migrate between different databases, the only option might be to use Django management to dump and import the database:

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

Migrowanie repozytoriów VCS

The VCS repositories stored under `DATA_DIR` need to be migrated as well. You can simply copy them or use **`rsync`** to do the migration more effectively.

Inne uwagi

Don't forget to move other services Weblate might have been using like Redis, Cron jobs or custom authentication backends.

2.2 Wdrożenia Weblate

Weblate can be easily installed in your cloud. Please find detailed guide for your platform:

- *Instalowanie za pomocą Dockera*
- *Instalacja za pomocą OpenShift*
- *Instalacja Kubernetes*

2.2.1 Third-party deployments for Weblate

Informacja: Following deployments are not developed or supported by Weblate team. Parts of the setup might vary from what is described in this documentation.

Stos Bitnami Weblate

Bitnami provides a Weblate stack for many platforms at <https://bitnami.com/stack/weblate>. The setup will be adjusted during installation, see <https://bitnami.com/stack/weblate/README.txt> for more documentation.

Pakiet Weblate Cloudron

Cloudron is a platform for self-hosting web applications. Weblate installed with Cloudron will be automatically kept up-to-date. The package is maintained by the Cloudron team at their Weblate package repo.

Cloudron install

Weblate w YunoHost

The self-hosting project YunoHost provides a package for Weblate. Once you have your YunoHost installation, you may install Weblate as any other application. It will provide you with a fully working stack with backup and restoration, but you may still have to edit your settings file for specific usages.

You may use your administration interface, or this button (it will bring you to your server):

Install with YUNOHOST

It also is possible to use the command-line interface:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Aktualizacja Weblate

2.3.1 Uaktualnienia obrazów platformy Docker

The official Docker image (see *Instalowanie za pomocą Dockera*) has all Weblate upgrade steps integrated. There are typically no manual steps needed besides pulling latest version.

Zobacz także:

Uaktualnianie kontenera platformy Docker

2.3.2 Ogólne instrukcje dotyczące aktualizacji

Before upgrading, please check the current *Wymagania dotyczące oprogramowania* as they might have changed. Once all requirements are installed or updated, please adjust your `settings.py` to match changes in the configuration (consult `settings_example.py` for correct values).

Always check *Instrukcje dotyczące poszczególnych wersji* before upgrade. In case you are skipping some versions, please follow instructions for all versions you are skipping in the upgrade. Sometimes it's better to upgrade to some intermediate version to ensure a smooth migration. Upgrading across multiple releases should work, but is not as well tested as single version upgrades.

Informacja: It is recommended to perform a full database backup prior to upgrade so that you can roll back the database in case upgrade fails, see *Tworzenie kopii zapasowych i przenoszenie weblate*.

1. Stop wsgi and Celery processes. The upgrade can perform incompatible changes in the database, so it is always safer to avoid old processes running while upgrading.
2. Aktualizacja kodu Weblate.

   For pip installs it can be achieved by:

   ```
   pip install -U "Weblate[all]"
   ```

   If you don't want to install all of the optional dependencies do:

   ```
   pip install -U Weblate
   ```

   With Git checkout you need to fetch new source code and update your installation:

   ```
   cd weblate-src
   git pull
   # Update Weblate inside your virtualenv
   . ~/weblate-env/bin/pip install -e .
   # Install dependencies directly when not using virtualenv
   pip install --upgrade -r requirements.txt
   # Install optional dependencies directly when not using virtualenv
   pip install --upgrade -r requirements-optional.txt
   ```
3. New Weblate release might have new *Opcjonalne zależności*, please check if they cover features you want.
4. Upgrade configuration file, refer to `settings_example.py` or *Instrukcje dotyczące poszczególnych wersji* for needed steps.
5. Aktualizacja struktury bazy danych:

```
weblate migrate --noinput
```

6. Collect updated static files (see *Uruchamianie serwera* and *Obsługa plików statycznych*):

```
weblate collectstatic --noinput --clear
```

7. Compress JavaScript and CSS files (optional, see *Kompresowanie zasobów klienta*):

```
weblate compress
```

8. If you are running version from Git, you should also regenerate locale files every time you are upgrading. You can do this by invoking:

```
weblate compilemessages
```

9. Verify that your setup is sane (see also *Instalacja produkcyjna*):

```
weblate check --deploy
```

10. Uruchom ponownie workera Celery (patrz *Zadania w tle korzystające z Celery*).

2.3.3 Instrukcje dotyczące poszczególnych wersji

Aktualizacja z 2.x

If you are upgrading from 2.x release, always first upgrade to 3.0.1 and then continue upgrading in the 3.x series. Upgrades skipping this step are not supported and will break.

Zobacz także:

Upgrade from 2.20 to 3.0 in Weblate 3.0 documentation

Aktualizacja z 3.x

If you are upgrading from 3.x release, always first upgrade to 4.0.4 or 4.1.1 and then continue upgrading in the 4.x series. Upgrades skipping this step are not supported and will break.

Zobacz także:

Upgrade from 3.11 to 4.0 in Weblate 4.0 documentation

Aktualizacja z 4.0 do 4.1

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

Istotne zmiany konfiguracji lub zależności:

- There are several changes in `settings_example.py`, most notable middleware changes, please adjust your settings accordingly.
- There are new file formats, you might want to include them in case you modified the `WEBLATE_FORMATS`.
- There are new quality checks, you might want to include them in case you modified the `CHECK_LIST`.
- There is change in `DEFAULT_THROTTLE_CLASSES` setting to allow reporting of rate limiting in the API.
- There are some new and updated requirements.
- There is a change in `INSTALLED_APPS`.

- The `MT_DEEPL_API_VERSION` setting has been removed in Version 4.7. The *DeepL* machine translation now uses the new `MT_DEEPL_API_URL` instead. You might need to adjust `MT_DEEPL_API_URL` to match your subscription.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Aktualizacja z 4.1 do 4.2

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

Istotne zmiany konfiguracji lub zależności:

- Upgrade from 3.x releases is not longer supported, please upgrade to 4.0 or 4.1 first.
- There are some new and updated requirements.
- There are several changes in `settings_example.py`, most notable new middleware and changed application ordering.
- The keys for JSON based formats no longer include leading dot. The strings are adjusted during the database migration, but external components might need adjustment in case you rely on keys in exports or API.
- The Celery configuration was changed to no longer use `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.
- The Weblate domain is now configured in the settings, see *`SITE_DOMAIN`* (or *`WEBLATE_SITE_DOMAIN`*). You will have to configure it before running Weblate.
- The username and email fields on user database now should be case insensitive unique. It was mistakenly not enforced with PostgreSQL.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Upgrade from 4.2 to 4.3

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

Istotne zmiany konfiguracji lub zależności:

- There are some changes in quality checks, you might want to include them in case you modified the *`CHECK_LIST`*.
- The source language attribute was moved from project to a component what is exposed in the API. You will need to update *Klient Weblate* in case you are using it.
- The database migration to 4.3 might take long depending on number of strings you are translating (expect around one hour of migration time per 100,000 source strings).
- There is a change in `INSTALLED_APPS`.
- There is a new setting *`SESSION_COOKIE_AGE_AUTHENTICATED`* which complements `SESSION_COOKIE_AGE`.
- In case you were using **`hub`** or **`lab`** to integrate with GitHub or GitLab, you will need to reconfigure this, see *`GITHUB_CREDENTIALS`* and *`GITLAB_CREDENTIALS`*.

Zmienione w wersji 4.3.1:

- The Celery configuration was changed to add `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.

Zmienione w wersji 4.3.2:

- The `post_update` method of add-ons now takes extra `skip_push` parameter.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Upgrade from 4.3 to 4.4

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

Istotne zmiany konfiguracji lub zależności:

- There is a change in `INSTALLED_APPS`, `weblate.configuration` has to be added there.
- Django 3.1 is now required.
- In case you are using MySQL or MariaDB, the minimal required versions have increased, see *MySQL i MariaDB*.

Zmienione w wersji 4.4.1:

- *Jednojęzyczny gettext* now uses both `msgid` and `msgctxt` when present. This will change identification of translation strings in such files breaking links to Weblate extended data such as screenshots or review states. Please make sure you commit pending changes in such files prior upgrading and it is recommended to force loading of affected component using `loadpo`.
- Increased minimal required version of translate-toolkit to address several file format issues.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Uaktualnienie z 4.4 do 4.5

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

Istotne zmiany konfiguracji lub zależności:

- The migration might take considerable time if you had big glossaries.
- Glossaries are now stored as regular components.
- The glossary API is removed, use regular translation API to access glossaries.
- There is a change in `INSTALLED_APPS` - `weblate.metrics` should be added.

Zmienione w wersji 4.5.1:

- There is a new dependency on the *pyahocorasick* module.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Uaktualnienie z 4.5 do 4.6

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

Istotne zmiany konfiguracji lub zależności:

- There are new file formats, you might want to include them in case you modified the `WEBLATE_FORMATS`.
- Interfejs API do tworzenia komponentów teraz automatycznie używa :ref:»internal-urls», patrz `POST /api/projects/(string:project)/components/`.
- There is a change in dependencies and `PASSWORD_HASHERS` to prefer Argon2 for passwords hashing.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Uaktualnienie z 4.6 do 4.7

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

Istotne zmiany konfiguracji lub zależności:

- There are several changes in `settings_example.py`, most notable middleware changes (`MIDDLEWARE`), please adjust your settings accordingly.
- The *DeepL* machine translation now has a generic `MT_DEEPL_API_URL` setting to adapt to different subscription models more flexibly. The `MT_DEEPL_API_VERSION` setting is no longer used.
- Django 3.2 is now required.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Uaktualnienie z 4.7 do 4.8

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

W tej wersji nie są wymagane żadne dodatkowe kroki uaktualniania.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Uaktualnienie z 4.8 do 4.9

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

- There is a change in storing metrics, the upgrade can take long time on larger sites.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Uaktualnienie z 4.9 do 4.10

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

- There is a change in per-project groups, the upgrade can take long time on sites with thousands of projects.
- Django 4.0 has made some incompatible changes, see Backwards incompatible changes in 4.0. Weblate still supports Django 3.2 for now, in case any of these are problematic. Most notable changes which might affect Weblate:
  - Dropped support for PostgreSQL 9.6, Django 4.0 supports PostgreSQL 10 and higher.
  - Format of `CSRF_TRUSTED_ORIGINS` was changed.
- The Docker container now uses Django 4.0, see above for changes.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Uaktualnienie z 4.10 do 4.11

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

- Weblate now requires Python 3.7 or newer.
- The implementation of *Zarządzanie kontrolą dostępu dla poszczególnych projektów* has changed, removing the project prefix from the group names. This affects API users.
- Weblate now uses `charset-normalizer` instead of `chardet` module for character set detection.
- **Changed in 4.11.1:** There is a change in `REST_FRAMEWORK` setting (removal of one of the backends in `DEFAULT_AUTHENTICATION_CLASSES`).

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Uaktualnienie z 4.11 do 4.12

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

- Nie są wymagane żadne specjalne kroki.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

Uaktualnienie z 4.12 do 4.13

Please follow *Ogólne instrukcje dotyczące aktualizacji* in order to perform update.

- The *Definicje języków* are now automatically updated on upgrade, use `UPDATE_LANGUAGES` to disable that.
- Handling of context and location has been changed for *Pliki RC systemu Windows*, *Pliki HTML*, *Format IDML*, and *Pliki tekstowe* file formats. In most cases the context is now shown as location.
- The machine translation services are now configured using the user interface, settings from the configuration file will be imported during the database migration.

Zobacz także:

Ogólne instrukcje dotyczące aktualizacji

2.3.4 Upgrading from Python 2 to Python 3

Weblate no longer supports Python older than 3.6. In case you are still running on older version, please perform migration to Python 3 first on existing version and upgrade later. See Upgrading from Python 2 to Python 3 in the Weblate 3.11.1 documentation.

2.3.5 Migrating from other databases to PostgreSQL

If you are running Weblate on other dabatase than PostgreSQL, you should consider migrating to PostgreSQL as Weblate performs best with it. The following steps will guide you in migrating your data between the databases. Please remember to stop both web and Celery servers prior to the migration, otherwise you might end up with inconsistent data.

Tworzenie bazy danych w PostgreSQL

It is usually a good idea to run Weblate in a separate database, and separate user account:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser -D -P weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Migracja przy użyciu zrzutów Django JSON

The simplest approach for migration is to utilize Django JSON dumps. This works well for smaller installations. On bigger sites you might want to use pgloader instead, see *Migrating to PostgreSQL using pgloader*.

1. Add PostgreSQL as additional database connection to the `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": "",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    },
}
```

2. Run migrations and drop any data inserted into the tables:

```
weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --database=postgresql
```

3. Dump legacy database and import to PostgreSQL

```
weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql
```

4. Adjust `DATABASES` to use just PostgreSQL database as default, remove legacy connection.

Weblate should be now ready to run from the PostgreSQL database.

Migrating to PostgreSQL using pgloader

The pgloader is a generic migration tool to migrate data to PostgreSQL. You can use it to migrate Weblate database.

1. Adjust your `settings.py` to use PostgreSQL as a database.
2. Migrate the schema in the PostgreSQL database:

```
weblate migrate
weblate sqlflush | weblate dbshell
```

3. Run the pgloader to transfer the data. The following script can be used to migrate the database, but you might want to learn more about pgloader to understand what it does and tweak it to match your setup:

```
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 Migracja z Pootle

As Weblate was originally written as replacement from Pootle, it is supported to migrate user accounts from Pootle. You can dump the users from Pootle and import them using `importusers`.

2.4 Tworzenie kopii zapasowych i przenoszenie weblate

2.4.1 Automated backup using BorgBackup

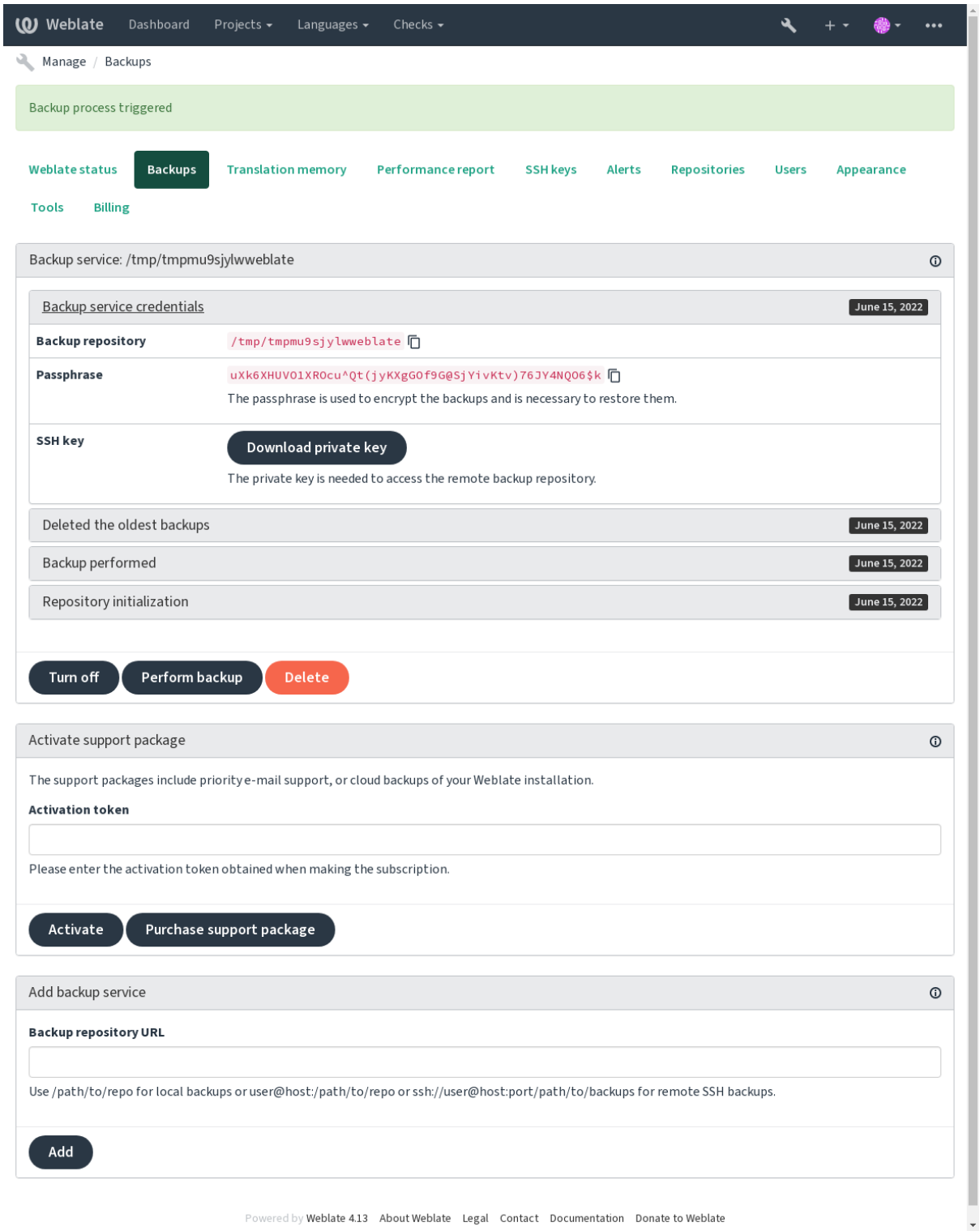
Nowe w wersji 3.9.

Weblate has built-in support for creating service backups using BorgBackup. Borg creates space-effective encrypted backups which can be safely stored in the cloud. The backups can be controlled in the management interface from the *Backups* tab.

Zmienione w wersji 4.4.1: Both PostgreSQL and MySQL/MariaDB databases are included in the automated backups.

The backups using Borg are incremental and Weblate is configured to keep following backups:

- Codzienne kopie zapasowe przez 14 dni wstecz
- Weekly backups for 8 weeks back
- Monthly backups for 6 months back



Klucz szyfrujący Borg

BorgBackup creates encrypted backups and you wouldn't be able to restore them without the passphrase. The passphrase is generated when adding a new backup service and you should copy it and keep it in a secure place.

If you are using *Weblate provisioned backup storage*, please backup your private SSH key too, as it's used to access your backups.

Zobacz także:

borg init

Dostosowywanie kopii zapasowej

- Kopia zapasowa bazy danych może być skonfigurowana za pomocą `DATABASE_BACKUP`.
- The backup creation can be customized using `BORG_EXTRA_ARGS`.

2.4.2 Weblate provisioned backup storage

The easiest way of backing up your Weblate instance is purchasing the backup service at weblate.org. This is how you get it running:

1. Kup *Usługę tworzenia kopii zapasowych* na https://weblate.org/support/#backup.
2. Enter the obtained key in the management interface, see *Integracja wsparcia*.
3. Weblate connects to the cloud service and obtains access info for the backups.
4. Turn on the new backup configuration from the *Backups* tab.
5. Wykonaj kopię zapasową swoich poświadczeń Borg, aby móc przywrócić kopie zapasowe, patrz *Klucz szyfrujący Borg*.

Podpowiedź: The manual step of turning everything on is there for your safety. Without your consent no data is sent to the backup repository obtained through the registration process.

2.4.3 Korzystanie z niestandardowego magazynu kopii zapasowych

You can also use your own storage for the backups. SSH can be used to store backups in the remote destination, the target server needs to have BorgBackup installed.

Zobacz także:

General w dokumentacji

Lokalny system plików

It is recommended to specify the absolute path for the local backup, for example */path/to/backup*. The directory has to be writable by the user running Weblate (see *Uprawnienia systemu plików*). If it doesn't exist, Weblate attempts to create it but needs the appropriate permissions to do so.

Podpowiedź: When running Weblate in Docker, please ensure the backup location is exposed as a volume from the Weblate container. Otherwise the backups will be discarded by Docker upon restarting the container it is in.

One option is to place backups into an existing volume, for example `/app/data/borgbackup`. This is an existing volume in the container.

You can also add a new container for the backups in the Docker Compose file for example by using `/borgbackup`:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

The directory where backups will be stored have to be owned by UID 1000, otherwise Weblate won't be able to write the backups there.

Zdalne kopie zapasowe

For creating remote backups, you will have to install BorgBackup onto another server that's accessible for your Weblate deployment via SSH using the Weblate SSH key:

1. Prepare a server where your backups will be stored.
2. Install the SSH server on it (you will get it by default with most Linux distributions).
3. Install BorgBackup on that server; most Linux distributions have packages available (see Installation).
4. Choose an existing user or create a new user that will be used for backing up.
5. Add Weblate SSH key to the user so that Weblate can SSH to the server without a password (see *Weblate Klucz SSH*).
6. Skonfiguruj lokalizację kopii zapasowej w Weblate jako `użytkownik@host:/ścieżka/do/backupów` lub `ssh://użytkownik@host:port/ścieżka/do/backupów`.

Podpowiedź: *Weblate provisioned backup storage* provides you automated remote backups without any effort.

Zobacz także:

Weblate Klucz SSH, General

2.4.4 Przywracanie z BorgBackup

1. Restore access to your backup repository and prepare your backup passphrase.
2. List all the backups on the server using `borg list REPOSITORY`.
3. Restore the desired backup to the current directory using `borg extract REPOSITORY::ARCHIVE`.
4. Restore the database from the SQL dump placed in the `backup` directory in the Weblate data dir (see *Zrzucone dane do kopii zapasowych*).
5. Copy the Weblate configuration (`backups/settings.py`, see *Zrzucone dane do kopii zapasowych*) to the correct location, see *Dostosowywanie konfiguracji*.

   When using Docker container, the settings file is already included in the container and you should restore the original environment variables. The `environment.yml` file might help you with this (see *Zrzucone dane do kopii zapasowych*).
6. Copy the whole restored data dir to the location configured by `DATA_DIR`.

   Korzystając z kontenerów Dockera, umieść dane w wolumenie danych, zobacz *Woluminy kontenerów platformy Docker*.

   Please make sure the files have correct ownership and permissions, see *Uprawnienia systemu plików*.

The Borg session might look like this:

```
$ 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:
```

Zobacz także:

borg list, borg extract

2.4.5 Ręczna kopia zapasowa

Depending on what you want to save, back up the type of data Weblate stores in each respective place.

Podpowiedź: If you are doing the manual backups, you might want to silence Weblate's warning about a lack of backups by adding `weblate.I028` to `SILENCED_SYSTEM_CHECKS` in `settings.py` or *`WEBLATE_SILENCED_SYSTEM_CHECKS`* for Docker.

```
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
```

Baza danych

The actual storage location depends on your database setup.

Podpowiedź: The database is the most important storage. Set up regular backups of your database. Without the database, all the translations are gone.

Natywna kopia zapasowa bazy danych

The recommended approach is to save a dump of the database using database-native tools such as **`pg_dump`** or **`mysqldump`**. It usually performs better than Django backup, and it restores complete tables with all their data.

You can restore this backup in a newer Weblate release, it will perform all the necessary migrations when running in `migrate`. Please consult *Aktualizacja Weblate* on more detailed info on how to upgrade between versions.

Kopia zapasowa bazy danych Django

Alternatively, you can back up your database using Django's `dumpdata` command. That way the backup is database agnostic and can be used in case you want to change the database backend.

Prior to restoring the database you need to be running exactly the same Weblate version the backup was made on. This is necessary as the database structure does change between releases and you would end up corrupting the data in some way. After installing the same version, run all database migrations using `migrate`.

Afterwards some entries will already be created in the database and you will have them in the database backup as well. The recommended approach is to delete such entries manually using the management shell (see *Wywoływanie poleceń zarządzania*):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Pliki

If you have enough backup space, simply back up the whole *DATA_DIR*. This is a safe bet even if it includes some files you don't want. The following sections describe what you should back up and what you can skip in detail.

Zrzucone dane do kopii zapasowych

Zmienione w wersji 4.7: The environment dump was added as `environment.yml` to help in restoring in the Docker environments.

Przechowywane w *DATA_DIR* `/backups`.

Weblate dumps various data here, and you can include these files for more complete backups. The files are updated daily (requires a running Celery beats server, see *Zadania w tle korzystające z Celery*). Currently, this includes:

- Weblate settings as `settings.py` (there is also expanded version in `settings-expanded.py`).
- Kopia zapasowa bazy danych PostgreSQL jako `database.sql`.
- Environment dump as `environment.yml`.

The database backups are saved as plain text by default, but they can also be compressed or entirely skipped using *DATABASE_BACKUP*.

To restore the database backup load it using database tools, for example:

```
psql --file=database.sql weblate
```

Repozytoria kontroli wersji

Przechowywane w *DATA_DIR* `/vcs`.

The version control repositories contain a copy of your upstream repositories with Weblate changes. If you have *Przesyłaj przy commitowaniu* enabled for all your translation components, all Weblate changes are included upstream. No need to back up the repositories on the Weblate side as they can be cloned again from the upstream location(s) with no data loss.

Klucze SSH i GPG

Stored in *DATA_DIR* `/ssh` and *DATA_DIR* `/home`.

If you are using SSH or GPG keys generated by Weblate, you should back up these locations. Otherwise you will lose the private keys and you will have to regenerate new ones.

Pliki przesłane przez użytkownika

Przechowywane w *DATA_DIR* `/media`.

You should back up all user uploaded files (e.g. *Kontekst wizualny dla ciągów*).

Zadania Celery

The Celery task queue might contain some info, but is usually not needed for a backup. At most you will lose updates not yet been processed to translation memory. It is recommended to perform the fulltext or repository update upon restoration anyhow, so there is no problem in losing these.

Zobacz także:

Zadania w tle korzystające z Celery

Wiersz poleceń do ręcznego tworzenia kopii zapasowych

Using a cron job, you can set up a Bash command to be executed on a daily basis, for example:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz␣
↪backups vcs ssh home media fonts secret
```

The string between the quotes after *XZ_OPT* allows you to choose your xz options, for instance the amount of memory used for compression; see https://linux.die.net/man/1/xz

You can adjust the list of folders and files to your needs. To avoid saving the translation memory (in backups folder), you can use:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz␣
↪backups/database.sql backups/settings.py vcs ssh home media fonts secret
```

2.4.6 Przywracanie ręcznej kopii zapasowej

1. Restore all data you have backed up.
2. Zaktualizuj wszystkie repozytoria za pomocą `updategit`.

   ```
   weblate updategit --all
   ```

2.4.7 Przenoszenie instalacji Weblate

Relocate your installation to a different system by following the backing up and restoration instructions above.

Zobacz także:

Upgrading from Python 2 to Python 3, *Migrating from other databases to PostgreSQL*

2.5 Uwierzytelnienie

2.5.1 Rejestracja użytkownika

The default setup for Weblate is to use python-social-auth, a form on the website to handle registration of new users. After confirming their e-mail a new user can contribute or authenticate by using one of the third party services.

You can also turn off registration of new users using `REGISTRATION_OPEN`.

The authentication attempts are subject to *Ograniczenie szybkości*.

2.5.2 Backendy uwierzytelniania

The built-in solution of Django is used for authentication, including various social options to do so. Using it means you can import the user database of other Django-based projects (see *Migracja z Pootle*).

Django can additionally be set up to authenticate against other means too.

Zobacz także:

Ustawienia uwierzytelniania describes how to configure authentication in the official Docker image.

2.5.3 Uwierzytelnianie społecznościowe

Thanks to Welcome to Python Social Auth's documentation!, Weblate support authentication using many third party services such as GitLab, Ubuntu, Fedora, etc.

Please check their documentation for generic configuration instructions in Django Framework.

Informacja: By default, Weblate relies on third-party authentication services to provide a validated e-mail address. If some of the services you want to use don't support this, please enforce e-mail validation on the Weblate side by configuring FORCE_EMAIL_VALIDATION for them. For example:

```
SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True
```

Zobacz także:

Pipeline

Enabling individual backends is quite easy, it's just a matter of adding an entry to the `AUTHENTICATION_BACKENDS` setting and possibly adding keys needed for a given authentication method. Please note that some backends do not provide user e-mail by default, you have to request it explicitly, otherwise Weblate will not be able to properly credit contributions users make.

Podpowiedź: Most of the authentication backends require HTTPS. Once HTTPS is enabled in your web server please configure Weblate to report it properly using *`ENABLE_HTTPS`*, or by *`WEBLATE_ENABLE_HTTPS`* in the Docker container.

Zobacz także:

Python Social Auth backend

Uwierzytelnianie OpenID

For OpenID-based services it's usually just a matter of enabling them. The following section enables OpenID authentication for OpenSUSE, Fedora and 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",
)
```

Zobacz także:

OpenID

Uwierzytelnianie GitHub

You need to register an OAuth application on GitHub and then tell Weblate all its secrets:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.github.GithubOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = "GitHub Client ID"
SOCIAL_AUTH_GITHUB_SECRET = "GitHub Client Secret"
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
```

The GitHub should be configured to have callback URL as `https://example.com/accounts/complete/github/`.

There are similar authentication backends for GitHub for Organizations and GitHub for Teams. Their settings are named `SOCIAL_AUTH_GITHUB_ORG_*` and `SOCIAL_AUTH_GITHUB_TEAM_*`, and they require additional setting of the scope - `SOCIAL_AUTH_GITHUB_ORG_NAME` or `SOCIAL_AUTH_GITHUB_TEAM_ID`. Their callback URLs are `https://example.com/accounts/complete/github-org/` and `https://example.com/accounts/complete/github-teams/`.

Informacja: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Ustaw prawidłową domenę strony*.

Zobacz także:

GitHub

Uwierzytelnianie Bitbucket

You need to register an application on Bitbucket and then tell Weblate all its secrets:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.bitbucket.BitbucketOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = "Bitbucket Client ID"
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = "Bitbucket Client Secret"
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True
```

Informacja: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Ustaw prawidłową domenę strony*.

Zobacz także:

Bitbucket

Google OAuth 2

To use Google OAuth 2, you need to register an application on <https://console.developers.google.com/> and enable the Google+ API.

The redirect URL is `https://WEBLATE SERVER/accounts/complete/google-oauth2/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"
```

Informacja: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Ustaw prawidłową domenę strony*.

Zobacz także:

Google

Facebook OAuth 2

As per usual with OAuth 2 services, you need to register your application with Facebook. Once this is done, you can set up Weblate to use it:

The redirect URL is `https://WEBLATE SERVER/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"]
```

Informacja: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Ustaw prawidłową domenę strony*.

Zobacz także:

Facebook

GitLab OAuth 2

For using GitLab OAuth 2, you need to register an application on <https://gitlab.com/profile/applications>.

The redirect URL is `https://WEBLATE SERVER/accounts/complete/gitlab/` and ensure you mark the *read_user* scope.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.gitlab.GitLabOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITLAB_KEY = "Application ID"
SOCIAL_AUTH_GITLAB_SECRET = "Secret"
SOCIAL_AUTH_GITLAB_SCOPE = ["read_user"]

# If you are using your own GitLab
# SOCIAL_AUTH_GITLAB_API_URL = 'https://gitlab.example.com/'
```

Informacja: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Ustaw prawidłową domenę strony*.

Zobacz także:

GitLab

Microsoft Azure Active Directory

Weblate can be configured to use common or specific tenants for authentication.

The redirect URL is `https://WEBLATE SERVER/accounts/complete/azuread-oauth2/` for common and `https://WEBLATE SERVER/accounts/complete/azuread-tenant-oauth2/` for tenant-specific authentication.

```
# 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
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET = ""
# Tenant ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID = ""
```

Informacja: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Ustaw prawidłową domenę strony*.

Zobacz także:

Microsoft Azure Active Directory

Slack

For using Slack OAuth 2, you need to register an application on <https://api.slack.com/apps>.

The redirect URL is `https://WEBLATE SERVER/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 = ""
```

Informacja: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Ustaw prawidłową domenę strony*.

Zobacz także:

Slack

Nadpisywanie nazw i ikon metod uwierzytelniania

You can override the authentication method display name and icon using using settings as `SOCIAL_AUTH_<NAME>_IMAGE` and `SOCIAL_AUTH_<NAME>_TITLE`. For example overriding naming for Auth0 would look like:

```
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

Wyłączanie uwierzytelniania hasłem

E-mail and password authentication can be turned off by removing `social_core.backends.email.EmailAuth` from `AUTHENTICATION_BACKENDS`. Always keep `weblate.accounts.auth.WeblateUserBackend` there, it is needed for core Weblate functionality.

Disabling e-mail authentication will disable all e-mail related functionality – user invitation or password reset feature.

Wskazówka: You can still use password authentication for the admin interface, for users you manually create there. Just navigate to `/admin/login/`.

For example authentication using only the openSUSE Open ID provider can be achieved using the following:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Uwierzytelnianie hasłem

The default `settings.py` comes with a reasonable set of `AUTH_PASSWORD_VALIDATORS`:

- Passwords can't be too similar to your other personal info.
- Passwords must contain at least 10 characters.
- Passwords can't be a commonly used password.
- Hasła nie mogą zawierać tylko i wyłącznie cyfr.
- Passwords can't consist of a single character or only whitespace.
- Passwords can't match a password you have used in the past.

You can customize this setting to match your password policy.

Additionally you can also install django-zxcvbn-password which gives quite realistic estimates of password difficulty and allows rejecting passwords below a certain threshold.

2.5.5 Uwierzytelnianie SAML

Nowe w wersji 4.1.1.

Please follow the Python Social Auth instructions for configuration. Notable differences:

- Weblate supports single IDP which has to be called `weblate` in `SOCIAL_AUTH_SAML_ENABLED_IDPS`.
- The SAML XML metadata URL is `/accounts/metadata/saml/`.
- Following settings are automatically filled in: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT`, `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Przykładowa konfiguracja:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.saml.SAMLAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
# Social auth backends setup
SOCIAL_AUTH_SAML_SP_ENTITY_ID = f"https://{SITE_DOMAIN}/accounts/metadata/saml/"
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbnl+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
SOCIAL_AUTH_SAML_ORG_INFO = {
    "en-US": {
        "name": "example",
        "displayname": "Example Inc.",
        "url": "http://example.com"
    }
}
SOCIAL_AUTH_SAML_TECHNICAL_CONTACT = {
    "givenName": "Tech Gal",
    "emailAddress": "technical@example.com"
}
SOCIAL_AUTH_SAML_SUPPORT_CONTACT = {
    "givenName": "Support Guy",
    "emailAddress": "support@example.com"
}
```

The default configuration extracts user details from following attributes, configure your IDP to provide them:

Atrybut	Referencja URI SAML
Imię i nazwisko	`urn:oid:2.5.4.3`
Imię	`urn:oid:2.5.4.42`
Nazwisko	`urn:oid:2.5.4.4`
Adres e-mail	`urn:oid:0.9.2342.19200300.100.1.3`
Nazwa użytkownika	`urn:oid:0.9.2342.19200300.100.1.1`

Podpowiedź: The example above and the Docker image define an IDP called `weblate`. You might need to configure this string as *Relay* in your IDP.

Zobacz także:

Configuring SAML in Docker, SAML

2.5.6 Uwierzytelnianie LDAP

LDAP authentication can be best achieved using the *django-auth-ldap* package. You can install it via usual means:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap
```

Podpowiedź: This package is included in the Docker container, see *Instalowanie za pomocą Dockera*.

Informacja: There are some incompatibilities in the Python LDAP 3.1.0 module, which might prevent you from using that version. If you get error AttributeError: «module» object has no attribute «_trace_level», downgrading python-ldap to 3.0.0 might help.

Once you have the package installed, you can hook it into the Django authentication:

```
# Add LDAP backed, keep Django one if you want to be able to sign in
# even without LDAP for admin account
AUTHENTICATION_BACKENDS = (
    "django_auth_ldap.backend.LDAPBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# LDAP server address
AUTH_LDAP_SERVER_URI = "ldaps://ldap.example.net"

# DN to use for authentication
AUTH_LDAP_USER_DN_TEMPLATE = "cn=%(user)s,o=Example"
# Depending on your LDAP server, you might use a different DN
# like:
# AUTH_LDAP_USER_DN_TEMPLATE = 'ou=users,dc=example,dc=com'

# List of attributes to import from LDAP upon sign in
# Weblate stores full name of the user in the full_name attribute
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "name",
    # Use the following if your LDAP server does not have full name
    # Weblate will merge them later
    # 'first_name': 'givenName',
    # 'last_name': 'sn',
    # Email is required for Weblate (used in VCS commits)
    "email": "mail",
}

# Hide the registration form
REGISTRATION_OPEN = False
```

Informacja: You should remove `'social_core.backends.email.EmailAuth'` from the `AUTHENTICATION_BACKENDS` setting, otherwise users will be able to set their password in Weblate, and authenticate using that. Keeping `'weblate.accounts.auth.WeblateUserBackend'` is still needed in order to make permissions and facilitate anonymous users. It will also allow you to sign in using a local admin account, if you have created it (e.g. by using `createadmin`).

Używanie hasła powiązania

If you can not use direct bind for authentication, you will need to use search, and provide a user to bind for the search. For example:

```
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)"
)
```

Integracja 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
```

Zobacz także:

Django Authentication Using LDAP, Authentication

2.5.7 Uwierzytelnianie CAS

CAS authentication can be achieved using a package such as *django-cas-ng*.

Step one is disclosing the e-mail field of the user via CAS. This has to be configured on the CAS server itself, and requires you run at least CAS v2 since CAS v1 doesn't support attributes at all.

Step two is updating Weblate to use your CAS server and attributes.

Aby zainstalować *django-cas-ng*:

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

Once you have the package installed you can hook it up to the Django authentication system by modifying the `settings.py` file:

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

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

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

Finally, a signal can be used to map the e-mail field to the user object. For this to work you have to import the signal from the *django-cas-ng* package and connect your code with this signal. Doing this in settings file can cause problems, therefore it's suggested to put it:

- In your app config's `django.apps.AppConfig.ready()` method
- In the project's `urls.py` file (when no models exist)

```
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()
```

Zobacz także:

Django CAS NG

2.5.8 Configuring third party Django authentication

Generally any Django authentication plugin should work with Weblate. Just follow the instructions for the plugin, just remember to keep the Weblate user backend installed.

Zobacz także:

Uwierzytelnianie LDAP, *Uwierzytelnianie CAS*

Typically the installation will consist of adding an authentication backend to `AUTHENTICATION_BACKENDS` and installing an authentication app (if there is any) into `INSTALLED_APPS`:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

2.6 Kontrola dostępu

Weblate używa szczegółowego systemu uprawnień, który jest przystosowany do przypisywania użytkownikom uprawnień dostępu do całej instancji projektu lub w ograniczonym jego zakresie.

Zmienione w wersji 3.0: Przed wersją Weblate 3.0 system uprawnień oparty był tylko na systemie uprawnień Django, ale teraz został stworzony specjalnie dla Weblate. Jeśli używasz czegoś starszego, zapoznaj się z dokumentacją dla konkretnej wersji, której używasz.

2.6.1 Prosta kontrola dostępu

Jeśli nie administrujesz całą instalacją Weblate, a jedynie masz dostęp do zarządzania określonymi projektami (jak w przypadku Hosted Weblate), opcje zarządzania kontrolą dostępu są ograniczone do poniższych ustawień. Jeśli nie potrzebujesz skomplikowanej konfiguracji, te ustawienia są dla Ciebie wystarczające.

Ustawienia kontroli dostępu do projektu

Informacja: Ta funkcja jest niedostępna dla projektów korzystających z planu Libre na Hosted Weblate.

Możesz ograniczyć dostęp użytkownika do poszczególnych projektów poprzez wybranie innego ustawienia *Kontrola dostępu*. Dostępne opcje to:

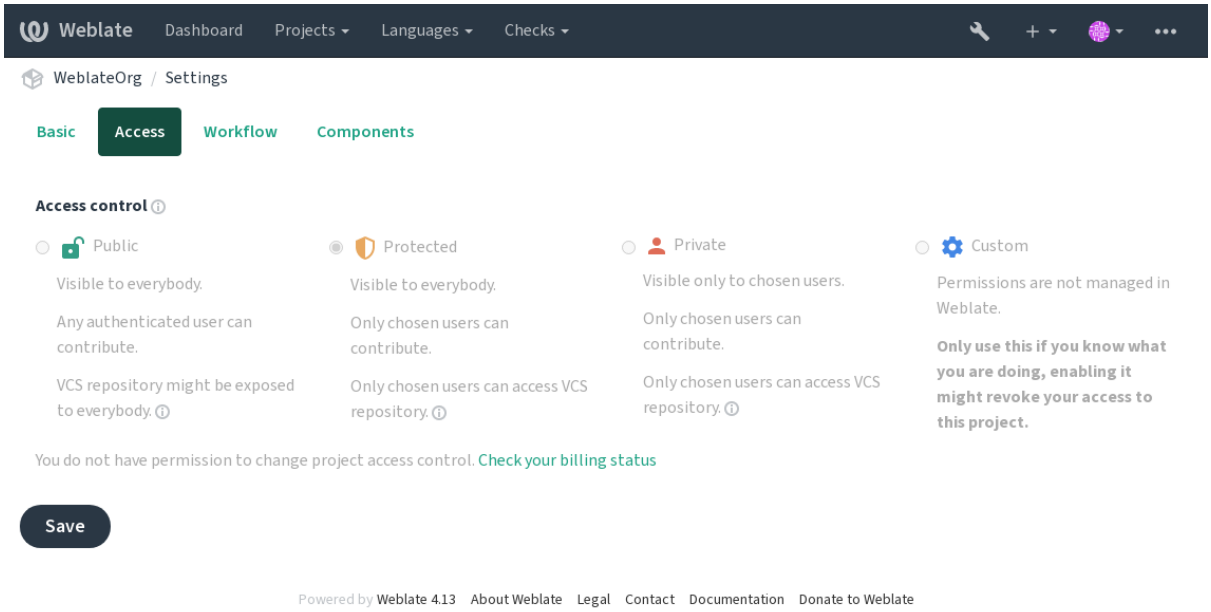
Publiczny
Widoczne publicznie, możliwe do przetłumaczenia dla wszystkich zalogowanych użytkowników.

Chroniony
Publicznie widoczne, ale możliwe do przetłumaczenia tylko dla wybranych użytkowników.

Prywatny
Widoczne i możliwe do przetłumaczenia tylko dla wybranych użytkowników.

Niestandardowy
Funkcje zarządzania użytkownikami zostaną wyłączone; domyślnie wszyscy użytkownicy nie mogą wykonywać żadnych działań w projekcie. Musisz skonfigurować wszystkie uprawnienia za pomocą *Niestandardowa kontrola dostępu*.

Kontrolę dostępu można zmienić w zakładce *Dostęp* w konfiguracji (*Zarządzaj* ↓ *Ustawienia*) odpowiedniego projektu.



Domyślna wartość może być zmieniona przez `DEFAULT_ACCESS_CONTROL`.

Informacja: Nawet dla projektów *Private*, niektóre informacje o twoim projekcie zostaną ujawnione: statystyki i podsumowanie językowe dla całej instancji będzie zawierać zliczenia dla wszystkich projektów, pomimo ustawienia kontroli dostępu. Twoja nazwa projektu i inne informacje nie mogą być przez to ujawnione.

Informacja: Rzeczywisty zestaw uprawnień dostępnych domyślnie dla użytkowników w projektach *Publicznych*, *Chronionych* i *Prywatnych* może zostać przedefiniowany przez administratora instancji Weblate przy użyciu *ustawienia niestandardowe*.

Ostrzeżenie: Włączając kontrolę dostępu *Custom*, Weblate usunie wszystkie *grupy specjalne* które utworzył dla wybranego projektu. Jeśli robisz to bez uprawnień administratora dla całej instancji Weblate, natychmiast stracisz dostęp do zarządzania projektem.

Zobacz także:

Kontrola dostępu

Zarządzanie kontrolą dostępu dla poszczególnych projektów

Użytkownicy z uprawnieniem *Zarządzaj dostępem do projektu* (patrz *Lista uprawnień i wbudowanych ról*) mogą zarządzać użytkownikami w projektach przez dodawanie ich do zespołów. Początkowy zbiór zespołów jest dostarczany przez Weblate, ale można zdefiniować dodatkowe, zapewniające bardziej szczegółową kontrolę dostępu. Możesz ograniczyć zespoły do języków i przypisać im wyznaczone role dostępu (patrz *Lista uprawnień i wbudowanych ról*).

Następujące zespoły są tworzone automatycznie dla każdego projektu:

W przypadku projektów „Publicznych”, „Chronionych” i „Prywatnych”:

Administracja
Zawiera wszystkie uprawnienia dostępne dla danego projektu.

Przegląd (tylko jeśli *review workflow* jest włączone)
Może zatwierdzać tłumaczenia podczas przeglądu.

Tylko dla projektów *Chroniony* i *Prywatny*:

Przetłumacz
Może tłumaczyć projekt i przesłać tłumaczenia wykonane w trybie offline.

Źródła
Może edytować ciągi źródłowe (jeśli jest to dozwolone w *project settings*) i informacje o ciągach źródłowych.

Języki
Może zarządzać przetłumaczonymi językami (dodawać lub usuwać tłumaczenia).

Słownik
Może zarządzać słownikiem (dodawać, usuwać lub przesyłać).

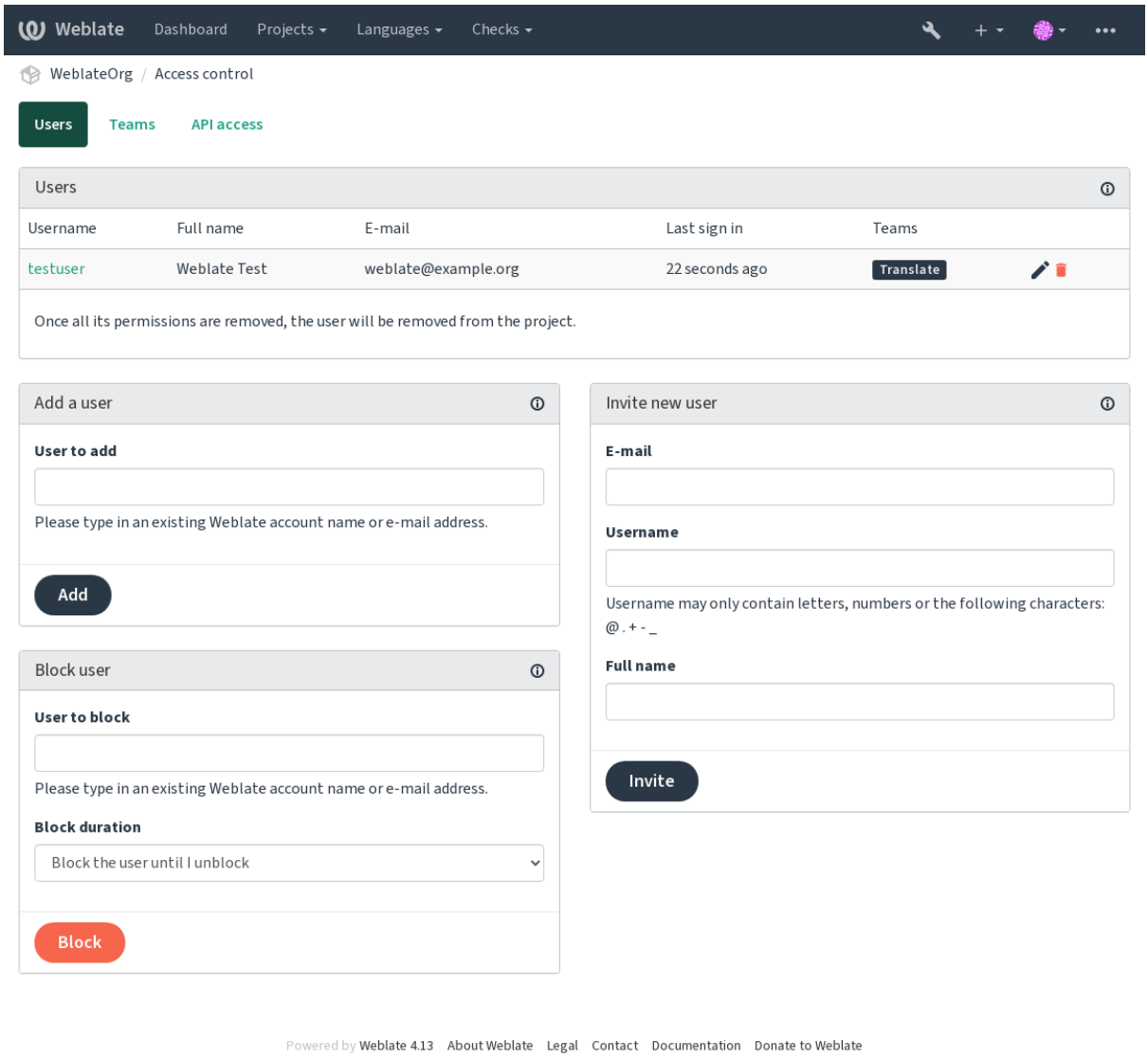
Pamięć
Może zarządzać pamięcią tłumaczeniową.

Zrzuty ekranu
Może zarządzać zrzutami ekranu (dodawać je lub usuwać oraz przypisywać do ciągów źródłowych).

Tłumaczenie automatyczne
Może korzystać z automatycznego tłumaczenia.

VCS
Może zarządzać VCS i uzyskiwać dostęp do wyeksportowanego repozytorium.

Rozliczenia
Może uzyskać dostęp do informacji i ustawień billingowych (patrz *Rozliczenia*).



Funkcje te są dostępne na stronie *Kontrola dostępu*, do której można uzyskać dostęp z menu projektu *Zarządzaj* ↓ *Użytkownicy*.

Zaproszenie nowego użytkownika

Ponadto, oprócz dodania do projektu istniejącego użytkownika, możliwe jest zapraszanie nowych. Każdy nowy użytkownik zostanie utworzony natychmiast, ale konto pozostanie nieaktywne do czasu zalogowania się za pomocą linku w zaproszeniu wysłanym na e-mail. W tym celu nie jest wymagane posiadanie żadnych uprawnień w całej witrynie, wystarczające byłyby uprawnienia do zarządzania dostępem w zakresie projektu (np. członkostwo w zespole „Administracja”).

Podpowiedź: Jeśli zaproszony użytkownik nie dotrzyma terminu ważności zaproszenia, może ustawić swoje hasło używając zaproszonego adresu e-mail w formularzu resetowania hasła, ponieważ konto jest już utworzone.

Nowe w wersji 3.11: Istnieje możliwość ponownego wysłania wiadomości e-mail z zaproszeniem użytkownika (unieważnienie wcześniej wysłanego zaproszenia).

Ten sam rodzaj zaproszeń jest dostępny na całej stronie z *management interface* na karcie *Użytkownicy*.

Blokowanie użytkowników

Nowe w wersji 4.7.

W przypadku, gdy niektórzy użytkownicy zachowują się źle w Twoim projekcie, masz możliwość zablokowania im udziału. Zablokowany użytkownik nadal będzie mógł zobaczyć projekt, jeśli ma do tego uprawnienia, ale nie będzie mógł wnosić wkładu.

Zarządzanie uprawnieniami dla poszczególnych projektów

Możesz ustawić swoje projekty na *Chronione* lub *Prywatne* i *zarządzać użytkownikami* dla każdego projektu w interfejsie użytkownika Weblate.

Domyślnie uniemożliwia to Weblate przyznanie dostępu zapewnianego przez *Users* i *Viewers* *default groups* ze względu na własną konfigurację tych grup. Nie przeszkadza to w przyznawaniu uprawnień do tych projektów w całej witrynie poprzez zmianę domyślnych grup, tworzenie nowej lub tworzenie dodatkowych niestandardowych ustawień dla poszczególnych komponentów, jak opisano w *Niestandardowa kontrola dostępu* poniżej.

Jedną z głównych korzyści zarządzania uprawnieniami przez interfejs użytkownika Weblate jest to, że możesz delegować je innym użytkownikom bez nadawania im uprawnień superużytkownika. W tym celu dodaj je do zespołu *Administracja* projektu.

2.6.2 Niestandardowa kontrola dostępu

Informacja: Ta funkcja jest niedostępna dla projektów korzystających z planu Libre na Hosted Weblate.

System uprawnień jest oparty na grupach i rolach, gdzie role określają zestaw uprawnień, a grupy przypisują te uprawnienia użytkownikom i tłumaczeniom; więcej szczegółów, zobacz: *Użytkownicy, role, grupy i uprawnienia*.

Najpotężniejsze funkcje systemu kontroli dostępu Weblate są na razie dostępne tylko przez *Django admin interface*. Możesz go używać do zarządzania uprawnieniami dowolnego projektu. Nie musisz koniecznie przełączać go na *Niestandardowy* *access control*, aby go wykorzystać. Jednak aby z niego korzystać, musisz mieć uprawnienia superużytkownika.

Jeśli nie interesują Cię szczegóły implementacji, a po prostu chcesz stworzyć wystarczająco prostą konfigurację opartą na ustawieniach domyślnych, lub nie masz dostępu do całej instalacji Weblate (jak na Hosted Weblate), zapoznaj się z sekcją *Prosta kontrola dostępu*.

Typowe ustawienia

Ta sekcja zawiera przegląd niektórych typowych konfiguracji, które mogą Cię zainteresować.

Zarządzanie uprawnieniami na całej stronie

Aby zarządzać uprawnieniami dla całej instancji naraz, dodaj użytkowników do odpowiednich *default groups*:

- *Użytkownicy* (domyślnie robi to *automatic group assignment*).
- *Recenzenci* (jeśli używasz *review workflow* z dedykowanymi recenzentami).
- *Menedżerowie* (jeśli chcesz delegować większość operacji zarządzania komuś innemu).

Wszystkie projekty powinny być skonfigurowane jako *Publiczny* (patrz *Ustawienia kontroli dostępu do projektu*), w przeciwnym razie uprawnienia dla całej strony zapewnione przez członkostwo w grupach *Użytkownicy* i *Recenzenci* nie przyniosą żadnego efektu.

Możesz również przyznać dodatkowe, wybrane przez siebie uprawnienia, grupom domyślnym. Na przykład możesz chcieć nadać uprawnienia do zarządzania zrzutami ekranu wszystkim „Użytkownikom”.

Możesz również zdefiniować kilka nowych grup niestandardowych. Jeśli chcesz nadal zarządzać swoimi uprawnieniami w całej witrynie dla tych grup, wybierz odpowiednią wartość dla *Wybór projektu* (np. *Wszystkie projekty* lub *Wszystkie projekty publiczne*).

Niestandardowe uprawnienia do języków, komponentów lub projektów

Możesz tworzyć własne grupy dedykowane do zarządzania uprawnieniami do różnych obiektów, takich jak języki, komponenty i projekty. Chociaż te grupy mogą przyznawać tylko dodatkowe uprawnienia, nie można cofnąć żadnego uprawnienia przyznanego przez grupy obejmujące całą witrynę lub poszczególne projekty, dodając kolejną grupę niestandardową.

Przykład:

Jeśli chcesz (z jakiegokolwiek powodu) zezwolić na tłumaczenie na określony język (powiedzmy „czeski”) tylko zamkniętemu zestawowi wiarygodnych tłumaczy, jednocześnie pozostawiając tłumaczenia na inne języki publicznie, będziesz musiał:

1. Usuń uprawnienie do tłumaczenia języka czeskiego wszystkim użytkownikom. W domyślnej konfiguracji można to zrobić, zmieniając *Użytkownicy* *default group*.

Tabela 1: Grupa *Użytkownicy*

Wybierz język	*Jak zdefiniowano*
Języki	Wszystkie oprócz *czeski*

2. Dodaj dedykowaną grupę dla tłumaczy języka czeskiego.

Tabela 2: Grupa *Tłumacze języka czeskiego*

Role	*Użytkownik zaawansowany*
Wybór projektu	*Wszystkie projekty publiczne*
Wybierz język	*Jak zdefiniowano*
Języki	*czeski*

3. Dodaj użytkowników, którym chcesz nadać uprawnienia do tej grupy.

Jak widać, zarządzanie uprawnieniami w ten sposób jest potężne, ale może być dość żmudną pracą. Nie możesz przekazać go innemu użytkownikowi, chyba że nadasz uprawnienia superużytkownika.

Użytkownicy, role, grupy i uprawnienia

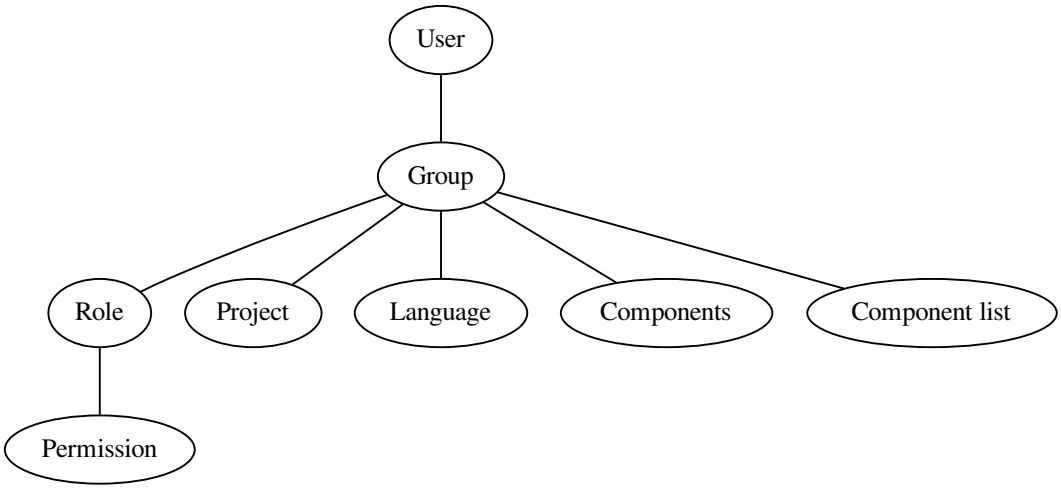
Modele uwierzytelniania składają się z kilku obiektów:

Uprawnienia
Indywidualne uprawnienia zdefiniowane przez weblate. Uprawnienia nie mogą być przypisane do użytkowników. Można to zrobić tylko poprzez przypisanie ról.

Rola
Rola definiuje zestaw uprawnień. Umożliwia to ponowne użycie tych zestawów w kilku miejscach, ułatwiając administrację.

Użytkownik
Użytkownik może należeć do kilku grup.

Grupa
Grupa łączy role, użytkowników i obiekty uwierzytelniające (projekty, języki i listy komponentów).



Informacja: Grupa może nie mieć przypisanych żadnych ról, w takim przypadku zakłada się, że każdy ma dostęp do przeglądania projektu (zobacz poniżej).

Dostęp do przeglądania projektu

Użytkownik musi być członkiem grupy powiązanej z projektem lub dowolnym komponentem wewnątrz tego projektu. Posiadanie członkostwa jest wystarczające, do przeglądania projektu nie są potrzebne żadne specjalne uprawnienia (jest to używane w domyślnej grupie *Viewers*, zobacz *Lista grup*).

Dostęp do przeglądania komponentu

A user can access unrestricted components once able to access the components' project (and will have all the permissions the user was granted for the project). With *Ograniczony dostęp* turned on, access to the component requires explicit permissions for the component (or a component list the component is in).

Zakres grup

Zakres uprawnień nadawanych przez role w grupach jest określany według następujących reguł:

- If the group specifies any *Component list*, all the permissions given to members of that group are granted for all the components in the component lists attached to the group, and an access with no additional permissions is granted for all the projects these components are in. *Components* and *Projects* are ignored.
- If the group specifies any *Components*, all the permissions given to the members of that group are granted for all the components attached to the group, and an access with no additional permissions is granted for all the projects these components are in. *Projects* are ignored.
- Otherwise, if the group specifies any *Projects*, either by directly listing them or by having *Projects selection* set to a value like *All public projects*, all those permissions are applied to all the projects, which effectively grants the same permissions to access all projects *unrestricted components*.
- The restrictions imposed by a group's *Languages* are applied separately, when it's verified if a user has an access to perform certain actions. Namely, it's applied only to actions directly related to the translation process itself like reviewing, saving translations, adding suggestions, etc.

Podpowiedź: Use *Language selection* or *Project selection* to automate inclusion of all languages or projects.

Przykład:

Przyjmijmy, że jest projekt `foo` z komponentami: `foo/bar` i `foo/baz` oraz następującą grupą:

Tabela 3: Grupa «Hiszpańscy administratorzy-recenzenci»

Role	*Przejrzyj ciągi*, *Zarządzaj repozytorium*
Komponenty	foo/bar
Języki	*hiszpański*

Członkowie tej grupy będą mieli następujące uprawnienia (przy założeniu domyślnych ustawień ról):

- Ogólny dostęp (przeglądanie) do całego projektu `foo` łącznie z obydwoma komponentami: `foo/bar` i `foo/baz`.
- Review strings in `foo/bar` Spanish translation (not elsewhere).
- Manage VCS for the whole `foo/bar` repository e.g. commit pending changes made by translators for all languages.

Automatyczne przypisywania grupy

On the bottom of the *Group* editing page in the *Django admin interface*, you can specify *Automatic group assignments*, which is a list of regular expressions used to automatically assign newly created users to a group based on their e-mail addresses. This assignment only happens upon account creation.

The most common use-case for the feature is to assign all new users to some default group. In order to do so, you will probably want to keep the default value (`^.*$`) in the regular expression field. Another use-case for this option might be to give some additional privileges to employees of your company by default. Assuming all of them use corporate e-mail addresses on your domain, this can be accomplished with an expression like `^.*@mycompany.com`.

Informacja: Automatic group assignment to *Users* and *Viewers* is always recreated when upgrading from one Weblate version to another. If you want to turn it off, set the regular expression to `^$` (which won't match anything).

Informacja: As for now, there is no way to bulk-add already existing users to some group via the user interface. For that, you may resort to using the *REST API*.

Domyślne grupy i role

After installation, a default set of groups is created (see *Lista grup*).

These roles and groups are created upon installation. The built-in roles are always kept up to date by the database migration when upgrading. You can't actually change them, please define a new role if you want to define your own set of permissions.

Lista uprawnień i wbudowanych ról

Zakres	Uprawnienie	Role
Rozliczenia (patrz *Rozliczenia*)	Wyświetl informacje rozliczeniowe	*Administracja*, *Rozliczenia*
Zmiany	Pobierz zmiany	*Administracja*
Komentarze	Wyślij komentarz	*Administracja*, *Edytuj źródło*, *Użytkownik zaaw*
	Usuń komentarz	*Administracja*
	Resolve comment	*Administracja*, *Przejrzyj ciągi*
Komponent	Edytuj ustawienia komponentów	*Administracja*
	Blokuj komponent, uniemożliwiając tłumaczenia	*Administracja*
Słownik	Dodaj wpis słownika	*Administracja*, *Zarządzaj słownikiem*, *Użytkow*
	Edytuj wpis słownika	*Administracja*, *Zarządzaj słownikiem*, *Użytkow*
	Usuń wpis słownika	*Administracja*, *Zarządzaj słownikiem*, *Użytkow*
	Prześlij wpisy do słownika	*Administracja*, *Zarządzaj słownikiem*, *Użytkow*
Automatyczne sugestie	Użyj automatycznych sugestii	*Administracja*, *Edytuj źródło*, *Użytkownik zaaw*
Pamięć tłumaczeniowa	Edytuj pamięć tłumaczeń	*Administracja*, *Zarządzaj pamięcią tłumaczenio*
	Usuń pamięć tłumaczeń	*Administracja*, *Zarządzaj pamięcią tłumaczenio*
Projekty	Edytuj ustawienia projektu	*Administracja*
	Zarządzaj dostępem do projektu	*Administracja*
Raporty	Pobierz raporty	*Administracja*
Zrzuty ekranu	Dodaj zrzut ekranu	*Administracja*, *Zarządzaj zrzutami ekranu*
	Edytuj zrzut ekranu	*Administracja*, *Zarządzaj zrzutami ekranu*
	Usuń zrzut ekranu	*Administracja*, *Zarządzaj zrzutami ekranu*
Ciągi źródłowe	Edytuj dodatkowe informacje o ciągu	*Administracja*, *Edytuj źródło*
Ciągi	Dodać nowy ciąg	*Administracja*
	Usuń ciąg	*Administracja*
	Odrzuć nieudaną kontrolę	*Administracja*, *Edytuj źródło*, *Użytkownik zaaw*
	Edytuj ciągi	*Administracja*, *Edytuj źródło*, *Użytkownik zaaw*
	Przejrzyj ciągi	*Administracja*, *Przejrzyj ciągi*
	Edytuj ciąg po wymuszeniu sugestii	*Administracja*, *Przejrzyj ciągi*
	Edytuj ciągi źródłowe	*Administracja*, *Edytuj źródło*, *Użytkownik zaaw*
Sugestie	Zaakceptuj sugestię	*Administracja*, *Edytuj źródło*, *Użytkownik zaaw*
	Dodaj sugestię	*Administracja*, *Edytuj źródło*, *Dodaj sugestię*, *U*
	Usuń sugestię	*Administracja*, *Użytkownik zaawansowany*
	Głosuj na sugestię	*Administracja*, *Edytuj źródło*, *Użytkownik zaaw*
Tłumaczenia	Dodaj język do tłumaczenia	*Administracja*, *Użytkownik zaawansowany*, *Za*
	Wykonaj automatyczne tłumaczenie	*Administracja*, *Tłumaczenie automatyczne*
	Usuń istniejące tłumaczenie	*Administracja*, *Zarządzaj językami*
	Pobierz plik tłumaczenia	*Administracja*, *Edytuj źródło*, *Uzyskaj dostęp do*
	Dodaj kilka języków do tłumaczenia	*Administracja*, *Zarządzaj językami*
Wgrywanie danych	Zdefiniuj autora przesłanego tłumaczenia	*Administracja*
	Zastąp istniejące ciągi przesyłaniem	*Administracja*, *Edytuj źródło*, *Użytkownik zaaw*
	Prześlij tłumaczenia	*Administracja*, *Edytuj źródło*, *Użytkownik zaaw*
VCS	Uzyskaj dostęp do wewnętrznego repozytorium	*Administracja*, *Uzyskaj dostęp do repozytorium*,
	Scommituj zmiany na wewnętrzne repozytorium	*Administracja*, *Zarządzaj repozytorium*
	Wypchnij zmianę z wewnętrznego repozytorium	*Administracja*, *Zarządzaj repozytorium*
	Zresetuj zmiany w wewnętrznym repozytorium	*Administracja*, *Zarządzaj repozytorium*
	Wyświetl lokalizację repozytorium nadrzędnego	*Administracja*, *Uzyskaj dostęp do repozytorium*,
	Zaktualizuj wewnętrzne repozytorium	*Administracja*, *Zarządzaj repozytorium*
Uprawnienia dla całej strony	Użyj interfejsu zarządzania	
	Dodaj nowe projekty	
	Dodaj definicje językowe	
	Zarządzaj definicjami języków	
	Zarządzaj grupami	
	Zarządzaj użytkownikami	
	Zarządzaj rolami	

Tabela 4 – kontynuacja poprzedniej stro

Zakres	Uprawnienie	Role
	Zarządzaj ogłoszeniami	
	Zarządzaj pamięcią tłumaczeniową	
	Manage machinery	
	Zarządzaj listami komponentów	

Informacja: Site-wide privileges are not granted to any default role. These are powerful and quite close to superuser status. Most of them affect all projects in your Weblate installation.

Lista grup

The following groups are created upon installation (or after executing `setupgroups`) and you are free to modify them. The migration will, however, re-create them if you delete or rename them.

Goście
Definiuje uprawnienia dla nieuwierzytelnionych użytkowników.

Ta grupa zawiera tylko anonimowych użytkowników (patrz :setting: *ANONYMOUS_USER_NAME*).

Możesz usunąć role z tej grupy, aby ograniczyć uprawnienia dla nieuwierzytelnionych użytkowników.

Role domyślne: *Dodaj sugestię*, *Uzyskaj dostęp do repozytorium*

Przeglądajacy
Ta rola zapewnia widoczność projektów publicznych dla wszystkich użytkowników. Domyślnie wszyscy użytkownicy są członkami tej grupy.

By default, *automatic group assignment* makes all new accounts members of this group when they join.

Role domyślne: brak

Użytkownicy
Domyślna grupa dla wszystkich użytkowników.

By default, *automatic group assignment* makes all new accounts members of this group when they join.

Role domyślne: *Użytkownik zaawansowany*

Recenzenci
Grupa dla recenzentów (patrz *Przepływy pracy tłumaczenia*).

Role domyślne: *Przejrzyj ciągi*

Menedżerowie
Grupa dla administratorów.

Role domyślne: *Administracja*

Ostrzeżenie: Never remove the predefined Weblate groups and users as this can lead to unexpected problems! If you have no use for them, you can removing all their privileges instead.

2.6.3 Dodatkowe ograniczenia dostępu

If you want to use your Weblate installation in a less public manner, i.e. allow new users on an invitational basis only, it can be done by configuring Weblate in such a way that only known users have an access to it. In order to do so, you can set `REGISTRATION_OPEN` to `False` to prevent registrations of any new users, and set `REQUIRE_LOGIN` to `/.*` to require signing in to access all the site pages. This is basically the way to lock your Weblate installation.

Podpowiedź: You can use built-in *Zaproszenie nowego użytkownika* to add new users.

2.7 Projekty tłumaczeń

2.7.1 Organizacja tłumaczeniowa

Weblate organizes translatable VCS content of project/components into a tree-like structure.

- The bottom level object is *Konfiguracja projektu*, which should hold all translations belonging together (for example translation of an application in several versions and/or accompanying documentation).
- On the level above, *Konfiguracja komponentu*, which is actually the component to translate, you define the VCS repository to use, and the mask of files to translate.
- Above *Konfiguracja komponentu* there are individual translations, handled automatically by Weblate as translation files (which match *Maska pliku* defined in *Konfiguracja komponentu*) appear in the VCS repository.

Weblate supports a wide range of translation formats (both bilingual and monolingual ones) supported by Translate Toolkit, see *Obsługiwane formaty plików*.

Informacja: You can share cloned VCS repositories using *Wewnętrzne adresy URL Weblate*. Using this feature is highly recommended when you have many components sharing the same VCS. It improves performance and decreases required disk space.

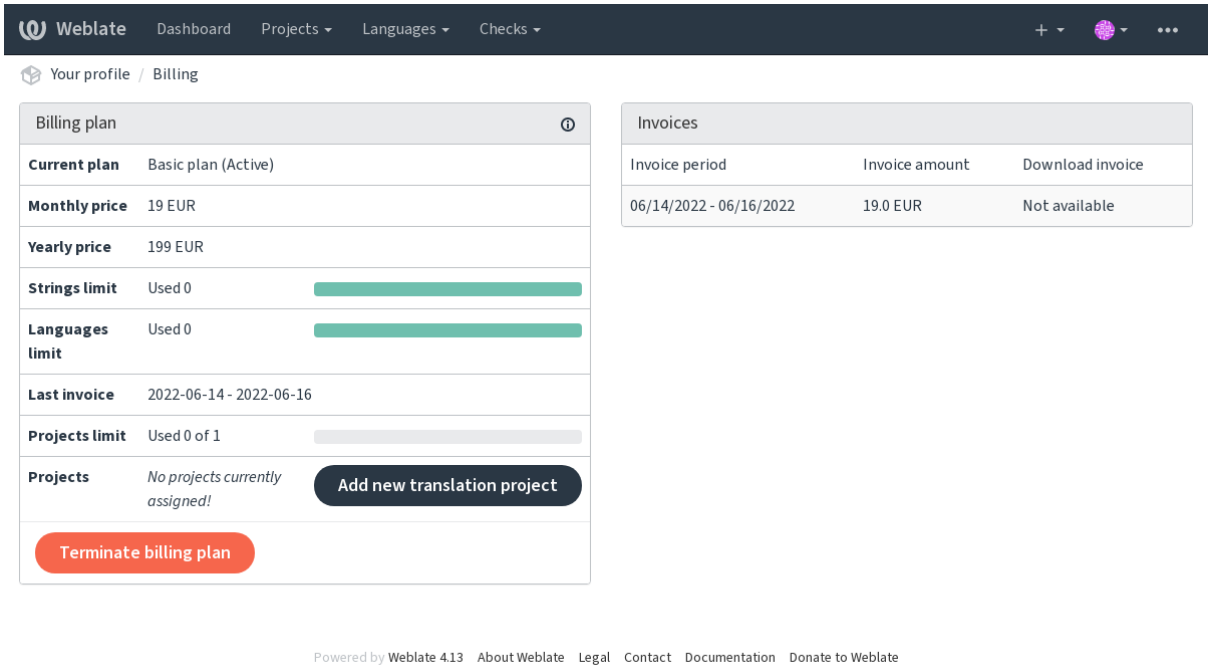
2.7.2 Dodawanie projektów i komponentów tłumaczeniowych

Zmienione w wersji 3.2: An interface for adding projects and components is included, and you no longer have to use *Interfejs administracyjny Django*.

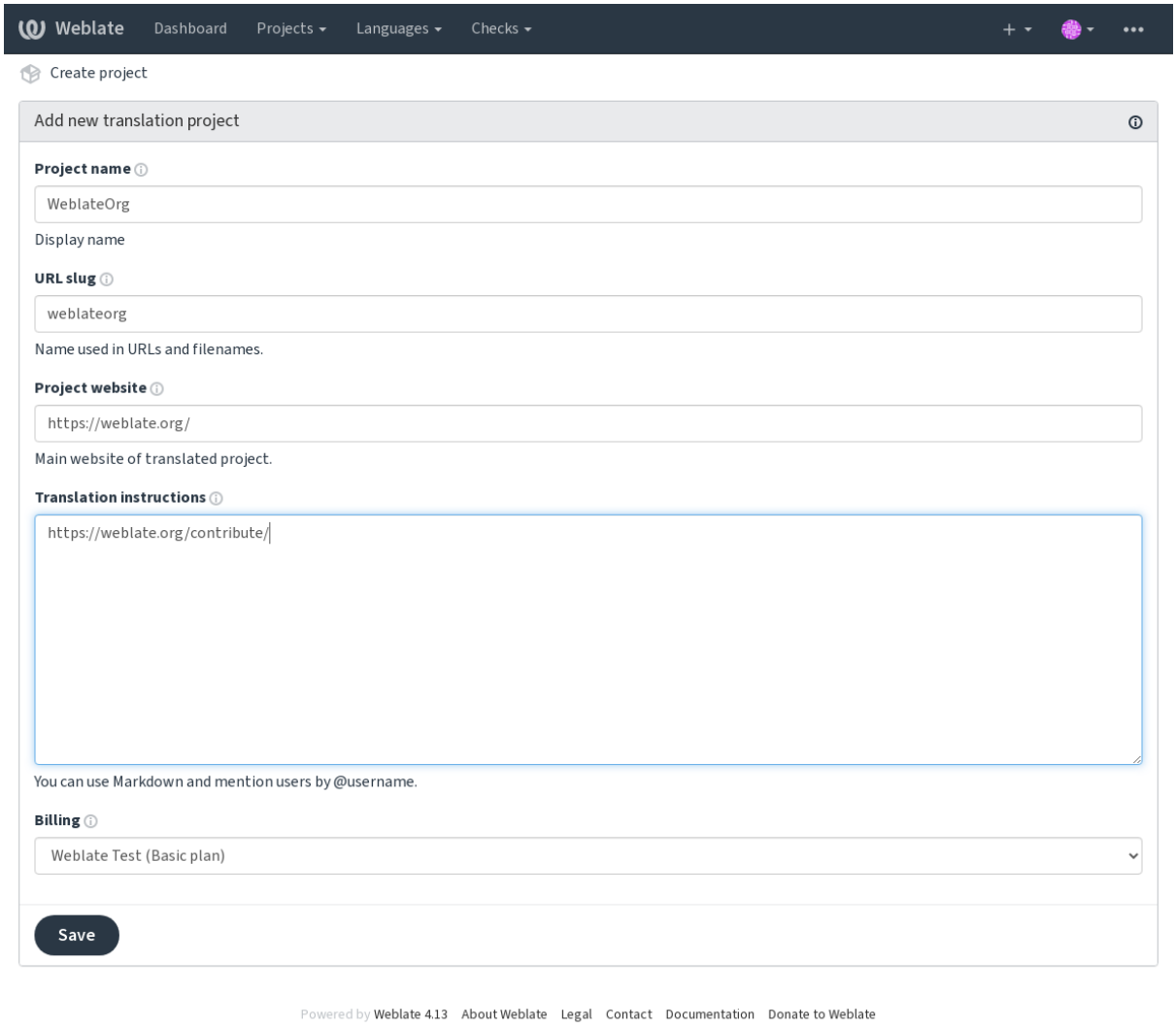
Zmienione w wersji 3.4: The process of adding components is now multi staged, with automated discovery of most parameters.

Based on your permissions, new translation projects and components can be created. It is always permitted for users with the *Add new projects* permission, and if your instance uses billing (e.g. like https://hosted.weblate.org/ see *Rozliczenia*), you can also create those based on your plans allowance from the user account that manages billing.

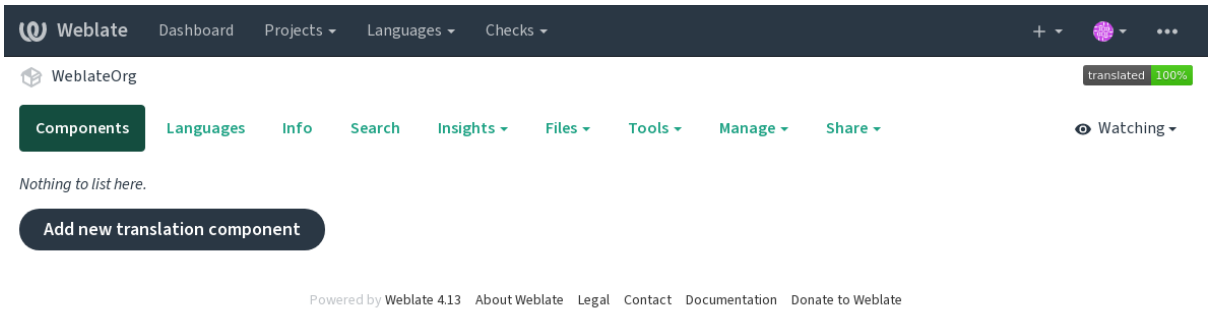
You can view your current billing plan on a separate page:



The project creation can be initiated from there, or using the menu in the navigation bar, filling in basic info about the translation project to complete addition of it:



After creating the project, you are taken directly to the project page:



Creating a new translation component can be initiated via a single click there. The process of creating a component is multi-staged and automatically detects most translation parameters. There are several approaches to creating component:

Z kontroli wersji
Creates component from remote version control repository.

Z istniejącego komponentu
Creates additional component to existing one by choosing different files.

Dodatkowa gałąź
Creates additional component to existing one, just for different branch.

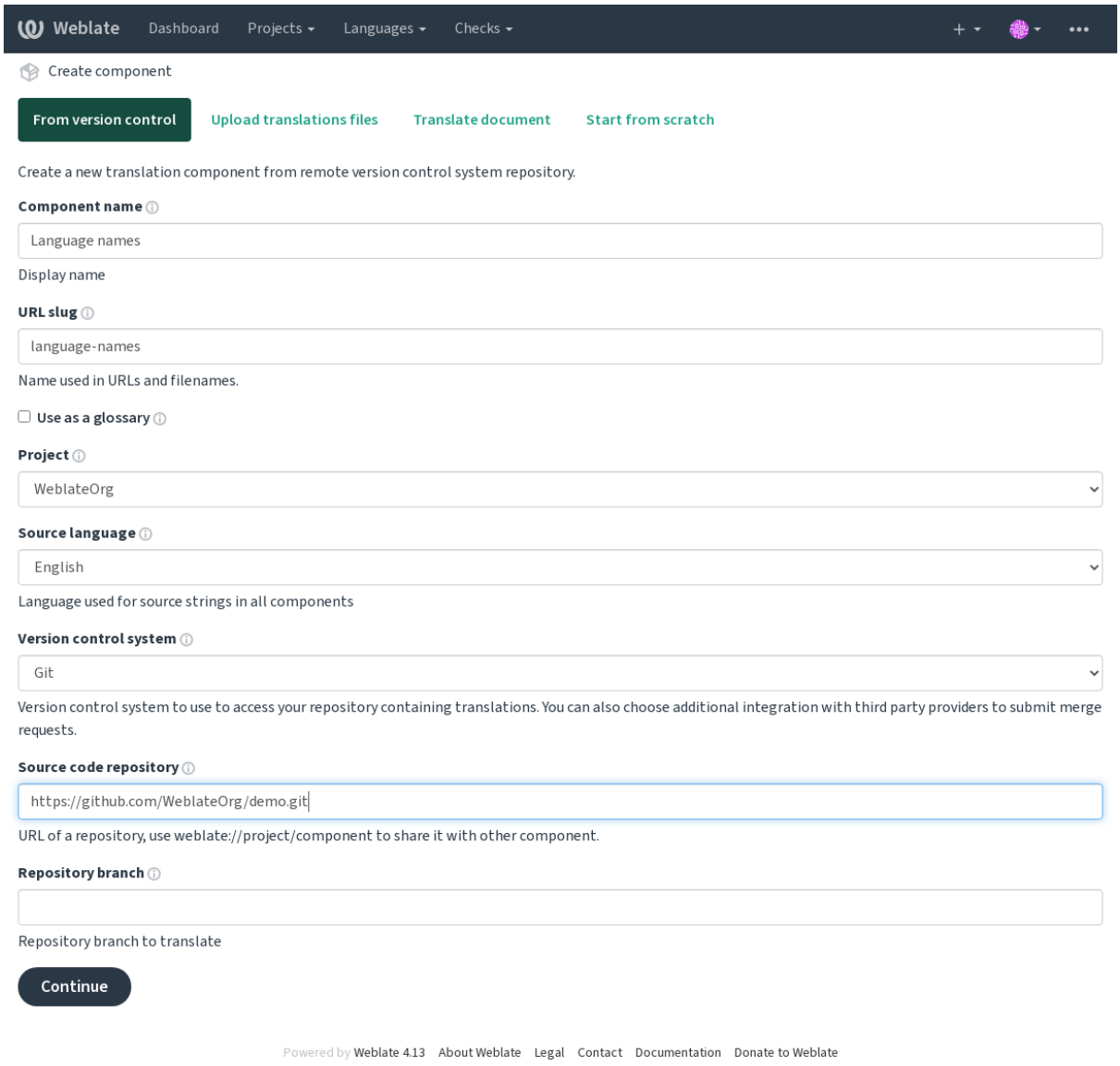
Prześlij pliki tłumaczeń
Upload translation files to Weblate in case you do not have version control or do not want to integrate it with Weblate. You can later update the content using the web interface or *REST API Weblate*.

Przetłumacz dokument
Prześlij pojedynczy dokument lub plik tłumaczenia i przetłumacz go.

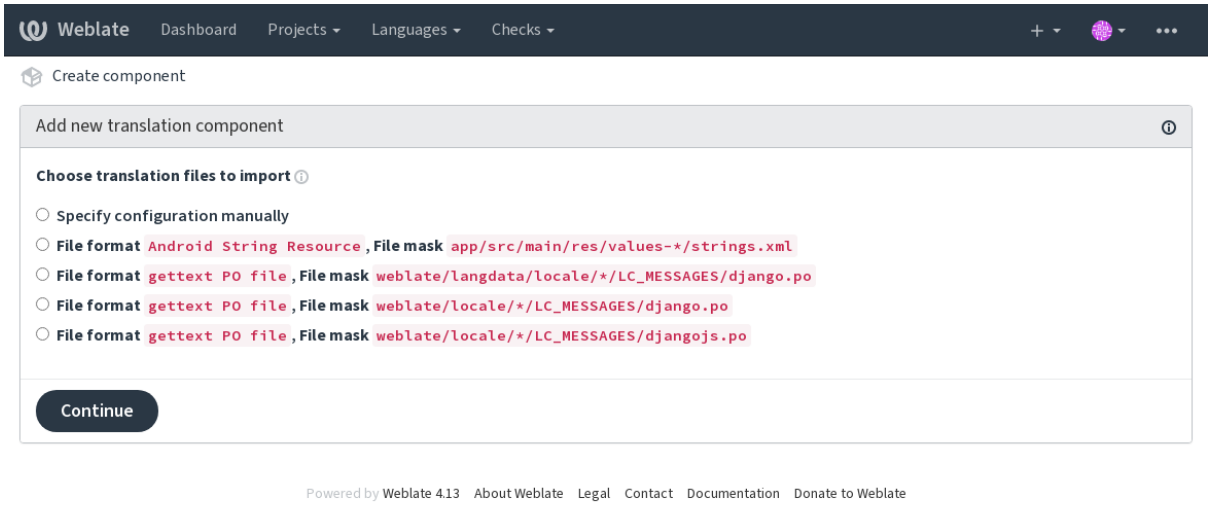
Zaczynaj od początku
Create blank translation project and add strings manually.

Once you have existing translation components, you can also easily add new ones for additional files or branches using same repository.

First you need to fill in name and repository location:



On the next page, you are presented with a list of discovered translatable resources:



As a last step, you review the translation component info and fill in optional details:

Weblate Dashboard Projects Languages Checks

Create component

Add new translation component

Project

WeblateOrg

Component name

Language names

Display name

URL slug

language-names

Name used in URLs and filenames.

Version control system

Git

Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.

Source code repository

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

URL of a repository, use weblate://project/component to share it with other component.

Repository branch

Repository branch to translate

Repository push URL

URL of a push repository, pushing is turned off if empty.

Push branch

Branch for pushing changes, leave empty to use repository branch

Repository browser

https://github.com/WeblateOrg/demo/blob/{{branch}}/{{filename}}#L{{line}}

Link to repository browser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders. You might want to strip leading directory by using {{filename|parentdir}}.

File format

gettext PO file

File mask

app/src/main/res/values-*/strings.xmlweblate/langdata/locale/*/LC_MESSAGES/django.po

Path of files to translate relative to repository root, use * instead of language code, for example: po/*.po or locale/*/LC_MESSAGES/django.po.

Monolingual base language file

app/src/main/res/values/strings.xml

Filename of translation base file, containing all strings and their source; it is recommended for monolingual translation formats.

☑ **Edit base file**

Whether users will be able to edit the base file for monolingual translations.

Intermediate language file

Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.

Template for new translations

weblate/langdata/locale/django.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Translation license

GNU General Public License v3.0 or later

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.

Language filter

^(cs|he|hu)$

Regular expression used to filter translation files when scanning for file mask.

Source language

English

Language used for source strings in all components

☐ **Use as a glossary**

You will be able to edit more options in the component settings after creating it.

Save

Powered by Weblate 4.13 About Weblate Legal Contact Documentation Donate to Weblate

Zobacz także:

Interfejs administracyjny Django, *Konfiguracja projektu*, *Konfiguracja komponentu*

2.7.3 Konfiguracja projektu

Create a translation project and then add a new component for translation in it. The project is like a shelf, in which real translations are stacked. All components in the same project share suggestions and their dictionary; the translations are also automatically propagated through all components in a single project (unless turned off in the component configuration), see *Pamięć tłumaczeniowa*.

Zobacz także:

/devel/integration

These basic attributes set up and inform translators of a project:

Nazwa projektu

Verbose project name, used to display the project name.

Końcówka adresu URL

Nazwa projektu odpowiednia dla adresów URL.

Strona internetowa projektu

URL where translators can find more info about the project.

This is a required parameter unless turned off by `WEBSITE_REQUIRED`.

Instrukcje tłumaczenia

Text describing localization process in the project, and any other information useful for translators. Markdown can be used for text formatting or inserting links.

Ustaw nagłówek „Zespół Tłumaczy”

Whether Weblate should manage the `Language-Team` header (this is a *GNU gettext* only feature right now).

Użyj wspólnej pamięci tłumaczeniowej

Whether to use shared translation memory, see *Współużytkowana pamięć tłumaczeniowa* for more details.

Domyślna wartość może być zmieniona przez `DEFAULT_SHARED_TM`.

Przyczynia się do wspólnej pamięci tłumaczeń

Whether to contribute to shared translation memory, see *Współużytkowana pamięć tłumaczeniowa* for more details.

Domyślna wartość może być zmieniona przez `DEFAULT_SHARED_TM`.

Kontrola dostępu

Configure per project access control, see *Ustawienia kontroli dostępu do projektu* for more details.

Domyślna wartość może być zmieniona przez `DEFAULT_ACCESS_CONTROL`.

Włącz przeglądy

Enable review workflow for translations, see *Dedykowani recenzenci*.

Włącz przeglądy źródeł

Enable review workflow for source strings, see *Przeglądy ciągów źródłowych*.

Zobacz także:

report-source, *Komentarze*

Włącz hooki

Whether unauthenticated *Hooki powiadomień* are to be used for this repository.

Zobacz także:

Plik języka pośredniego, *Quality gateway for the source strings*, *Formaty dwujęzyczne i jednojęzyczne*, *Definicje języków*

Aliasy językowe

Define language codes mapping when importing translations into Weblate. Use this when language codes are inconsistent in your repositories and you want to get a consistent view in Weblate or in case you want to use non-standard naming of your translation files.

The typical use case might be mapping American English to English: `en_US:en`

Multiple mappings to be separated by comma: `en_GB:en,en_US:en`

Using non standard code: `ia_FOO:ia`

Podpowiedź: The language codes are mapped when matching the translation files and the matches are case sensitive, so make sure you use the source language codes in same form as used in the filenames.

Zobacz także:

Parsowanie kodów językowych

2.7.4 Konfiguracja komponentu

A component is a grouping of something for translation. You enter a VCS repository location and file mask for which files you want translated, and Weblate automatically fetches from this VCS, and finds all matching translatable files.

Zobacz także:

/devel/integration

You can find some examples of typical configurations in the *Obsługiwane formaty plików*.

Informacja: It is recommended to keep translation components to a reasonable size - split the translation by anything that makes sense in your case (individual apps or add-ons, book chapters or websites).

Weblate easily handles translations with 10000s of strings, but it is harder to split work and coordinate among translators with such large translation components.

Should the language definition for a translation be missing, an empty definition is created and named as „cs_CZ (generated)". You should adjust the definition and report this back to the Weblate authors, so that the missing languages can be included in next release.

The component contains all important parameters for working with the VCS, and for getting translations out of it:

Nazwa komponentu

Verbose component name, used to display the component name.

Końcówka adresu URL komponentu

Nazwa komponentu odpowiednia dla adresów URL.

Projekt komponentu

Konfiguracja projektu, do którego należy komponent.

System kontroli wersji

VCS to use, see *Integracja kontroli wersji* for details.

Zobacz także:

Wypychanie zmian z Weblate

Repozytorium kodu źródłowego

VCS repository used to pull changes.

Zobacz także:

See *Dostęp do repozytoriów* for more details on specifying URLs.

Podpowiedź: This can either be a real VCS URL or `weblate://project/component` indicating that the repository should be shared with another component. See *Wewnętrzne adresy URL Weblate* for more details.

URL repozytorium dla push

Repository URL used for pushing. This setting is used only for *Git* and *Mercurial* and push support is turned off for these when this is empty.

For linked repositories, this is not used and setting from linked component applies.

Zobacz także:

See *Dostęp do repozytoriów* for more details on how to specify a repository URL and *Wypychanie zmian z Weblate* for more details on pushing changes from Weblate.

Przeglądarka repozytorium

URL of repository browser used to display source files (location of used messages). When empty, no such links will be generated. You can use *Znaczniki szablonu*.

For example on GitHub, use something like: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

In case your paths are relative to different folder (path contains `..`), you might want to strip leading directory by `parentdir` filter (see *Znaczniki szablonu*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

URL repozytorium eksportu

URL where changes made by Weblate are exported. This is important when *Ciągła lokalizacja* is not used, or when there is a need to manually merge changes. You can use *Eksporter Git* to automate this for Git repositories.

Gałąź repozytorium

Which branch to checkout from the VCS, and where to look for translations.

For linked repositories, this is not used and setting from linked component applies.

Wypchnij gałąź

Branch for pushing changes, leave empty to use *Gałąź repozytorium*.

For linked repositories, this is not used and setting from linked component applies.

Informacja: This is currently only supported for Git, GitLab and GitHub, it is ignored for other VCS integrations.

Zobacz także:

Wypychanie zmian z Weblate

Maska pliku

Mask of files to translate, including path. It should include one „*" replacing language code (see *Definicje języków* for info on how this is processed). In case your repository contains more than one translation file (e.g. more gettext domains), you need to create a component for each of them.

Na przykład `po/*.po` lub `locale/*/LC_MESSAGES/django.po`.

In case your filename contains special characters such as `[`, `]`, these need to be escaped as `[[]` or `[]]`.

Zobacz także:

Formaty dwujęzyczne i jednojęzyczne, *What does mean „There are more files for the single language (en)"?*

Jednojęzyczny bazowy plik języka

Base file containing string definitions for *Komponenty jednojęzyczne*.

Zobacz także:

Formaty dwujęzyczne i jednojęzyczne, *What does mean „There are more files for the single language (en)"?*

Edytuj plik bazowy

Whether to allow editing the base file for *Komponenty jednojęzyczne*.

Plik języka pośredniego

Intermediate language file for *Komponenty jednojęzyczne*. In most cases this is a translation file provided by developers and is used when creating actual source strings.

When set, the source strings are based on this file, but all other languages are based on *Jednojęzyczny bazowy plik języka*. In case the string is not translated into the source language, translating to other languages is prohibited. This provides *Quality gateway for the source strings*.

Zobacz także:

Quality gateway for the source strings, *Formaty dwujęzyczne i jednojęzyczne*, *What does mean „There are more files for the single language (en)"?*

Szablon dla nowych tłumaczeń

Base file used to generate new translations, e.g. `.pot` file with gettext.

Podpowiedź: In many monolingual formats Weblate starts with empty file by default. Use this in case you want to have all strings present with empty value when creating new translation.

Zobacz także:

adding-translation, *Dodawanie nowych tłumaczeń*, *Dodawanie nowego tłumaczenia*, *Formaty dwujęzyczne i jednojęzyczne*, *What does mean „There are more files for the single language (en)"?*

Format pliku

Translation file format, see also *Obsługiwane formaty plików*.

Adres do zgłaszania błędów łańcucha źródłowego

Email address used for reporting upstream bugs. This address will also receive notification about any source string comments made in Weblate.

Zezwól na propagację tłumaczenia

You can turn off propagation of translations to this component from other components within same project. This really depends on what you are translating, sometimes it's desirable to have make use of a translation more than once.

It's usually a good idea to turn this off for monolingual translations, unless you are using the same IDs across the whole project.

Default value can be changed by `DEFAULT_TRANSLATION_PROPAGATION`.

Zobacz także:

Zachowanie takich samych tłumaczeń między komponentami

Włącz sugestie

Whether translation suggestions are accepted for this component.

Głosowanie na sugestie

Turns on vote casting for suggestions, see *Głosowanie na sugestie*.

Automatyczne akceptowanie sugestii

Automatically accept voted suggestions, see *Głosowanie na sugestie*.

Flagi tłumaczeń

Customization of quality checks and other Weblate behavior, see *Dostosowywanie zachowania za pomocą flag*.

Wymuszone kontrole

List of checks which can not be ignored, see *Wymuszanie kontroli*.

Informacja: Enforcing the check does not automatically enable it, you still should enabled it using *Dostosowywanie zachowania za pomocą flag* in *Flagi tłumaczeń* or *Dodatkowe informacje o ciągach źródłowych*.

Licencja tłumaczenia

License of the translation (does not need to be the same as the source code license).

Zgoda współpracownika

Zgoda użytkownika, która musi zostać zatwierdzona, zanim użytkownik będzie mógł przetłumaczyć ten komponent.

Dodawanie nowego tłumaczenia

How to handle requests for creation of new languages. Available options:

Skontaktuj się z opiekunami
User can select desired language and the project maintainers will receive a notification about this. It is up to them to add (or not) the language to the repository.

Wskaż adres URL instrukcji tłumaczenia
User is presented a link to page which describes process of starting new translations. Use this in case more formal process is desired (for example forming a team of people before starting actual translation).

Utwórz nowy plik językowy
User can select language and Weblate automatically creates the file for it and translation can begin.

Wyłącz dodawanie nowych tłumaczeń
There will be no option for user to start new translation.

Podpowiedź: The project admins can add new translations even if it is disabled here when it is possible (either *Szablon dla nowych tłumaczeń* or the file format supports starting from an empty file).

Zobacz także:

adding-translation, *Dodawanie nowych tłumaczeń*

Zarządzaj ciągami

Nowe w wersji 4.5.

Configures whether users in Weblate will be allowed to add new strings and remove existing ones. Adjust this to match your localization workflow - how the new strings are supposed to be introduced.

For bilingual formats, the strings are typically extracted from the source code (for example by using **`xgettext`**) and adding new strings in Weblate should be disabled (they would be discarded next time you update the translation files). In Weblate you can manage strings for every translation and it does not enforce the strings in all translations to be consistent.

For monolingual formats, the strings are managed only on source language and are automatically added or removed in the translations. The strings appear in the translation files once they are translated.

Zobacz także:

Formaty dwujęzyczne i jednojęzyczne, adding-new-strings, `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Styl kodu języka

Dostosuj kod języka używany do generowania nazwy pliku dla tłumaczeń utworzonych przez Weblate.

Zobacz także:

Dodawanie nowych tłumaczeń, *Kod języka*, *Parsowanie kodów językowych*

Styl scalania

You can configure how updates from the upstream repository are handled. The actual implementation depends on VCS, see *Integracja kontroli wersji*.

Rebase
: Rebases Weblate commits on top of upstream repository on update. This provides clean history without extra merge commits.

  Rebasing can cause you trouble in case of complicated merges, so carefully consider whether or not you want to enable them.

  You might need to enable force pushing by choosing *Git z wymuszonym wypchnięciem* as *System kontroli wersji*, especially when pushing to a different branch.

Scal
: Upstream repository changes are merged into Weblate one. This setting utilizes fast-forward when possible. This is the safest way, but might produce a lot of merge commits.

Scal bez przewijania do przodu
: Upstream repository changes are merged into Weblate one with doing a merge commit every time (even when fast-forward would be possible). Every Weblate change will appear as a merge commit in Weblate repository.

Default value can be changed by `DEFAULT_MERGE_STYLE`.

Commituj, dodawaj, usuwaj, scalaj, dodawaj i scalaj żądania z wiadomościami

Message used when committing a translation, see *Znaczniki szablonu*.

Wartość domyślna może być zmieniona przez `DEFAULT_ADD_MESSAGE`, `DEFAULT_ADDON_MESSAGE`, `DEFAULT_COMMIT_MESSAGE`, `DEFAULT_DELETE_MESSAGE`, `DEFAULT_MERGE_MESSAGE`, `DEFAULT_PULL_MESSAGE`.

Przesyłaj przy commitowaniu

Whether committed changes should be automatically pushed to the upstream repository. When enabled, the push is initiated once Weblate commits changes to its underlying repository (see *Leniwe zatwierdzenia*). To actually enable pushing *Repository push URL* has to be configured as well.

Wiek zmian do scommitowania

Sets how old (in hours) changes have to be before they are committed by background task or the `commit_pending` management command. All changes in a component are committed once there is at least one change older than this period.

Default value can be changed by `COMMIT_PENDING_HOURS`.

Podpowiedź: There are other situations where pending changes might be committed, see *Leniwe zatwierdzenia*.

Błąd blokowania

Locks the component (and linked components, see *Wewnętrzne adresy URL Weblate*) upon the first failed push or merge into its upstream repository, or pull from it. This avoids adding another conflicts, which would have to be resolved manually.

The component will be automatically unlocked once there are no repository errors left.

Język źródłowy

Language used for source strings. Change this if you are translating from something else than English.

Podpowiedź: In case you are translating bilingual files from English, but want to be able to do fixes in the English translation as well, choose *English (Developer)* as a source language to avoid conflict between the name of the source language and the existing translation.

For monolingual translations, you can use intermediate translation in this case, see *Plik języka pośredniego*.

Filtr języka

Wyrażenie regularne używane do filtrowania tłumaczeń podczas skanowania w poszukiwaniu maski pliku. Może być wykorzystane do ograniczenia listy języków zarządzanych przez Weblate.

Informacja: You need to list language codes as they appear in the filename.

Kilka przykładów filtrowania:

Opis filtru	Wyrażenie regularne
Tylko wybrane języki	`^(cs|de|es)$`
Wykluczanie języków	`^(?!(it|fr)$).+$`
Filter two letter codes only	`^..$`
Wyklucz pliki niejęzykowe	`^(?!(blank)$).+$`
Uwzględnij wszystkie pliki (domyślnie)	`^[^.]+$`

Warianty wyrażenia regularnego

Regular expression used to determine the variants of a string, see variants.

Informacja: Most of the fields can be edited by project owners or administrators, in the Weblate interface.

Zobacz także:

Does Weblate support other VCSes than Git and Mercurial?, alerts

Priorytet

Komponenty o wyższym priorytecie są oferowane tłumaczom w pierwszej kolejności.

Ograniczony dostęp

By default the component is visible to anybody who has access to the project, even if the person can not perform any changes in the component. This makes it easier to keep translation consistency within the project.

Restricting access at a component, or component-list level takes over access permission to a component, regardless of project-level permissions. You will have to grant access to it explicitly. This can be done through granting access to a new user group and putting users in it, or using the default *custom* or *private* access control groups.

The default value can be changed in `DEFAULT_RESTRICTED_COMPONENT`.

Podpowiedź: This applies to project admins as well — please make sure you will not loose access to the component after toggling the status.

Udostępniony w projektach

You can choose additional projects where the component will be visible. Useful for shared libraries which you use in several projects.

Informacja: Sharing a component doesn't change its access control. It only makes it visible when browsing other projects. Users still need access to the actual component to browse or translate it.

Użyj jako słownika

Nowe w wersji 4.5.

Umożliwia używanie tego komponentu jako słownika. Możesz skonfigurować sposób wyświetlania za pomocą *Kolor słownika*.

The glossary will be accessible in all projects defined by *Udostępniony w projektach*.

It is recommended to enable *Zarządzaj ciągami* on glossaries in order to allow adding new words to them.

Zobacz także:

Słownik

Kolor słownika

Display color for a glossary used when showing word matches.

2.7.5 Znaczniki szablonu

Weblate uses simple markup language in several places where text rendering is needed. It is based on The Django template language, so it can be quite powerful.

Obecnie jest stosowany w:

- Commit message formatting, see *Konfiguracja komponentu*
- **Kilka dodatków**
  - *Wykrycie komponentów*
  - *Generator statystyk*
  - *Wykonywanie skryptów z dodatku*

There following variables are available in the component templates:

`{{ language_code }}`
Kod języka

`{{ language_name }}`
Nazwa języka

`{{ component_name }}`
Nazwa komponentu

`{{ component_slug }}`
Końcówka adresu URL komponentu

`{{ project_name }}`
Nazwa projektu

`{{ project_slug }}`
Końcówka adresu URL projektu

`{{ url }}`
Adres URL tłumaczenia

`{{ filename }}`
Nazwa pliku tłumaczenia

`{{ stats }}`
Translation stats, this has further attributes, examples below.

`{{ stats.all }}`
Całkowita liczba ciągów

`{{ stats.fuzzy }}`
Liczba ciągów wymagających przeglądu

`{{ stats.fuzzy_percent }}`
Procent ciągów wymagających przeglądu

`{{ stats.translated }}`
Liczba przetłumaczonych ciągów

`{{ stats.translated_percent }}`
Procent przetłumaczonych ciągów

`{{ stats.allchecks }}`
Number of strings with failing checks

`{{ stats.allchecks_percent }}`
Percent of strings with failing checks

`{{ author }}`
Author of current commit, available only in the commit scope.

{{ addon_name }}
Name of currently executed add-on, available only in the add-on commit message.

The following variables are available in the repository browser or editor templates:

{{branch}}
bieżąca gałąź

{{line}}
wiersz w pliku

{{filename}}
filename, you can also strip leading parts using the `parentdir` filter, for example `{{filename|parentdir}}`

You can combine them with filters:

```
{{ component|title }}
```

Możesz użyć warunków:

```
{% if stats.translated_percent > 80 %}Well translated!{% endif %}
```

There is additional tag available for replacing characters:

```
{% replace component "-" " " %}
```

You can combine it with filters:

```
{% replace component|capfirst "-" " " %}
```

There are also additional filter to manipulate with filenames:

```
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 }}
```

… i inne funkcje szablonu Django.

2.7.6 Prędkość importowania

Fetching VCS repository and importing translations to Weblate can be a lengthy process, depending on size of your translations. Here are some tips:

Optymalizacja konfiguracji

The default configuration is useful for testing and debugging Weblate, while for a production setup, you should do some adjustments. Many of them have quite a big impact on performance. Please check *Instalacja produkcyjna* for more details, especially:

- Configure Celery for executing background tasks (see *Zadania w tle korzystające z Celery*)
- *Włączanie buforowania*
- *Użyj potężnego silnika bazy danych*
- *Wyłączenie trybu debugowania*

Sprawdzanie limitów zasobów

If you are importing huge translations or repositories, you might be hit by resource limitations of your server.

- Check the amount of free memory, having translation files cached by the operating system will greatly improve performance.
- Disk operations might be bottleneck if there is a lot of strings to process—the disk is pushed by both Weblate and the database.
- Additional CPU cores might help improve performance of background tasks (see *Zadania w tle korzystające z Celery*).

Wyłączenie niepotrzebnych kontroli

Some quality checks can be quite expensive, and if not needed, can save you some time during import if omitted. See `CHECK_LIST` for info on configuration.

2.7.7 Automatyczne tworzenie komponentów

In case your project has dozen of translation files (e.g. for different gettext domains, or parts of Android apps), you might want to import them automatically. This can either be achieved from the command-line by using `import_project` or `import_json`, or by installing the *Wykrycie komponentów* add-on.

To use the add-on, you first need to create a component for one translation file (choose the one that is the least likely to be renamed or removed in future), and install the add-on on this component.

For the management commands, you need to create a project which will contain all components and then run `import_project` or `import_json`.

Zobacz także:

Polecenia zarządzania, *Wykrycie komponentów*

2.8 Definicje języków

To present different translations properly, info about language name, text direction, plural definitions and language code is needed.

2.8.1 Parsowanie kodów językowych

While parsing translations, Weblate attempts to map language code (usually the ISO 639-1 one) from the *Maska pliku* to any existing language object.

You can further adjust this mapping at project level by *Aliasy językowe*.

If no exact match can be found, an attempt will be made to best fit it into an existing language. Following steps are tried:

- Case insensitive lookups.
- Normalizing underscores and dashes.
- Wyszukiwanie wbudowanych aliasów językowych.
- Wyszukiwanie według nazwy języka.
- Ignoring the default country code for a given language—choosing `cs` instead of `cs_CZ`.

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

Podpowiedź: In case you see something unwanted as a language, you might want to adjust *Filtr języka* to ignore such file when parsing translations.

Zobacz także:

Kod języka, *Dodawanie nowych tłumaczeń*

2.8.2 Zmiana definicji języka

You can change language definitions in the languages interface (`/languages/` URL).

While editing, make sure all fields are correct (especially plurals and text direction), otherwise translators will be unable to properly edit those translations.

2.8.3 Wbudowane definicje języków

Definitions for about 600 languages are included in Weblate and the list is extended in every release. Whenever Weblate is upgraded (more specifically whenever **`weblate migrate`** is executed, see *Ogólne instrukcje dotyczące aktualizacji*) the database of languages is updated to include all language definitions shipped in Weblate.

This feature can be disable using `UPDATE_LANGUAGES`. You can also enforce updating the database to match Weblate built-in data using `setuplang`.

Zobacz także:

Rozszerzanie wbudowanych definicji językowych

2.8.4 Niejednoznaczne kody językowe i makrojęzyki

In many cases it is not a good idea to use macrolanguage code for a translation. The typical problematic case might be Kurdish language, which might be written in Arabic or Latin script, depending on actual variant. To get correct behavior in Weblate, it is recommended to use individual language codes only and avoid macrolanguages.

Zobacz także:

Macrolanguages definition, List of macrolanguages

2.8.5 Definicje języków

Each language consists of following fields:

Kod języka

Code identifying the language. Weblate prefers two letter codes as defined by ISO 639-1, but uses ISO 639-2 or ISO 639-3 codes for languages that do not have two letter code. It can also support extended codes as defined by BCP 47.

Zobacz także:

Parsowanie kodów językowych, *Dodawanie nowych tłumaczeń*

Nazwa języka

Visible name of the language. The language names included in Weblate are also being localized depending on user interface language.

Kierunek tekstu

Determines whether language is written right to left or left to right. This property is autodetected correctly for most of the languages.

Liczba mnoga

Number of plurals used in the language.

Formuła liczby mnogiej

Gettext compatible plural formula used to determine which plural form is used for given count.

Zobacz także:

Liczba mnoga, GNU gettext utilities: Plural forms, Language Plural Rules by the Unicode Consortium

2.8.6 Dodawanie nowych tłumaczeń

Zmienione w wersji 2.18: In versions prior to 2.18 the behaviour of adding new translations was file format specific.

Weblate can automatically start new translation for all of the file formats.

Some formats expect to start with an empty file and only translated strings to be included (for example *Android string resources*), while others expect to have all keys present (for example *GNU gettext*). The document-based formats (for example *Format OpenDocument*) start with a copy of the source document and all strings marked as needing editing. 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 *Plik JSON*).

When you specify *Szablon dla nowych tłumaczeń* in *Konfiguracja komponentu*, Weblate will use this file to start new translations. Any exiting translations will be removed from the file when doing so.

When *Szablon dla nowych tłumaczeń* is empty and the file format supports it, an empty file is created where new strings will be added once they are translated.

The *Styl kodu języka* allows you to customize language code used in generated filenames:

Domyślnie na podstawie formatu pliku
Dependent on file format, for most of them POSIX is used.

Styl POSIX z podkreśleniem jako separatorem
Typically used by gettext and related tools, produces language codes like `pt_BR`.

Styl POSIX z podkreśleniem jako separatorem, wraz z kodem kraju
POSIX style language code including the country code even when not necessary (for example `cs_CZ`).

Styl BCP za pomocą łącznika jako separatora
Typically used on web platforms, produces language codes like `pt-BR`.

Styl BCP za pomocą łącznika jako separatora, wraz z kodem kraju
BCP style language code including the country code even when not necessary (for example `cs-CZ`).

Styl Android
Only used in Android apps, produces language codes like `pt-rBR`.

Styl Java
Used by Java—mostly BCP with legacy codes for Chinese.

Styl Linuksa
Locales as used by Linux, uses legacy codes for Chinese and POSIX style notation.

Additionally, any mappings defined in *Aliasy językowe* are applied in reverse.

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

Zobacz także:

Kod języka, *Parsowanie kodów językowych*

2.9 Ciągła lokalizacja

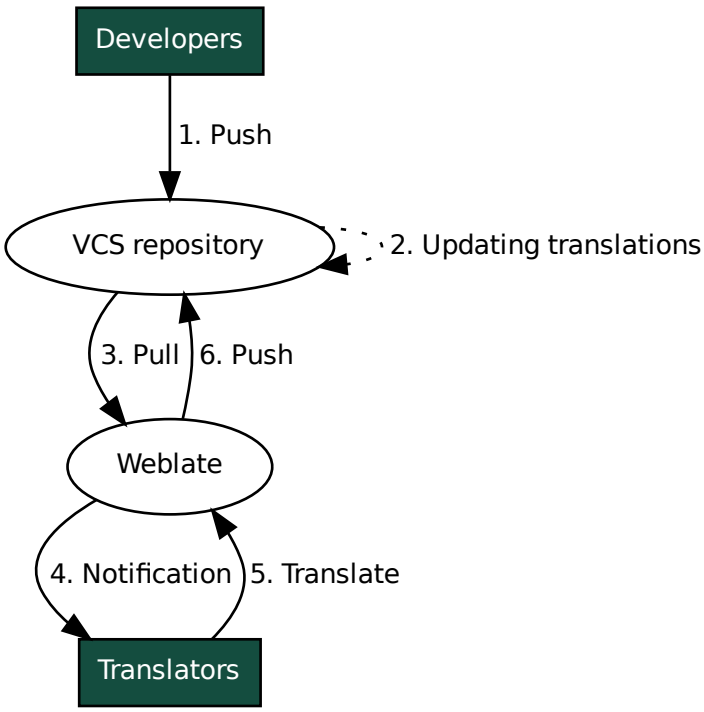
Istnieje infrastruktura, dzięki której Twoje tłumaczenie ściśle odpowiada rozwojowi. W ten sposób tłumacze mogą pracować nad tłumaczeniami przez cały czas, zamiast pracować nad ogromną ilością nowego tekstu tuż przed wydaniem.

Zobacz także:

/devel/integration describes basic ways to integrate your development with Weblate.

To jest proces:

1. Developers make changes and push them to the VCS repository.
2. Optionally the translation files are updated (this depends on the file format, see *Why does Weblate still show old translation strings when I've updated the template?*).
3. Weblate pulls changes from the VCS repository, see *Aktualizacja repozytoriów*.
4. Once Weblate detects changes in translations, translators are notified based on their subscription settings.
5. Translators submit translations using the Weblate web interface, or upload offline changes.
6. Once the translators are finished, Weblate commits the changes to the local repository (see *Leniwe zatwierdzenia*) and pushes them back if it has permissions to do so (see *Wypychanie zmian z Weblate*).



2.9.1 Aktualizacja repozytoriów

You should set up some way of updating backend repositories from their source.

- Use *Hooki powiadomień* to integrate with most of common code hosting services:
  - *Automatyczne otrzymywanie zmian z GitHub*
  - *Automatyczne otrzymywanie zmian z GitHub*
  - *Automatyczne otrzymywanie zmian z Bitbucket*
  - *Automatyczne odbieranie zmian Pagure*
  - *Automatically receiving changes from Azure Repos*
- Manually trigger update either in the repository management or using *REST API Weblate* or *Klient Weblate*
- Enable `AUTO_UPDATE` to automatically update all components on your Weblate instance
- Execute `updategit` (with selection of project or `--all` to update all)

Whenever Weblate updates the repository, the post-update addons will be triggered, see *Dodatki*.

Unikanie konfliktów scalania

The merge conflicts from Weblate arise when same file was changed both in Weblate and outside it. There are two approaches to deal with that - avoid edits outside Weblate or integrate Weblate into your updating process, so that it flushes changes prior to updating the files outside Weblate.

The first approach is easy with monolingual files - you can add new strings within Weblate and leave whole editing of the files there. For bilingual files, there is usually some kind of message extraction process to generate translatable files from the source code. In some cases this can be split into two parts - one for the extraction generates template (for example gettext POT is generated using **xgettext**) and then further process merges it into actual translations (the gettext PO files are updated using **msgmerge**). You can perform the second step within Weblate and it will make sure that all pending changes are included prior to this operation.

The second approach can be achieved by using *REST API Weblate* to force Weblate to push all pending changes and lock the translation while you are doing changes on your side.

The script for doing updates can look like this:

```
# 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
```

If you have multiple components sharing same repository, you need to lock them all separately:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Informacja: The example uses *Klient Weblate*, which needs configuration (API keys) to be able to control Weblate remotely. You can also achieve this using any HTTP client instead of wlc, e.g. curl, see *REST API Weblate*.

Zobacz także:

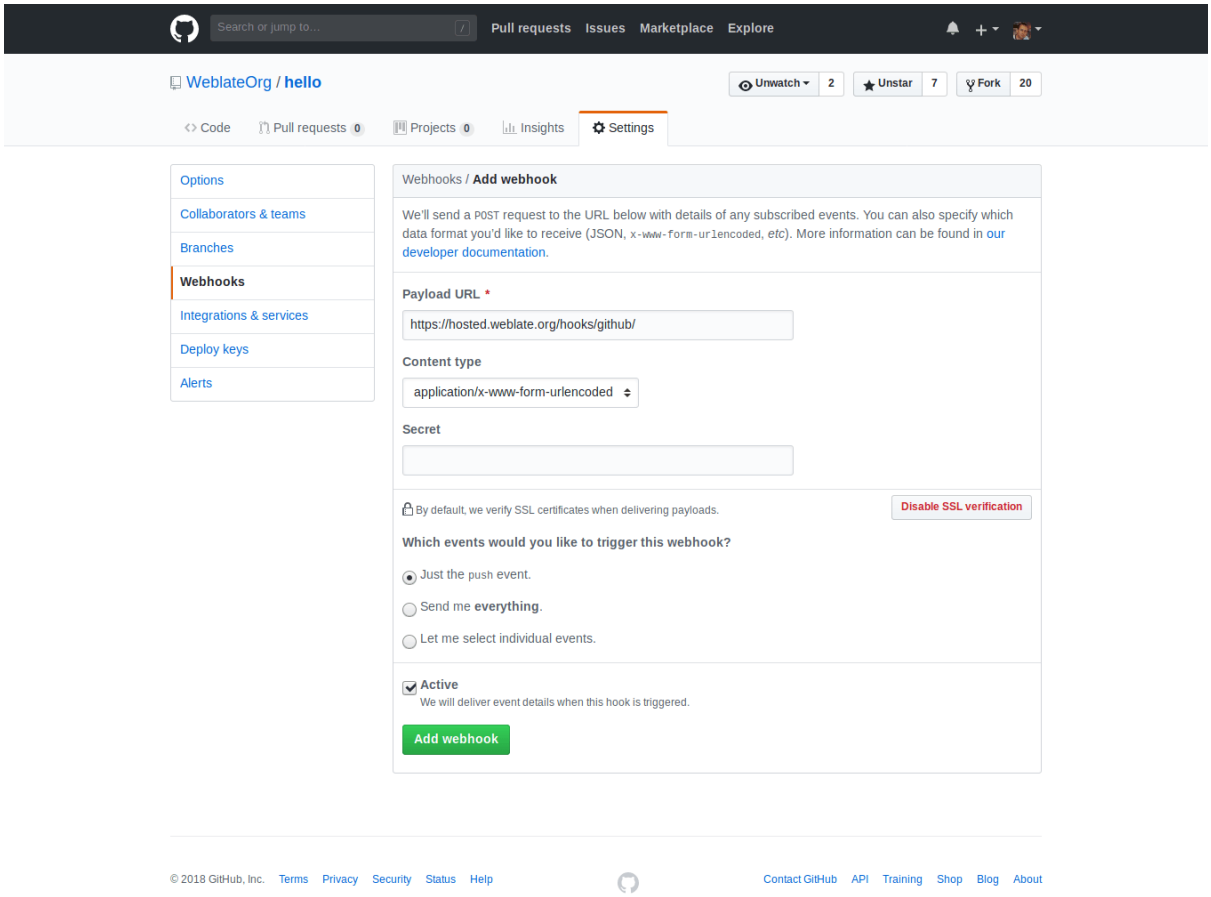
Klient Weblate

Automatyczne otrzymywanie zmian z GitHub

Weblate comes with native support for GitHub.

If you are using Hosted Weblate, the recommended approach is to install the Weblate app, that way you will get the correct setup without having to set much up. It can also be used for pushing changes back.

To receive notifications on every push to a GitHub repository, add the Weblate Webhook in the repository settings (*Webhooks*) as shown on the image below:



For the payload URL, append `/hooks/github/` to your Weblate URL, for example for the Hosted Weblate service, this is `https://hosted.weblate.org/hooks/github/`.

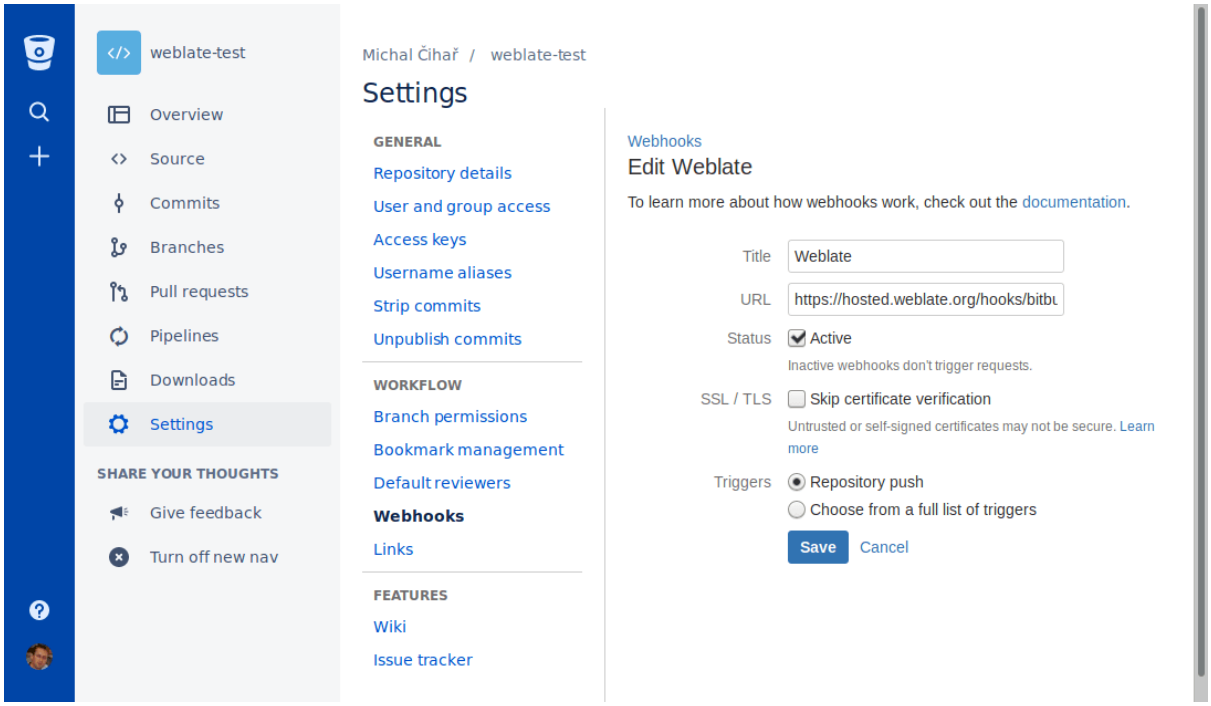
You can leave other values at default settings (Weblate can handle both content types and consumes just the *push* event).

Zobacz także:

`POST /hooks/github/`, *Accessing repositories from Hosted Weblate*

Automatyczne otrzymywanie zmian z Bitbucket

Weblate has support for Bitbucket webhooks, add a webhook which triggers upon repository push, with destination to `/hooks/bitbucket/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/bitbucket/`).



Zobacz także:

`POST /hooks/bitbucket/`, *Accessing repositories from Hosted Weblate*

Automatyczne otrzymywanie zmian z GitHub

Weblate has support for GitLab hooks, add a project webhook with destination to `/hooks/gitlab/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/gitlab/`).

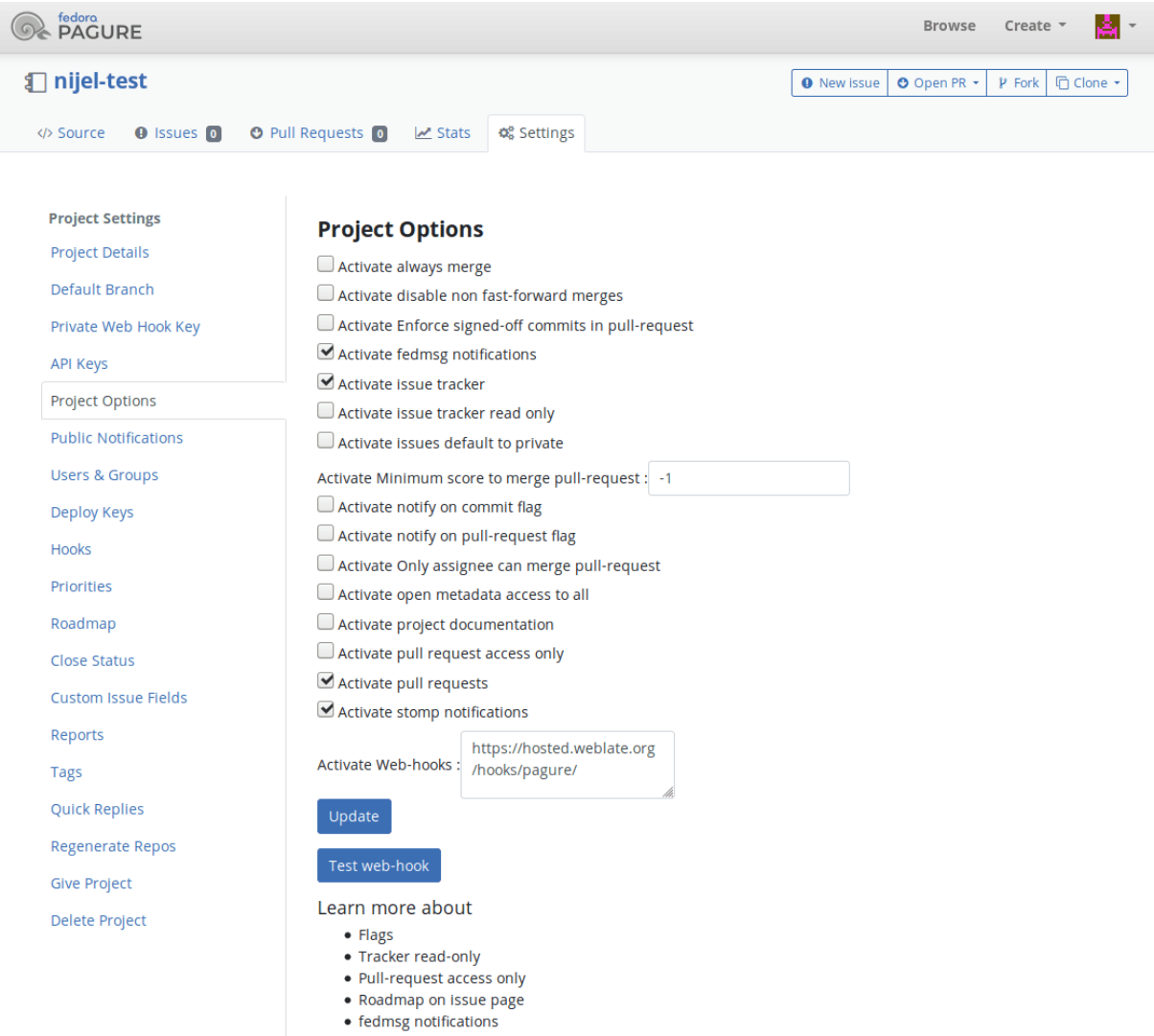
Zobacz także:

`POST /hooks/gitlab/`, *Accessing repositories from Hosted Weblate*

Automatyczne odbieranie zmian Pagure

Nowe w wersji 3.3.

Weblate has support for Pagure hooks, add a webhook with destination to `/hooks/pagure/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/pagure/`). This can be done in *Activate Web-hooks* under *Project options*:



Zobacz także:

`POST /hooks/pagure/`, *Accessing repositories from Hosted Weblate*

Automatically receiving changes from Azure Repos

Nowe w wersji 3.8.

Weblate has support for Azure Repos web hooks, add a webhook for *Code pushed* event with destination to `/hooks/azure/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/azure/`). This can be done in *Service hooks* under *Project settings*.

Zobacz także:

Web hooks in Azure DevOps manual, `POST /hooks/azure/`, *Accessing repositories from Hosted Weblate*

Automatically receiving changes from Gitea Repos

Nowe w wersji 3.9.

Weblate has support for Gitea webhooks, add a *Gitea Webhook* for *Push events* event with destination to `/hooks/gitea/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/gitea/`). This can be done in *Webhooks* under repository *Settings*.

Zobacz także:

Webhooks in Gitea manual, `POST /hooks/gitea/`, *Accessing repositories from Hosted Weblate*

Automatically receiving changes from Gitee Repos

Nowe w wersji 3.9.

Weblate has support for Gitee webhooks, add a *WebHook* for *Push* event with destination to `/hooks/gitee/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/gitee/`). This can be done in *WebHooks* under repository *Management*.

Zobacz także:

Webhooks in Gitee manual, `POST /hooks/gitee/`, *Accessing repositories from Hosted Weblate*

Automatyczna aktualizacja repozytoriów co noc

Weblate automatically fetches remote repositories nightly to improve performance when merging changes later. You can optionally turn this into doing nightly merges as well, by enabling `AUTO_UPDATE`.

2.9.2 Wypychanie zmian z Weblate

Each translation component can have a push URL set up (see *URL repozytorium dla push*), and in that case Weblate will be able to push change to the remote repository. Weblate can be also be configured to automatically push changes on every commit (this is default, see *Przesyłaj przy commitowaniu*). If you do not want changes to be pushed automatically, you can do that manually under *Repository maintenance* or using API via `wlc push`.

The push options differ based on the *Integracja kontroli wersji* used, more details are found in that chapter.

In case you do not want direct pushes by Weblate, there is support for *GitHub pull requests*, *GitLab merge requests*, *Pagure merge requests* pull requests or *Gerrit* reviews, you can activate these by choosing *GitHub*, *GitLab*, *Gerrit* or *Pagure* as *System kontroli wersji* in *Konfiguracja komponentu*.

Overall, following options are available with Git, GitHub and GitLab:

Pożądana konfiguracja	*System kontroli wersji*	*URL repozytorium dla push*	*Wypchnij gałąź*
Bez wypychania zmian	*Git*	*pusto*	*pusto*
Wypychaj bezpośrednio	*Git*	SSH URL	*pusto*
Wypychanie do oddzielnej gałęzi	*Git*	SSH URL	Nazwa gałęzi
Pull request na GitHubie z forka	*GitHub pull requests*	*pusto*	*pusto*
Pull request na GitHubie z gałęzi	*GitHub pull requests*	SSH URL[1]	Nazwa gałęzi
Żądanie scalenia GitLab z forka	*GitLab merge requests*	*pusto*	*pusto*
GitLab merge request from branch	*GitLab merge requests*	SSH URL[1]	Nazwa gałęzi
Pagure merge request from fork	*Pagure merge requests*	*pusto*	*pusto*
Pagure merge request from branch	*Pagure merge requests*	SSH URL[1]	Nazwa gałęzi

[1] Can be empty in case *Repozytorium kodu źródłowego* supports pushing.

Informacja: You can also enable automatic pushing of changes after Weblate commits, this can be done in *Przesyłaj przy commitowaniu*.

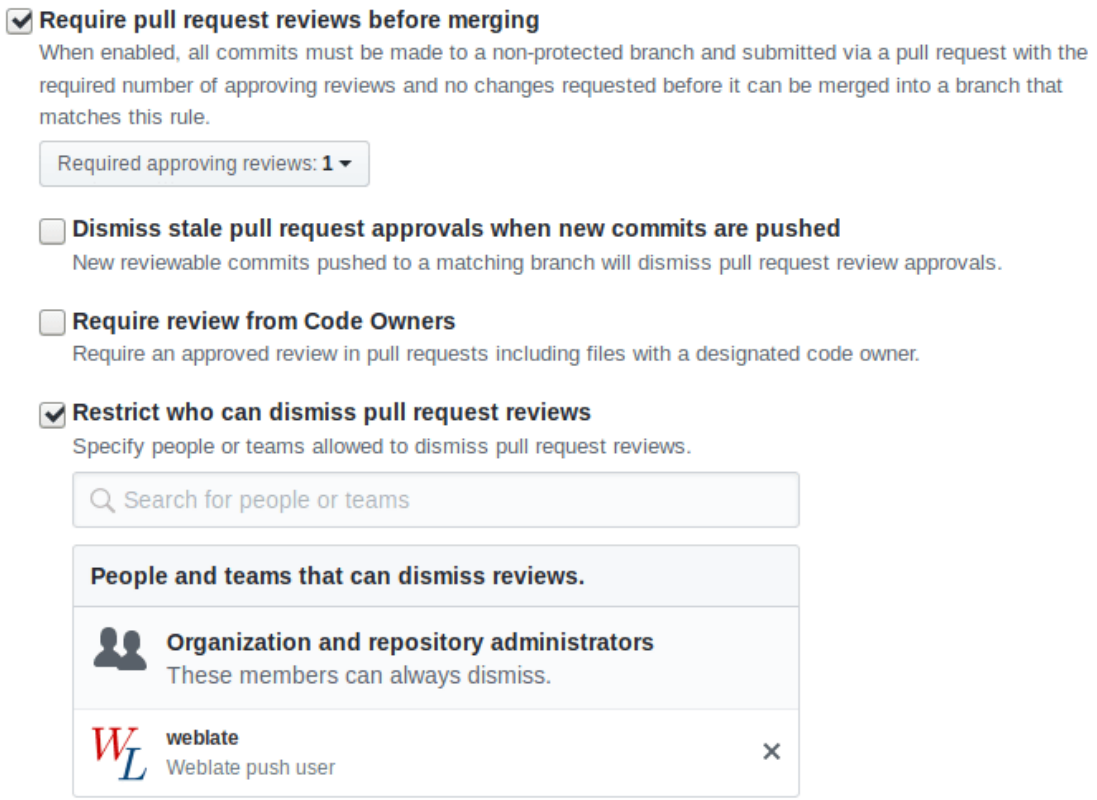
Zobacz także:

See *Dostęp do repozytoriów* for setting up SSH keys, and *Leniwe zatwierdzenia* for info about when Weblate decides to commit changes.

Chronione gałęzie

If you are using Weblate on protected branch, you can configure it to use pull requests and perform actual review on the translations (what might be problematic for languages you do not know). An alternative approach is to waive this limitation for the Weblate push user.

For example on GitHub this can be done in the repository configuration:



2.9.3 Interakcja z innymi

Weblate makes it easy to interact with others using its API.

Zobacz także:

REST API Weblate

2.9.4 Leniwe zatwierdzenia

The behaviour of Weblate is to group commits from the same author into one commit if possible. This greatly reduces the number of commits, however you might need to explicitly tell it to do the commits in case you want to get the VCS repository in sync, e.g. for merge (this is by default allowed for the *Managers* group, see *Lista uprawnień i wbudowanych ról*).

The changes in this mode are committed once any of the following conditions are fulfilled:

- Somebody else changes an already changed string.
- A merge from upstream occurs.
- An explicit commit is requested.
- Change is older than period defined as *Wiek zmian do scommitowania* on *Konfiguracja komponentu*.

Podpowiedź: Commits are created for every component. So in case you have many components you will still see lot of commits. You might utilize *Zesquashowane commity na Git* add-on in that case.

If you want to commit changes more frequently and without checking of age, you can schedule a regular task to perform a commit:

```
CELERY_BEAT_SCHEDULE = {
    # Unconditionally commit all changes every 2 minutes
    "commit": {
        "task": "weblate.trans.tasks.commit_pending",
        # Omitting 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,
    }
}
```

2.9.5 Przetwarzanie repozytorium za pomocą skryptów

The way to customize how Weblate interacts with the repository is *Dodatki*. Consult *Wykonywanie skryptów z dodatku* for info on how to execute external scripts through add-ons.

2.9.6 Zachowanie takich samych tłumaczeń między komponentami

Once you have multiple translation components, you might want to ensure that the same strings have same translation. This can be achieved at several levels.

Propagacja tłumaczeń

With *Zezwól na propagację tłumaczenia* enabled (what is the default, see *Konfiguracja komponentu*), all new translations are automatically done in all components with matching strings. Such translations are properly credited to currently translating user in all components.

Informacja: The translation propagation requires the key to be match for monolingual translation formats, so keep that in mind when creating translation keys.

Kontrola spójności

The *Niespójność* check fires whenever the strings are different. You can utilize this to review such differences manually and choose the right translation.

Tłumaczenie automatyczne

Automatic translation based on different components can be way to synchronize the translations across components. You can either trigger it manually (see *Tłumaczenie automatyczne*) or make it run automatically on repository update using add-on (see *Tłumaczenie automatyczne*).

2.10 Licencjonowanie tłumaczeń

You can specify which license translations are contributed under. This is especially important to do if translations are open to the public, to stipulate what they can be used for.

You should specify *Konfiguracja komponentu* license info. You should avoid requiring a contributor license agreement, though it is possible.

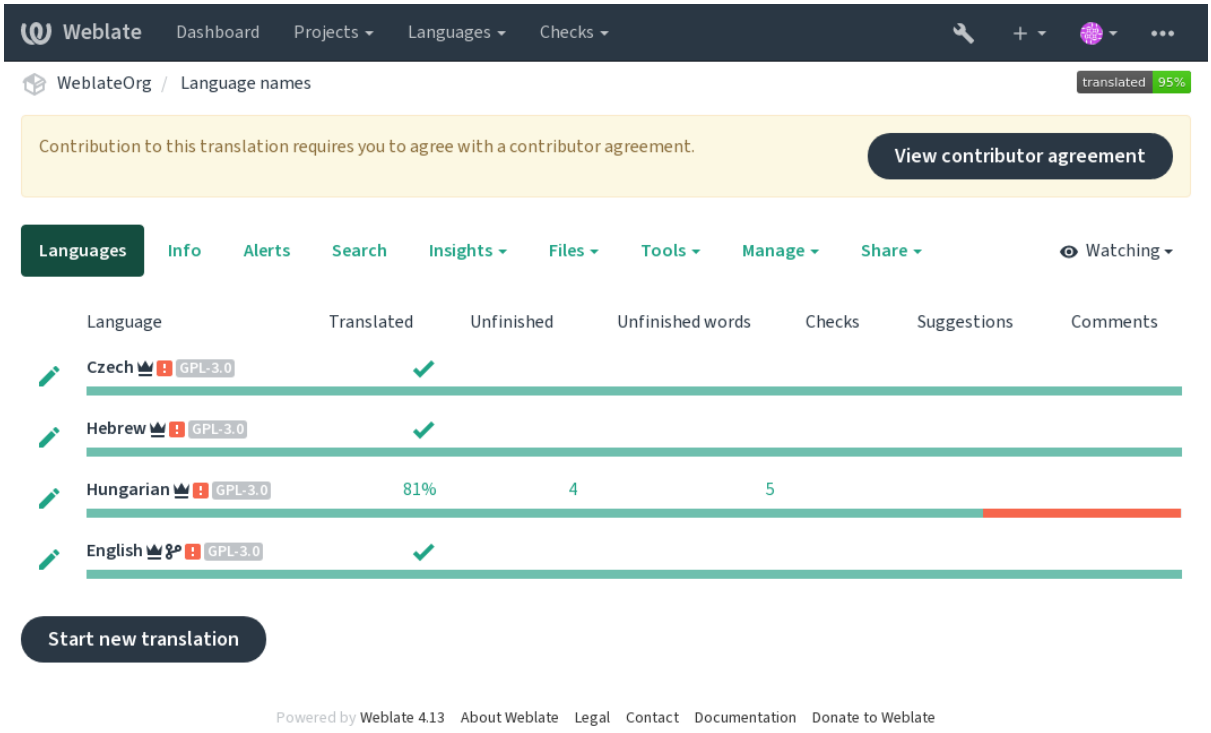
2.10.1 Informacje o licencji

Upon specifying license info (license name and URL), this info is shown in the translation info section of the respective *Konfiguracja komponentu*.

Usually this is best place to post licensing info if no explicit consent is required. If your project or translation is not libre you most probably need prior consent.

2.10.2 Zgoda współpracownika

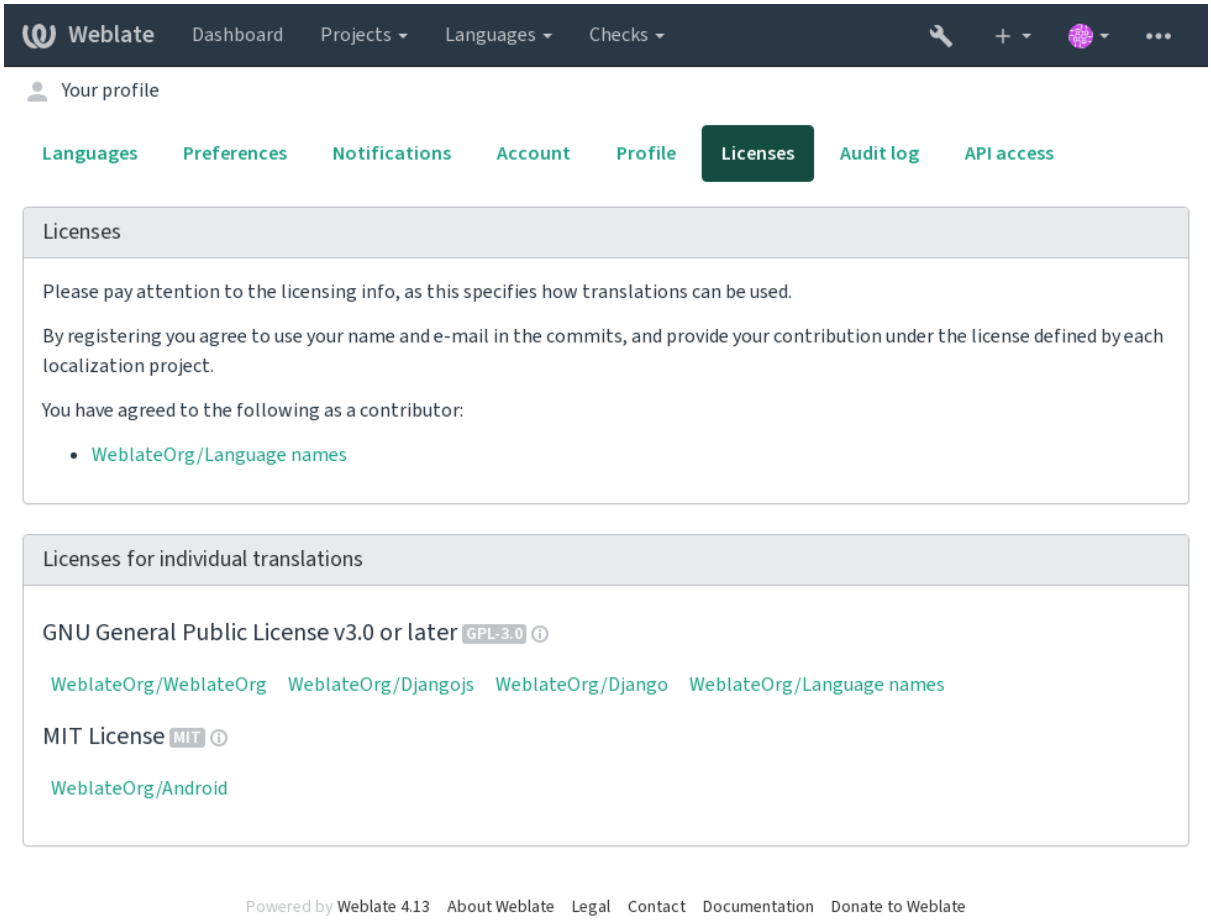
If you specify a contributor license agreement, only users who have agreed to it will be able to contribute. This is a clearly visible step when accessing the translation:



The entered text is formatted into paragraphs and external links can be included. HTML markup can not be used.

2.10.3 Licencje użytkowników

Any user can review all translation licenses of all public projects on the instance from their profile:



2.11 Proces tłumaczenia

2.11.1 Głosowanie na sugestie

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

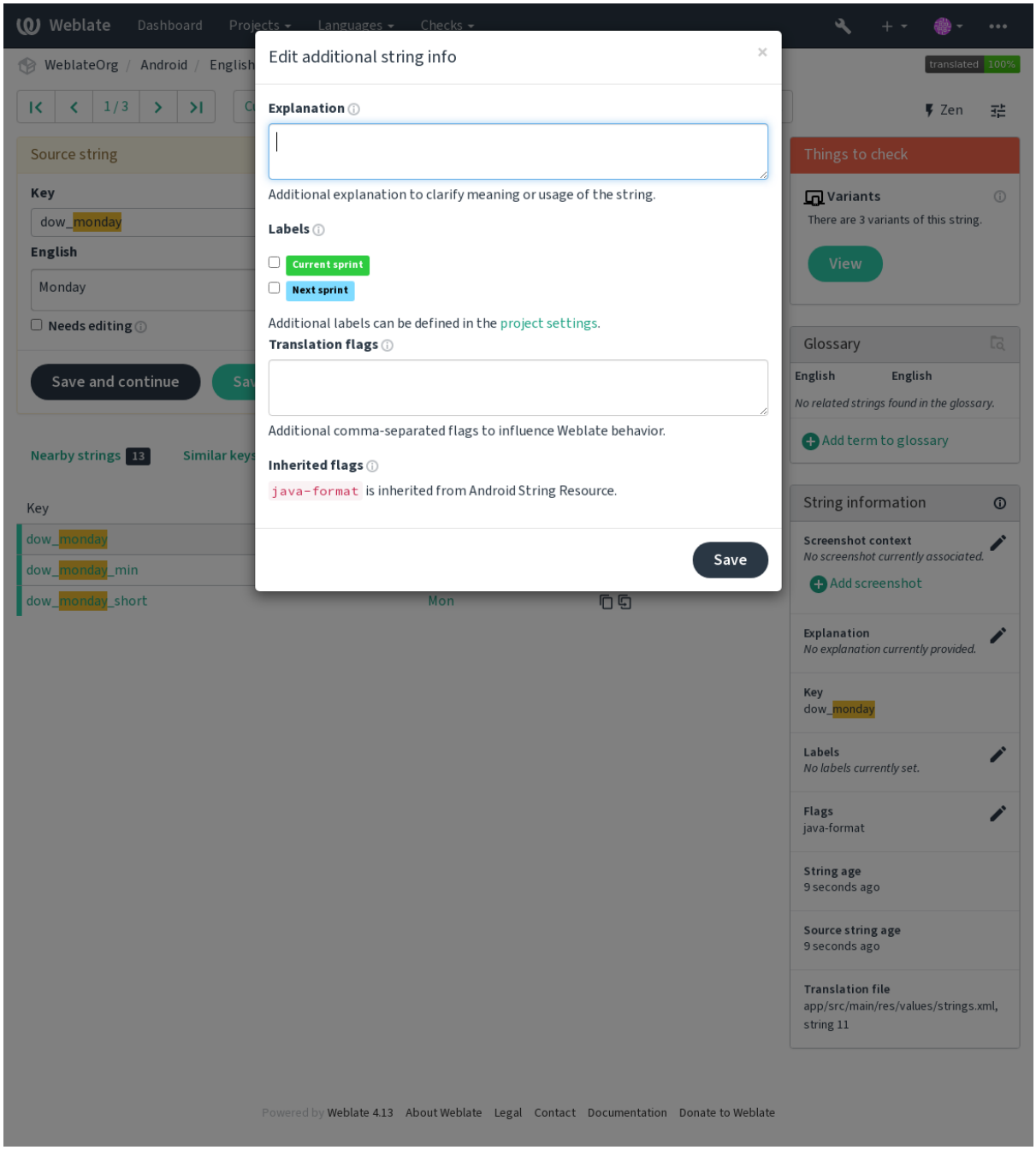
Informacja: Once automatic acceptance is set up, normal users lose the privilege to directly save translations or accept suggestions. This can be overridden with the *Edit string when suggestions are enforced permission*.

You can combine these with *access control* into one of the following setups:

- Users suggest and vote for suggestions and a limited group controls what is accepted. - Turn on voting. - Turn off automatic acceptance. - Don't let users save translations.
- Users suggest and vote for suggestions with automatic acceptance once the defined number of them agree. - Turn on voting. - Set the desired number of votes for automatic acceptance.
- Optional voting for suggestions. (Can optionally be used by users when they are unsure about a translation by making multiple suggestions.) - Only turn on voting.

2.11.2 Dodatkowe informacje o ciągach źródłowych

Enhance the translation process by adding additional info to the strings including explanations, string priorities, check flags and visual context. Some of that info may be extracted from the translation files and some may be added by editing the additional string info:



Access this directly from the translation interface by clicking the „Edit” icon next to *Screenshot context* or *Flags*.

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 169/1570 · 157

Save and continue | Save and stay | Suggest | Skip

Nearby strings 26 | Comments | Automatic suggestions | Other languages 4 | History

Context	English	Czech	Actions
	Files	Soubory	
	Automatic translation	Automatický překlad	
	Add new translation string	Add new translation string	
	Translation status	Stav překladu	
	%(count)s word	%(count)s slovo	
	Other components	Další součásti	
	Translation file	Soubor s překladem	
	Download	Stáhnout	
	Browse all translation changes	Procházet všechny změny v překladu.	
	Automatic translation takes existing translations in this project and applies them to the current component. It can be used to push translations to a different branch, to fix inconsistent translations or to translate a new component using translation memory.	Automatický překlad použije stávající překlady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí překladové paměti.	
	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.	
	You can add new translation string here, it will automatically appear in all translations.	Zde můžete přidat nový řetězec k překladu, automaticky se objeví ve všech jazycích.	
	The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.	Nahraný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nezapomeňte to povolit.	
	The uploaded file will be merged with the current translation.	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 správě, 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 aktivita	
	Last change	Poslední změna	
	Last author	Poslední autor	
Question for a mathematics-based CAPTCHA, the %s is an arithmetic problem	What is %s?	Kolik to je?	
	The string uses three dots (...) instead of an ellipsis character (…)		

Glossary

English	Czech	
machine translation	strojový překlad	WeblateOrg
project	projekt	WeblateOrg

Add term to glossary

String information

Screenshot context
No screenshot currently associated.
Add screenshot

Explanation
Help text for automatic translation tool

Labels
No labels currently set.

Flags
No flags currently set.

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

String age
2 seconds ago

Source string age
2 seconds ago

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

Powered by Weblate 4.13 About Weblate Legal Contact Documentation Donate to Weblate

Priorytetywanie ciągów

Nowe w wersji 2.0.

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

Podpowiedź: This can be used to order the flow of translation in a logical manner.

Zobacz także:

Kontrole jakości

Flagi tłumaczeń

Nowe w wersji 2.4.

Zmienione w wersji 3.3: Previously called *Quality checks flags*, it no longer configures only checks.

Customization of quality checks and other Weblate behavior, see *Dostosowywanie zachowania za pomocą flag*.

The string flags are also inherited from the *Flagi tłumaczeń* at *Konfiguracja komponentu* and flags from the translation file (see *Obsługiwane formaty plików*).

Zobacz także:

Kontrole jakości, *Dostosowywanie zachowania za pomocą flag*

Objaśnienie

Zmienione w wersji 4.1: In previous versions this has been called *Extra context*.

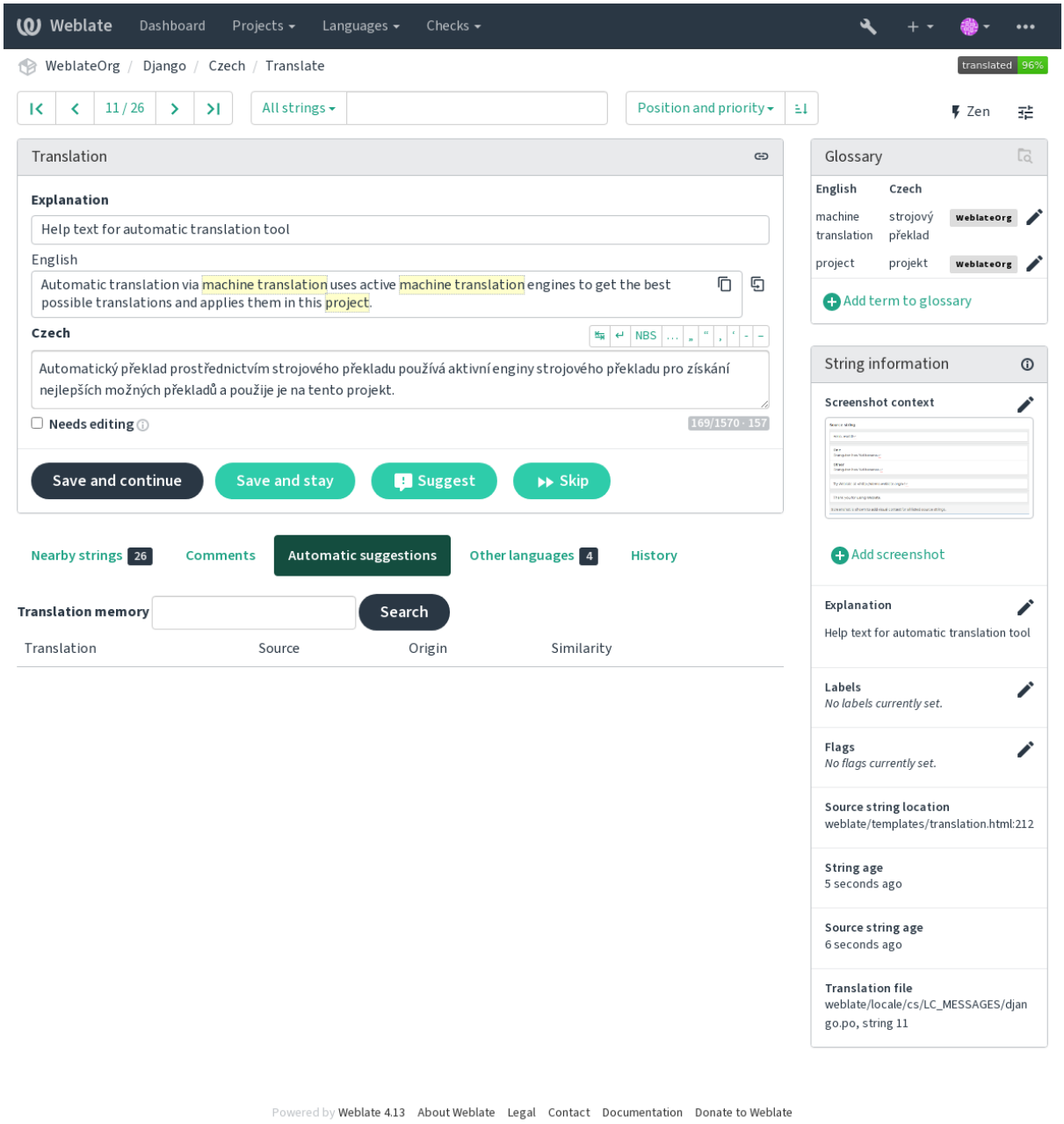
Use the explanation to clarify scope or usage of the translation. You can use Markdown to include links and other markup.

Kontekst wizualny dla ciągów

Nowe w wersji 2.9.

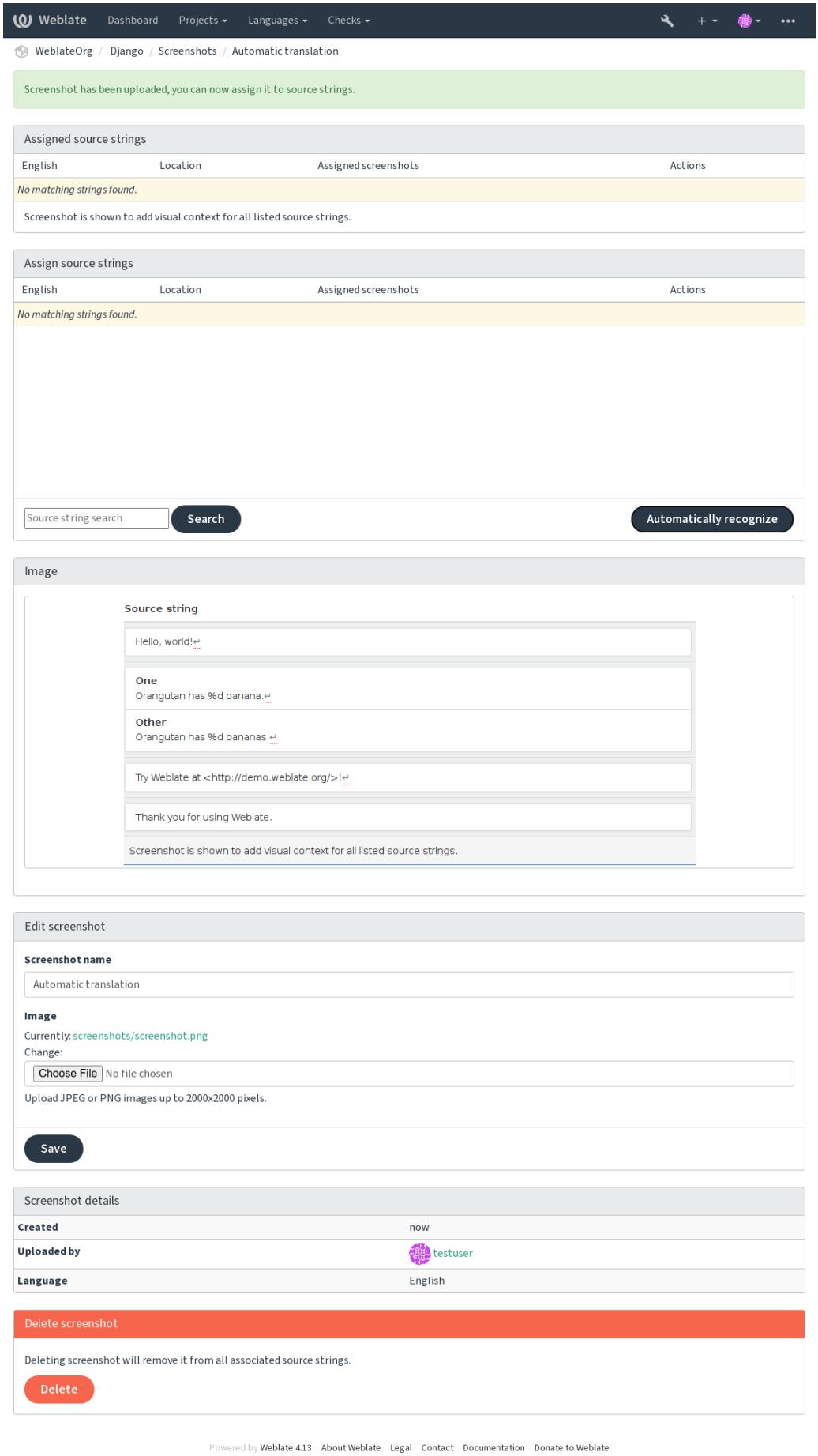
You can upload a screenshot showing a given source string in use within your program. This helps translators understand where it is used, and how it should be translated.

The uploaded screenshot is shown in the translation context sidebar:



In addition to *Dodatkowe informacje o ciągach źródłowych*, screenshots have a separate management interface under the *Tools* menu. Upload screenshots, assign them to source strings manually, or use optical character recognition to do so.

Once a screenshot is uploaded, this interface handles management and source string association:


Weblate
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Screenshots / Automatic translation
Screenshot has been uploaded, you can now assign it to source strings.
Assigned source strings
English
Location
Assigned screenshots
Actions
No matching strings found.
Screenshot is shown to add visual context for all listed source strings.
Assign source strings
English
Location
Assigned screenshots
Actions
No matching strings found.
Source string search
Search
Automatically recognize
Image
Source string
Hello, world!
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
testuser
Language
English
Delete screenshot
Deleting screenshot will remove it from all associated source strings.
Delete
Powered by Weblate 4.13
About Weblate
Legal
Contact
Documentation
Donate to Weblate

2.12 Kontrole i korekty

2.12.1 Niestandardowe automatyczne korekty

You can also implement your own automatic fixup in addition to the standard ones and include them in `AUTOFIX_LIST`.

The automatic fixes are powerful, but can also cause damage; be careful when writing one.

For example, the following automatic fixup would replace every occurrence of the string `foo` in a translation with `bar`:

```
#
# Copyright © 2012-2022 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
```

To install custom checks, provide a fully-qualified path to the Python class in the `AUTOFIX_LIST`, see *Custom quality checks, add-ons and auto-fixes*.

2.12.2 Dostosowywanie zachowania za pomocą flag

You can fine-tune the Weblate behavior by using flags. This can be done on the source string level (see *Dodatkowe informacje o ciągach źródłowych*), or in the *Konfiguracja komponentu* (*Flagi tłumaczeń*). Some file formats also allow to specify flags directly in the format (see *Obsługiwane formaty plików*).

The flags are comma-separated, the parameters are separated with colon. You can use quotes to include whitespace or special chars in the string. For example:

```
placeholders:"special:value":"other value", regex:.*
```

Both single and double quotes are accepted, special characters are being escaped using backslash:

```
placeholders:"quoted \"string\"":'single \'quoted\''
```

Here is a list of flags currently accepted:

rst-text
Treat a text as an reStructuredText document, affects *Niezmienione tłumaczenie*.

dos-eol
Uses DOS end-of-line markers instead of Unix ones (`\r\n` instead of `\n`).

read-only
The string is read-only and should not be edited in Weblate, see *Ciągi tylko do odczytu*.

priority:N
Priority of the string. Higher priority strings are presented first for translation. The default priority is 100, the higher priority a string has, the earlier it is offered for translation.

max-length:N
Limit the maximal length for a string to N characters, see *Maksymalna długość tłumaczenia*.

xml-text
Treat text as XML document, affects *Składnia XML* and *Znaczniki XML*.

font-family:NAME
Define font-family for rendering checks, see *Zarządzanie czcionkami*.

font-weight:WEIGHT
Define font-weight for rendering checks, see *Zarządzanie czcionkami*.

font-size:SIZE
Define font-size for rendering checks, see *Zarządzanie czcionkami*.

font-spacing:SPACING
Define letter spacing for rendering checks, see *Zarządzanie czcionkami*.

icu-flags:FLAGS
Zdefiniuj flagi do dostosowywania zachowania kontroli jakości *ICU MessageFormat*.

icu-tag-prefix:PREFIX
Ustaw wymagany prefiks dla tagów XML dla kontroli jakości *ICU MessageFormat*.

placeholders:NAME:NAME2:...
Placeholder strings expected in translation, see *Symbole zastępcze*.

replacements:FROM:TO:FROM2:TO2...
Replacements to perform when checking resulting text parameters (for example in *Maksymalny rozmiar tłumaczenia* or *Maksymalna długość tłumaczenia*). The typical use case for this is to expand placeables to ensure that the text fits even with long values, for example: `replacements:%s:"John Doe"`.

variants:SOURCE
Mark this string as a variant of string with matching source. See variants.

regex:REGEX
Wyrażenie regularne pasujące do tłumaczenia, zobacz: ref: *check-regex*.

forbidden
Indicates forbidden translation in a glossary, see *Zabronione tłumaczenia*.

strict-same
Make „Unchanged translation” avoid using built-in words blacklist, see *Niezmienione tłumaczenie*.

check-glossary
Włącz kontrolę jakości *Nie jest zgodny z słownikiem*.

angularjs-format
Włącz kontrolę jakości *Ciąg znaków interpolacji AngularJS*.

c-format
Włącz kontrolę jakości *Format C*.

c-sharp-format
Włącz kontrolę jakości *Format C#*.

es-format
Włącz kontrolę jakości *Literały szablonu ECMAScript*.

i18next-interpolation
Enable the *interpolacja i18next* quality check.

icu-message-format
Włącz kontrolę jakości *ICU MessageFormat*.

java-format
Włącz kontrolę jakości *Format Java*.

java-messageformat
Włącz kontrolę jakości *Format Java Message*.

javascript-format
Włącz kontrolę jakości *Format JavaScript*.

lua-format
Włącz kontrolę jakości *Format Lua*.

object-pascal-format
Włącz kontrolę jakości *Format Object Pascal*.

percent-placeholders
Włącz kontrolę jakości *Procent symboli zastępczych*.

perl-format
Włącz kontrolę jakości *Format języka Perl*.

php-format
Włącz kontrolę jakości *Format PHP*.

python-brace-format
Enable the *Format klamrowy Python* quality check.

python-format
Enable the *Format Python* quality check.

qt-format
Włącz kontrolę jakości *Format Qt*.

qt-plural-format
Włącz kontrolę jakości *Forma liczby mnogiej Qt*.

ruby-format
Włącz kontrolę jakości *Format Ruby*.

scheme-format
Włącz kontrolę jakości *Format Scheme*.

vue-format
Włącz kontrolę jakości *Formatowanie Vue I18n*.

md-text
Treat text as a Markdown document. Enable *Łącza Markdown*, *Odniesienia do Markdown*, and *Składnia Markdown* quality checks.

case-insensitive
Adjust checks behavior to be case-insensitive. Currently affects only *Symbole zastępcze* quality check.

safe-html
Włącz kontrolę jakości *Niebezpieczny HTML*.

url
The string should consist of only a URL. Enable the *URL* quality check.

ignore-all-checks
Ignoruj wszystkie kontrole jakości.

ignore-bbcode
Pomiń kontrolę jakości *Znaczniki BBCode*.

ignore-duplicate
Pomiń kontrolę jakości *Kolejne zduplikowane wyrazy*.

ignore-check-glossary
Pomiń kontrolę jakości *Nie jest zgodny z słownikiem*.

ignore-double-space
Pomiń kontrolę jakości *Podwójna spacja*.

ignore-angularjs-format
Pomiń kontrolę jakości *Ciąg znaków interpolacji AngularJS*.

ignore-c-format
Pomiń kontrolę jakości *Format C*.

ignore-c-sharp-format
Pomiń kontrolę jakości *Format C#*.

ignore-es-format
Pomiń kontrolę jakości *Literały szablonu ECMAScript*.

ignore-i18next-interpolation
Pomiń kontrolę jakości *interpolacja i18next*.

ignore-icu-message-format
Pomiń kontrolę jakości *ICU MessageFormat*.

ignore-java-format
Pomiń kontrolę jakości *Format Java*.

ignore-java-messageformat
Pomiń kontrolę jakości *Format Java Message*.

ignore-javascript-format
Pomiń kontrolę jakości *Format JavaScript*.

ignore-lua-format
Pomiń kontrolę jakości *Format Lua*.

ignore-object-pascal-format
Pomiń kontrolę jakości *Format Object Pascal*.

ignore-percent-placeholders
Pomiń kontrolę jakości *Procent symboli zastępczych*.

ignore-perl-format
Pomiń kontrolę jakości *Format języka Perl*.

ignore-php-format
Pomiń kontrolę jakości *Format PHP*.

ignore-python-brace-format
Pomiń kontrolę jakości *Format klamrowy Python*.

ignore-python-format
Pomiń kontrolę jakości *Format Python*.

ignore-qt-format
Pomiń kontrolę jakości *Format Qt*.

ignore-qt-plural-format
Pomiń kontrolę jakości *Forma liczby mnogiej Qt*.

ignore-ruby-format
Pomiń kontrolę jakości *Format Ruby*.

ignore-scheme-format
Pomiń kontrolę jakości *Format Scheme*.

ignoruj-vue-format
Pomiń kontrolę jakości *Formatowanie Vue I18n*.

ignore-translated
Pomiń kontrolę jakości *Zostało przetłumaczone*.

ignore-inconsistent
Pomiń kontrolę jakości *Niespójność*.

ignore-kashida
Pomiń kontrolę jakości *Użyto litery Kashida*.

ignore-md-link
Pomiń kontrolę jakości *Łącza Markdown*.

ignore-md-reflink
Pomiń kontrolę jakości *Odniesienia do Markdown*.

ignore-md-syntax
Pomiń kontrolę jakości *Składnia Markdown*.

ignore-max-length
Pomiń kontrolę jakości *Maksymalna długość tłumaczenia*.

ignore-max-size
Pomiń kontrolę jakości *Maksymalny rozmiar tłumaczenia*.

ignore-escaped-newline
Pomiń kontrolę jakości *Niedopasowane \n*.

ignore-end-colon
Pomiń kontrolę jakości *Niedopasowany dwukropek*.

ignore-end-ellipsis
Pomiń kontrolę jakości *Niedopasowane wielokropki*.

ignore-end-exclamation
Pomiń kontrolę jakości *Niedopasowany wykrzyknik*.

ignore-end-stop
Pomiń kontrolę jakości *Niedopasowana kropka*.

ignore-end-question
Pomiń kontrolę jakości *Niedopasowany znak zapytania*.

ignore-end-semicolon
Pomiń kontrolę jakości *Niedopasowany średnik*.

ignore-newline-count
Pomiń kontrolę jakości *Niedopasowanie podziałów wierszy*.

ignore-plurals
Pomiń kontrolę jakości *Brakująca forma liczby mnogiej*.

ignore-placeholders
Pomiń kontrolę jakości *Symbole zastępcze*.

ignore-punctuation-spacing
Pomiń kontrolę jakości *Odstępy między czcionkami*.

ignore-regex
Pomiń kontrolę jakości *Wyrażenie regularne*.

ignore-same-plurals
Pomiń kontrolę jakości *Te same liczby mnogie*.

ignore-begin-newline
Pomiń kontrolę jakości *Początek od nowej linii*.

ignore-begin-space
Pomiń kontrolę jakości *Spacje początkowe*.

ignore-end-newline
Pomiń kontrolę jakości *Zakończenie nową linią*.

ignore-end-space
Pomiń kontrolę jakości *Końcowa spacja*.

ignore-same
Pomiń kontrolę jakości *Niezmienione tłumaczenie*.

ignore-safe-html
Pomiń kontrolę jakości *Niebezpieczny HTML*.

ignore-url
Pomiń kontrolę jakości *URL*.

ignore-xml-tags
Pomiń kontrolę jakości *Znaczniki XML*.

ignore-xml-invalid
Pomiń kontrolę jakości *Składnia XML*.

ignore-zero-width-space
Pomiń kontrolę jakości *Znak spacji o zerowej szerokości*.

ignore-ellipsis
Pomiń kontrolę jakości *Wielokropek*.

ignore-icu-message-format-syntax
Pomiń kontrolę jakości *Składnia ICU MessageFormat*.

ignore-long-untranslated
Pomiń kontrolę jakości *Długo nieprzetłumaczone*.

ignore-multiple-failures
Pomiń kontrolę jakości *Wiele nieudanych kontroli*.

ignore-unnamed-format
Pomiń kontrolę jakości *Wiele nienazwanych zmiennych*.

ignore-optional-plural
Pomiń kontrolę jakości *Niespluralizowane*.

Informacja: Generally the rule is named `ignore-*` for any check, using its identifier, so you can use this even for your custom checks.

These flags are understood both in *Konfiguracja komponentu* settings, per source string settings and in the translation file itself (for example in GNU gettext).

2.12.3 Wymuszanie kontroli

Nowe w wersji 3.11.

You can configure a list of checks which can not be ignored by setting *Wymuszone kontrole* in *Konfiguracja komponentu*. Each listed check can not be dismissed in the user interface and any string failing this check is marked as *Needs editing* (see *Stan tłumaczenia*).

2.12.4 Zarządzanie czcionkami

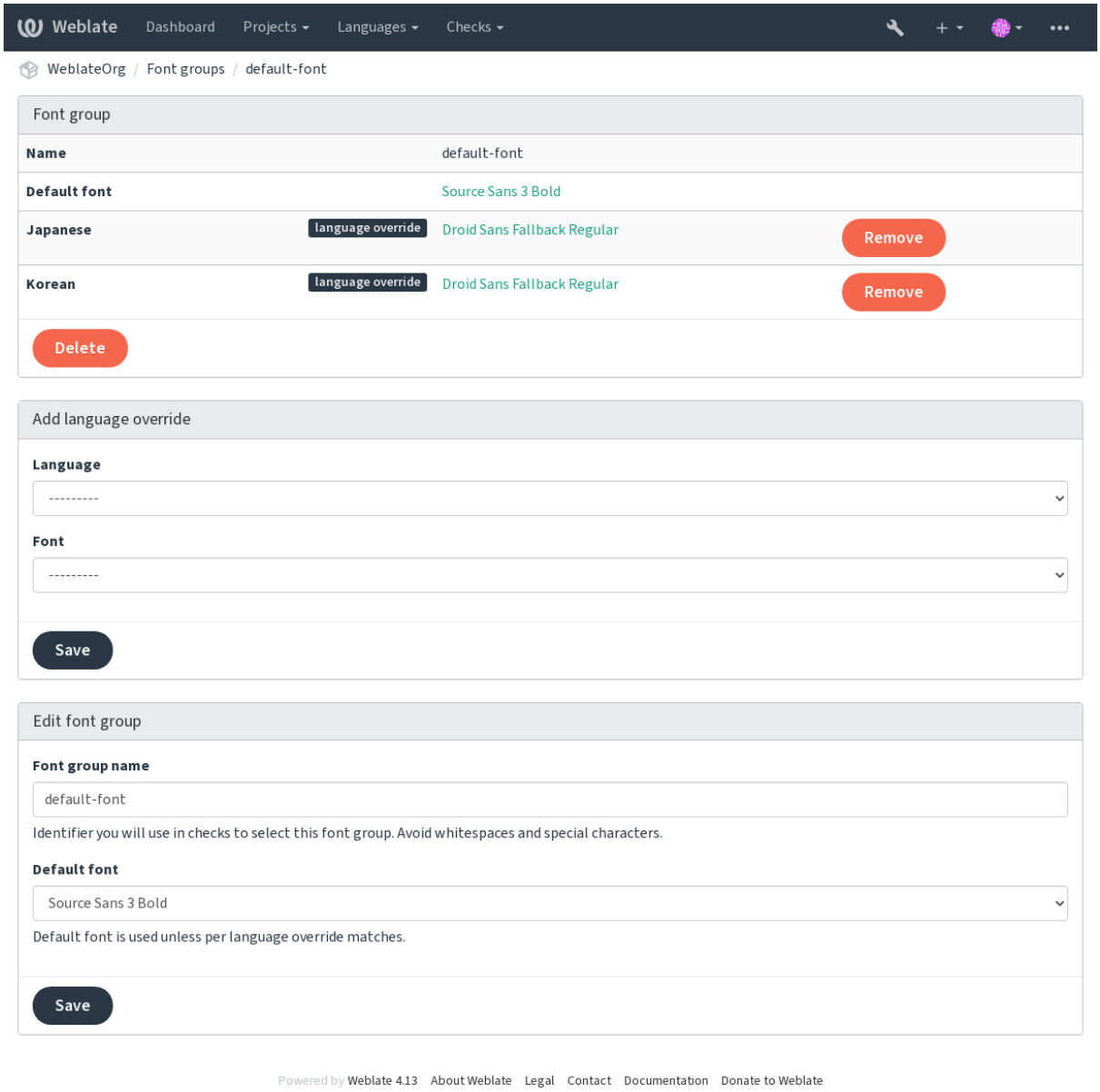
Nowe w wersji 3.7.

Podpowiedź: Fonts uploaded into Weblate are used purely for purposes of the *Maksymalny rozmiar tłumaczenia* check, they do not have an effect in Weblate user interface.

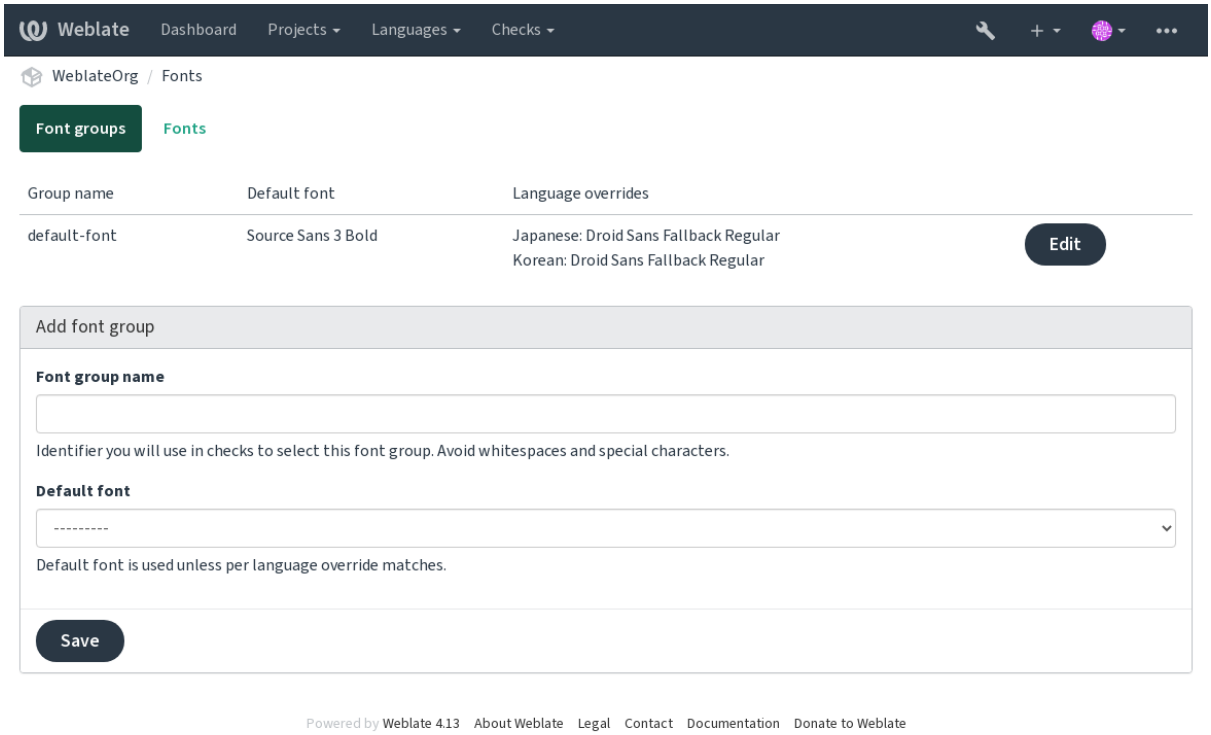
The *Maksymalny rozmiar tłumaczenia* check used to calculate dimensions of the rendered text needs font to be loaded into Weblate and selected using a translation flag (see *Dostosowywanie zachowania za pomocą flag*).

Weblate font management tool in *Fonts* under the *Manage* menu of your translation project provides interface to upload and manage fonts. TrueType or OpenType fonts can be uploaded, set up font-groups and use those in the check.

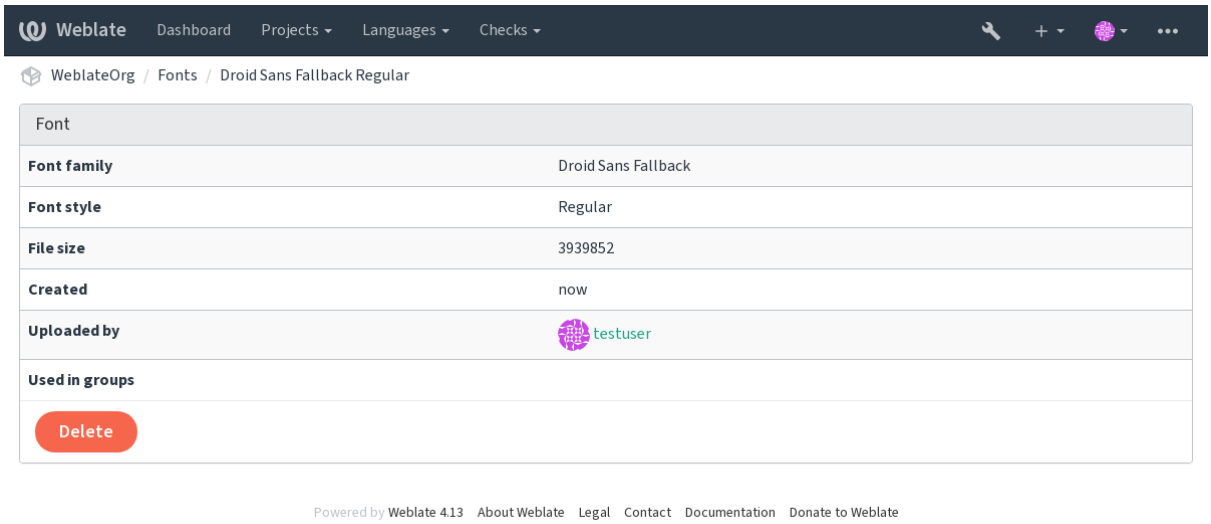
The font-groups allow you to define different fonts for different languages, which is typically needed for non-latin languages:



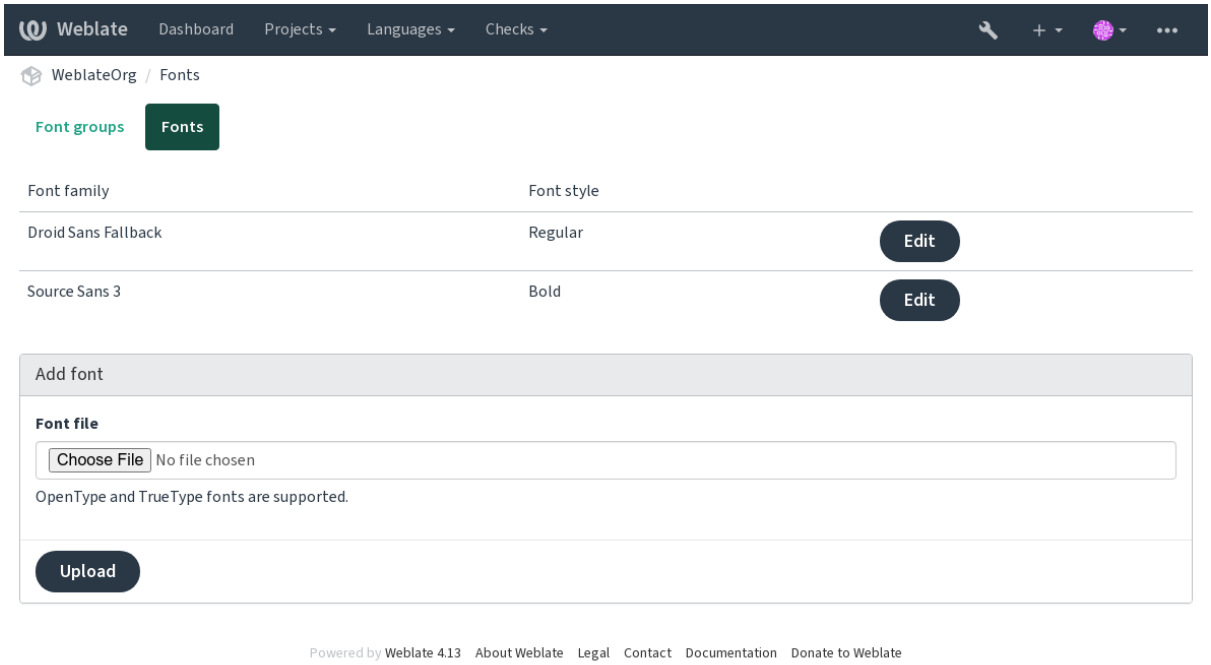
The font-groups are identified by name, which can not contain whitespace or special characters, so that it can be easily used in the check definition:



Font-family and style is automatically recognized after uploading them:



You can have a number of fonts loaded into Weblate:



To use the fonts for checking the string length, pass it the appropriate flags (see *Dostosowywanie zachowania za pomocą flag*). You will probably need the following ones:

`max-size:500`
Określa maksymalną szerokość w pikselach.

`font-family:ubuntu`
Defines font group to use by specifying its identifier.

`font-size:22`
Określa rozmiar czcionki w pikselach.

2.12.5 Pisanie własnych kontroli

A wide range of quality checks are built-in, (see *Kontrole jakości*), though they might not cover everything you want to check. The list of performed checks can be adjusted using `CHECK_LIST`, and you can also add custom checks.

1. Podklasa *weblate.checks.Check*
2. Ustaw kilka atrybutów.
3. Implement either the `check` (if you want to deal with plurals in your code) or the `check_single` method (which does it for you).

Kilka przykładów:

To install custom checks, provide a fully-qualified path to the Python class in the `CHECK_LIST`, see *Custom quality checks, add-ons and auto-fixes*.

Checking translation text does not contain „foo”

This is a pretty simple check which just checks whether the translation is missing the string „foo”.

```
#
# Copyright © 2012-2022 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/>.
#
"""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
```

Checking that Czech translation text plurals differ

Check using language info to verify the two plural forms in Czech language are not same.

```
#
# Copyright © 2012-2022 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
```

(ciąg dalszy na następnej stronie)

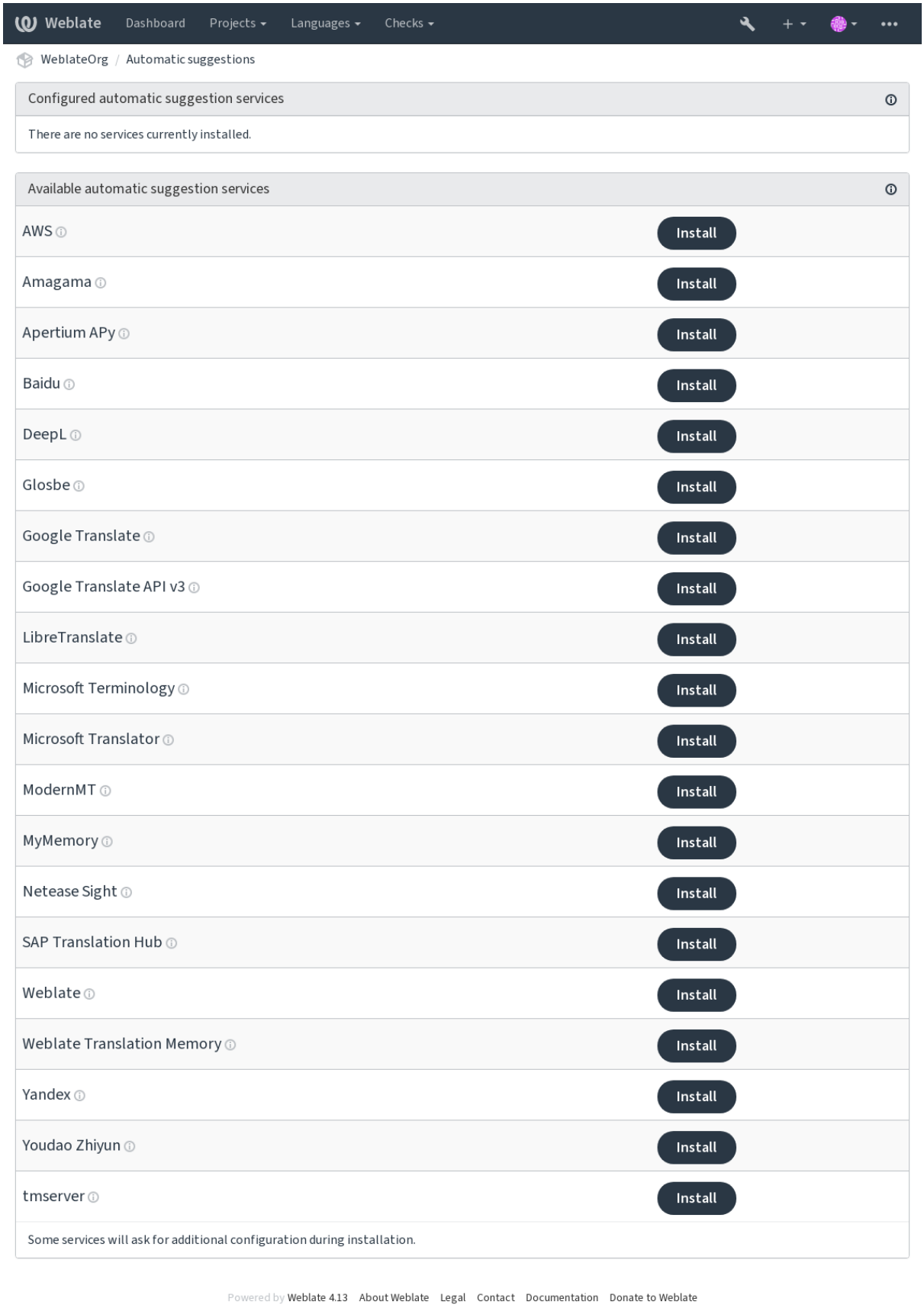
(kontynuacja poprzedniej strony)

```
# 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


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 Konfigurowanie automatycznych sugestii

Zmienione w wersji 4.13: Prior to Weblate 4.13, the services were configured in the *Konfiguracja*.

The support for several machine translation and translation memory services is built-in. Each service can be turned on by the administrator for whole site or at the project settings:



Informacja: They come subject to their terms of use, so ensure you are allowed to use them how you want.

Usługi tłumaczą z języka źródłowego zgodnie z konfiguracją w :ref:»component», patrz :ref:»component-source_language».

Zobacz także:

Automatyczne sugestie

2.13.1 Amagama

ID usługi
`amagama`

Konfiguracja
Ta usługa nie ma konfiguracji.

Special installation of *tmserver* run by the authors of Virtaal.

Zobacz także:

Installing amaGama, Amagama, amaGama Translation Memory

2.13.2 Apertium APy

ID usługi
`apertium-apy`

Konfiguracja

`url`	URL API	

A libre software machine translation platform providing translations to a limited set of languages.

The recommended way to use Apertium is to run your own Apertium-APy server.

Zobacz także:

Apertium website, Apertium APy documentation

2.13.3 AWS

Nowe w wersji 3.1.

ID usługi
`aws`

Konfiguracja

`key`	ID klucza dostępu	
`secret`	Tajny klucz API	
`region`	Nazwa regionu	

Amazon Translate is a neural machine translation service for translating text to and from English across a breadth of supported languages.

Zobacz także:

Amazon Translate Documentation

2.13.4 Baidu

Nowe w wersji 3.2.

ID usługi
`baidu`

Konfiguracja

`key`	ID klienta	
`secret`	Tajny klucz klienta	

Machine translation service provided by Baidu.

This service uses an API and you need to obtain an ID and API key from Baidu to use it.

Zobacz także:

Baidu Translate API

2.13.5 DeepL

Nowe w wersji 2.20.

ID usługi
`deepl`

Konfiguracja

`url`	URL API	
`key`	Klucz API	

DeepL is paid service providing good machine translation for a few languages. You need to purchase *DeepL API* subscription or you can use legacy *DeepL Pro (classic)* plan.

API URL to use with the DeepL service. At the time of writing, there is the v1 API as well as a free and a paid version of the v2 API.

`https://api.deepl.com/v2/` (default in Weblate)
Is meant for API usage on the paid plan, and the subscription is usage-based.

`https://api-free.deepl.com/v2/`
Is meant for API usage on the free plan, and the subscription is usage-based.

`https://api.deepl.com/v1/`
Is meant for CAT tools and is usable with a per-user subscription.

Previously Weblate was classified as a CAT tool by DeepL, so it was supposed to use the v1 API, but now is supposed to use the v2 API. Therefore it defaults to v2, and you can change it to v1 in case you have an existing CAT subscription and want Weblate to use that.

The easiest way to find out which one to use is to open an URL like the following in your browser:

https://api.deepl.com/v2/translate?text=Hello&target_lang=FR&auth_key=XXX

Replace the XXX with your auth_key. If you receive a JSON object which contains „Bonjour”, you have the correct URL; if not, try the other three.

Zobacz także:

DeepL website, DeepL pricing, DeepL API documentation

2.13.6 Glosbe

ID usługi
`glosbe`

Konfiguracja
Ta usługa nie ma konfiguracji.

Free dictionary and translation memory for almost every living language.

The API is gratis to use, but usage of the translations is subject to the license of the used data source. There is a limit of calls that may be done from one IP in a set period of time, to prevent abuse.

Zobacz także:

Glosbe website

2.13.7 Google Translate

ID usługi
`google-translate`

Konfiguracja

`key`	Klucz API	

Machine translation service provided by Google.

This service uses the Google Translation API, and you need to obtain an API key and turn on billing in the Google API console.

Zobacz także:

Dokumentacja tłumacza Google

2.13.8 Google Translate API v3

ID usługi
`google-translate-api-v3`

Konfiguracja

`credentials`	Informacje o koncie usługi Tłumacz Google	
`project`	Projekt Tłumacza Google	
`location`	Lokalizacja Tłumacza Google	

Machine translation service provided by Google Cloud services.

Zobacz także:

Google translate documentation, Getting started with authentication on Google Cloud, Creating Google Translate project, Google Cloud App Engine locations

2.13.9 LibreTranslate

Nowe w wersji 4.7.1.

ID usługi
`libretranslate`

Konfiguracja

`url`	URL API	
`key`	Klucz API	

LibreTranslate is a free and open-source service for machine translations. The public instance requires an API key, but LibreTranslate can be self-hosted and there are several mirrors available to use the API for free.

`https://libretranslate.com/` (official public instance)
Requires an API key to use outside of the website.

Zobacz także:

LibreTranslate website, LibreTranslate repository, LibreTranslate mirrors

2.13.10 Microsoft Terminology

Nowe w wersji 2.19.

ID usługi
`microsoft-terminology`

Konfiguracja
Ta usługa nie ma konfiguracji.

The Microsoft Terminology Service API allows you to programmatically access the terminology, definitions and user interface (UI) strings available in the Language Portal through a web service.

Zobacz także:

Microsoft Terminology Service API

2.13.11 Microsoft Translator

Nowe w wersji 2.10.

ID usługi
`microsoft-translator`

Konfiguracja

`key`	Klucz API	
`endpoint_url`	Adres URL punktu końcowego aplikacji	
`base_url`	Bazowy adres URL aplikacji	Dostępne opcje: `api.cognitive.microsofttranslator.com` – Global (non-regional) `api-apc.cognitive.microsofttranslator.com` – Asia Pacific `api-eur.cognitive.microsofttranslator.com` – Europe `api-nam.cognitive.microsofttranslator.com` – North America `api.translator.azure.cn` – China
`region`	Region aplikacji	

Machine translation service provided by Microsoft in Azure portal as a one of Cognitive Services.

Weblate implements Translator API V3.

Translator Text API V2

The key you use with Translator API V2 can be used with API 3.

Translator Text API V3

You need to register at Azure portal and use the key you obtain there. With new Azure keys, you also need to set `region` to locale of your service.

Podpowiedź: For Azure China, please use your endpoint from the Azure Portal.

Zobacz także:

Cognitive Services - Text Translation API, Microsoft Azure Portal, Base URLs, „Authenticating with a Multi-service resource" „Authenticating with an access token" section

2.13.12 ModernMT

Nowe w wersji 4.2.

ID usługi
`modernmt`

Konfiguracja

`url`	URL API	
`key`	Klucz API	

Zobacz także:

`API ModernMT<https://www.modernmt.com/api/#translation>`_,

2.13.13 MyMemory

ID usługi
`mymemory`

Konfiguracja

`email`	Kontakt e-mail	
`username`	Nazwa użytkownika	
`key`	Klucz API	

Huge translation memory with machine translation.

Free, anonymous usage is currently limited to 100 requests/day, or to 1000 requests/day when you provide a contact e-mail address in `email`. You can also ask them for more.

Zobacz także:

Strona internetowa MyMemory

2.13.14 Netease Sight

Nowe w wersji 3.3.

ID usługi
`netease-sight`

Konfiguracja

`key`	ID klienta	
`secret`	Tajny klucz klienta	

Usługa tłumaczenia maszynowego świadczona przez NetEase.

This service uses an API, and you need to obtain key and secret from NetEase.

Zobacz także:

NetEase Sight Translation Platform

2.13.15 SAP Translation Hub

ID usługi
`sap-translation-hub`

Konfiguracja

`url`	URL API	
`key`	Klucz API	
`username`	Nazwa użytkownika SAP	
`password`	Hasło SAP	
`enable_mt`	Włącz tłumaczenie maszynowe	

Machine translation service provided by SAP.

You need to have a SAP account (and the SAP Translation Hub enabled in the SAP Cloud Platform) to use this service.

You can also configure whether to also use machine translation services, in addition to the term database.

Informacja: To access the Sandbox API, you need to set `url` and `key`.

To access the productive API, you need to set `url`, `username` and `password`.

Zobacz także:

SAP Translation Hub API

2.13.16 tmserver

ID usługi
`tmserver`

Konfiguracja

`url`	URL API	

You can run your own translation memory server by using the one bundled with Translate-toolkit and let Weblate talk to it. You can also use it with an amaGama server, which is an enhanced version of tmserver.

1. First you will want to import some data to the translation memory:

```
build_tmdb -d /var/lib/tm/db -s en -t cs locale/cs/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t de locale/de/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t fr locale/fr/LC_MESSAGES/django.po
```

2. Start tmserver to listen to your requests:

```
tmserver -d /var/lib/tm/db
```

3. Configure Weblate to talk to it, the default URL is `http://localhost:8888/tmserver/`.

Zobacz także:

tmserver Installing amaGama, Amagama, Amagama Translation Memory

2.13.17 Weblate

ID usługi
`weblate`

Konfiguracja
Ta usługa nie ma konfiguracji.

Weblate machine translation service can provide translations for strings that are already translated inside Weblate. It looks for exact matches in the existing strings.

2.13.18 Pamięć tłumaczeniowa Weblate

Nowe w wersji 2.20.

ID usługi
`weblate-translation-memory`

Konfiguracja
Ta usługa nie ma konfiguracji.

Use *Pamięć tłumaczeniowa* as a machine translation service. Any string that has been translated in past (or uploaded to the translation memory) can be translated in this way.

2.13.19 Yandex

ID usługi
`yandex`

Konfiguracja

`key`	Klucz API	

Machine translation service provided by Yandex.

This service uses a Translation API, and you need to obtain an API key from Yandex.

Zobacz także:

Yandex Translate API, Powered by Yandex.Translate

2.13.20 Youdao Zhiyun

Nowe w wersji 3.2.

ID usługi
`youdao-zhiyun`

Konfiguracja

`key`	ID klienta	
`secret`	Tajny klucz klienta	

Machine translation service provided by Youdao.

This service uses an API, and you need to obtain an ID and an API key from Youdao.

Zobacz także:

Youdao Zhiyun Natural Language Translation Service

2.13.21 Niestandardowe tłumaczenie maszynowe

You can also implement your own machine translation services using a few lines of Python code. This example implements machine translation in a fixed list of languages using `dictionary` Python module:

```
#
# Copyright © 2012-2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#
"""Machine translation example."""

import dictionary

from weblate.machinery.base import MachineTranslation


class SampleTranslation(MachineTranslation):
    """Sample machine translation interface."""

    name = "Sample"

    def download_languages(self):
        """Return list of languages your machine translation supports."""
        return {"cs"}

    def download_translations(
        self,
        source,
        language,
        text: str,
        unit,
        user,
        search: bool,
        threshold: int = 75,
    ):
        """Return tuple with translations."""
        for t in dictionary.translate(text):
            yield {"text": t, "quality": 100, "service": self.name, "source": text}
```

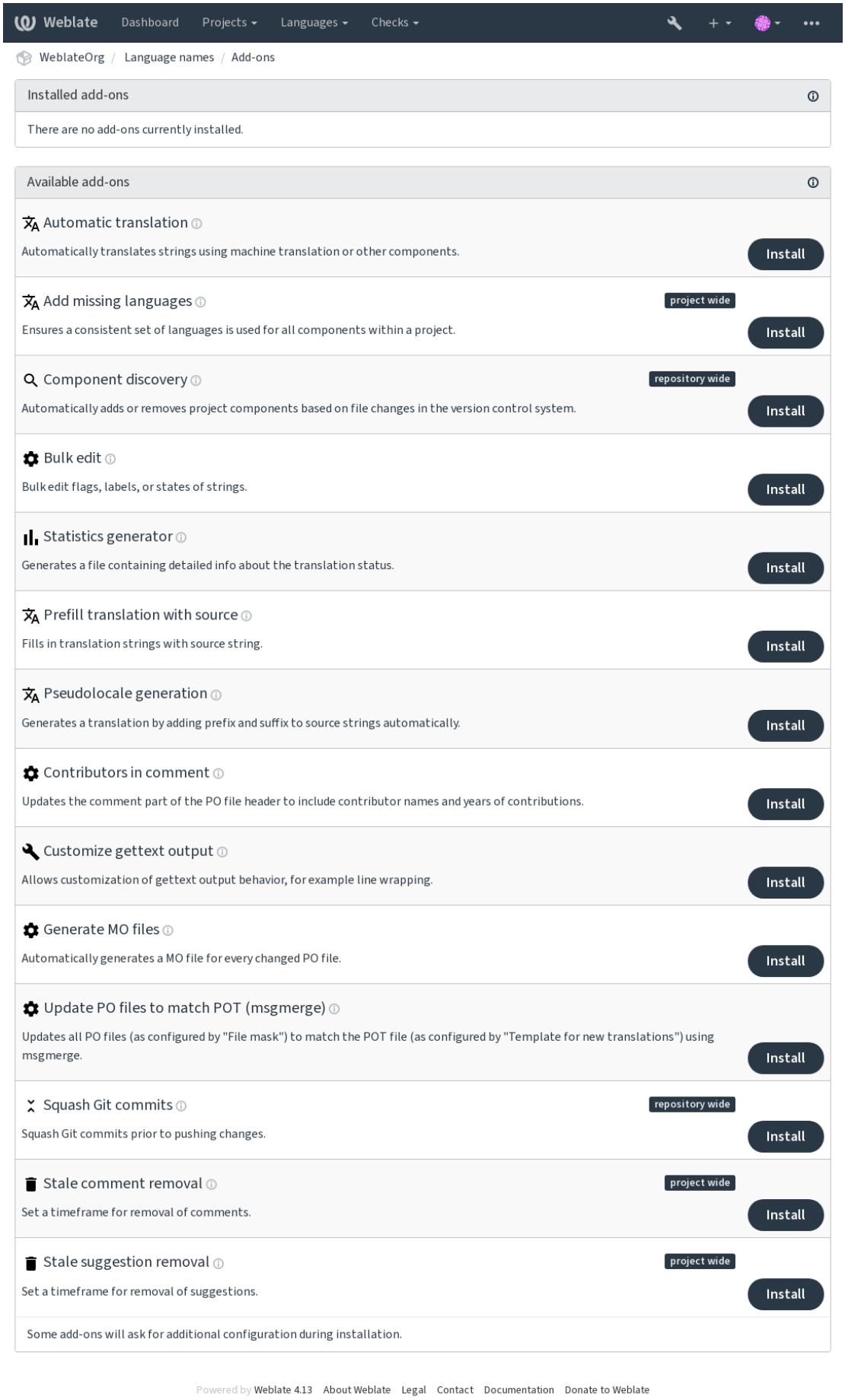
You can list your own class in *WEBLATE_MACHINERY* and Weblate will start using that.

2.14 Dodatki

Nowe w wersji 2.19.

Add-ons provide ways to customize and automate the translation workflow. Admins can add and manage add-ons from the *Manage* ↓ *Add-ons* menu of each respective translation component.

Podpowiedź: You can also configure add-ons using *API*, `DEFAULT_ADDONS`, or `install_addon`.


Weblate
Dashboard
Projects
Languages
Checks
WeblateOrg / Language names / Add-ons
Installed add-ons
There are no add-ons currently installed.
Available add-ons
Automatic translation
Automatically translates strings using machine translation or other components.
Install
Add missing languages
project wide
Ensures a consistent set of languages is used for all components within a project.
Install
Component discovery
repository wide
Automatically adds or removes project components based on file changes in the version control system.
Install
Bulk edit
Bulk edit flags, labels, or states of strings.
Install
Statistics generator
Generates a file containing detailed info about the translation status.
Install
Prefill translation with source
Fills in translation strings with source string.
Install
Pseudolocale generation
Generates a translation by adding prefix and suffix to source strings automatically.
Install
Contributors in comment
Updates the comment part of the PO file header to include contributor names and years of contributions.
Install
Customize gettext output
Allows customization of gettext output behavior, for example line wrapping.
Install
Generate MO files
Automatically generates a MO file for every changed PO file.
Install
Update PO files to match POT (msgmerge)
Updates all PO files (as configured by "File mask") to match the POT file (as configured by "Template for new translations") using msgmerge.
Install
Squash Git commits
repository wide
Squash Git commits prior to pushing changes.
Install
Stale comment removal
project wide
Set a timeframe for removal of comments.
Install
Stale suggestion removal
project wide
Set a timeframe for removal of suggestions.
Install
Some add-ons will ask for additional configuration during installation.
Powered by Weblate 4.13 About Weblate Legal Contact Documentation Donate to Weblate

2.14.1 Wbudowane dodatki

Tłumaczenie automatyczne

Nowe w wersji 3.9.

ID dodatku
`weblate.autotranslate.autotranslate`

Konfiguracja

`mode`	Tryb automatycznego tłumaczenia	Dostępne opcje: `suggest` – Dodaj jako sugestię `translate` – Dodaj jako tłumaczenie `fuzzy` – Dodaj jako wymagające edycji
`filter_type`	Filtr wyszukiwania	Pamiętaj, że tłumaczenie wszystkich ciągów spowoduje odrzucenie wszystkich istniejących tłumaczeń. Dostępne opcje: `all` – Wszystkie ciągi `nottranslated` – Nieprzetłumaczone ciągi `todo` – Niedokończone ciągi `fuzzy` – Ciągi oznaczone do edycji `check:inconsistent` – Failing check: Inconsistent
`auto_source`	Źródło automatycznego tłumaczenia	Dostępne opcje: `others` – Inne komponenty tłumaczenia `mt` – Tłumaczenie maszynowe
`component`	Komponenty	Wprowadź komponent do użycia jako źródło; pozostaw puste, aby użyć wszystkich komponentów w bieżącym projekcie.
`engines`	Silniki tłumaczenia maszynowego	
`threshold`	Ocena wyniku	

Triggers
aktualizacja komponentów, codziennie

Automatycznie tłumaczy ciągi przy użyciu tłumaczenia maszynowego lub innych komponentów.

Jest wyzwalane:

- Gdy w komponencie pojawią się nowe ciągi.
- Once in a month for every component, this can be configured using `BACKGROUND_TASKS`.

Zobacz także:

Tłumaczenie automatyczne, *Zachowanie takich samych tłumaczeń między komponentami*

Lokalizacja JavaScript CDN

Nowe w wersji 4.2.

ID dodatku
`weblate.cdn.cdnjs`

Konfiguracja

`threshold`	Próg tłumaczenia	Próg uwzględniania tłumaczeń.
`css_selector`	Selektor CSS	Selektor CSS do wykrywania lokalizowalnych elementów.
`cookie_name`	Nazwa pliku cookie języka	Nazwa pliku cookie przechowującego preferencje językowe.
`files`	Wyodrębnianie ciągów z plików HTML	Lista nazw plików w bieżącym repozytorium lub zdalnych adresów URL do przeanalizowania w celu przetłumaczenia ciągów.

Triggers
daily, repository post-commit, repository post-update

Publikuje tłumaczenia w content delivery network dla lokalizacji JavaScript lub HTML.

Can be used to localize static HTML pages, or to load localization in the JavaScript code.

Generates a unique URL for your component you can include in HTML pages to localize them. See weblate-cdn for more details.

Zobacz także:

cdn-addon-config, weblate-cdn, cdn-addon-extract, cdn-addon-html

Usuń puste ciągi

Nowe w wersji 4.4.

ID dodatku
`weblate.cleanup.blank`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
repository post-commit, repository post-update

Usuwa ciągi bez tłumaczenia z plików tłumaczeń.

Use this to not have any empty strings in translation files (for example if your localization library displays them as missing instead of falling back to the source string).

Zobacz także:

Does Weblate update translation files besides translations?

Wyczyść pliki tłumaczenia

ID dodatku
`weblate.cleanup.generic`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
repository pre-commit, repository post-update

Zaktualizuj wszystkie pliki tłumaczeń tak, aby odpowiadały jednojęzykowemu plikowi podstawowemu. W przypadku większości formatów plików oznacza to usunięcie nieaktualnych kluczy tłumaczeń, których nie ma już w pliku podstawowym.

Zobacz także:

Does Weblate update translation files besides translations?

Dodaj brakujące języki

ID dodatku
`weblate.consistency.languages`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
codziennie, po dodaniu repozytorium

Zapewnia, że dla wszystkich komponentów w projekcie używany jest spójny zestaw języków.

Brakujące języki są sprawdzane raz na 24 godziny oraz gdy nowe języki są dodawane w Weblate.

W przeciwieństwie do większości ten dodatek wpływa na cały projekt.

Podpowiedź: Auto-translate the newly added strings with *Tłumaczenie automatyczne*.

Wykrycie komponentów

ID dodatku
`weblate.discovery.discovery`

Konfiguracja

`match`	Wyrażenie regularne dopasowujące pliki tłumaczeń	
`file_format`	Format pliku	
`name_template`	Dostosuj nazwę komponentu	
`base_file_template`	Zdefiniuj jednojęzyczną bazową nazwę pliku	Pozostaw puste dla plików tłumaczeń dwujęzycznych.
`new_base_template`	Zdefiniuj plik bazowy dla nowych tłumaczeń	Nazwa pliku używanego do tworzenia nowych tłumaczeń. Dla Gettext wybierz plik .pot.
`intermediate_template`	Plik języka pośredniego	Nazwa pliku tłumaczenia pośredniego. W większości przypadków jest to plik tłumaczenia dostarczony przez deweloperów i jest używany podczas tworzenia faktycznych ciągów źródłowych.
`language_regex`	Filtr języka	Wyrażenie regularne do filtrowania plików tłumaczeń podczas skanowania w poszukiwaniu maski pliku.
`copy_addons`	Klonuj dodatki z głównego komponentu do nowo utworzonych	
`remove`	Usuń komponenty dla nieistniejących plików	
`confirm`	Potwierdzam, że powyższe dopasowania wyglądają prawidłowo	

Triggers
po aktualizacji repozytorium

Automatycznie dodaje lub usuwa komponenty projektu na podstawie zmian plików w systemie kontroli wersji.

Triggered each time the VCS is updated, and otherwise similar to the `import_project` management command. This way you can track multiple translation components within one VCS.

The matching is done using regular expressions enabling complex configuration, but some knowledge is required to do so. Some examples for common use cases can be found in the add-on help section.

Once you hit *Save*, a preview of matching components will be presented, from where you can check whether the configuration actually matches your needs:

Weblate Dashboard Projects Languages Checks

WeblateOrg / Language names / Add-ons / Component discovery

Configure add-on

- Please review and confirm the matched components.

Component	Matched files
The following components would be created	
Djangojs	`weblate/locale/he/LC_MESSAGES/djangojs.po` (he) `weblate/locale/cs/LC_MESSAGES/djangojs.po` (cs) `weblate/locale/hu/LC_MESSAGES/djangojs.po` (hu)
Django	`weblate/locale/hu/LC_MESSAGES/django.po` (hu) `weblate/locale/cs/LC_MESSAGES/django.po` (cs) `weblate/locale/he/LC_MESSAGES/django.po` (he)

☐ **I confirm the above matches look correct**

Regular expression to match translation files against

weblate/locale/(?P<language>[^/]*)/LC_MESSAGES/(?P<component>[^/]*)\.po

File format

gettext PO file

Customize the component name

{{ component|title }}

Define the monolingual base filename

Leave empty for bilingual translation files.

Define the base file for new translations

weblate/locale/{{ component }}.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Language filter

^(cs|he|hu)$

Regular expression to filter translation files against when scanning for file mask.

☑ **Clone add-ons from the main component to the newly created ones**

☐ **Remove components for inexistent files**

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description

`(?P<language>[^/.]*)/(?P<component>[^/]*)\.po`	`cs/application.po`
	`cs/website.po`
	`de/application.po`
	`de/website.po`
		One folder per language containing translation files for components.

`locale/(?P<language>[^/.]*)/LC_MESSAGES/(?P<component>[^/]*)\.po`	`locale/cs/LC_MESSAGES/application.po`
	`locale/cs/LC_MESSAGES/website.po`
	`locale/de/LC_MESSAGES/application.po`
	`locale/de/LC_MESSAGES/website.po`
		Usual structure for storing gettext PO files.

`src/locale/(?P<component>[^/]*)\.(?P<language>[^/.]*)\.po`	`src/locale/application.cs.po`
	`src/locale/website.cs.po`
	`src/locale/application.de.po`
	`src/locale/website.de.po`
		Using both component and language name within filename.

`locale/(?P<language>[^/.]*)/(?P<component>[^/]*)/(?P=language)\.po`	`locale/cs/application/cs.po`
	`locale/cs/website/cs.po`
	`locale/de/application/de.po`
	`locale/de/website/de.po`
		Using language in both path and filename.

`res/values-(?P<language>[^/.]*)/strings-(?P<component>[^/]*)\.xml`	`res/values-cs/strings-about.xml`
	`res/values-cs/strings-help.xml`
	`res/values-de/strings-about.xml`
	`res/values-de/strings-help.xml`
		Android resource strings, split into several files.

You can use Django template markup in both component name and the monolingual base filename, for example:

`{{ component }}`
Component filename match
`{{ component|title }}`
Component filename with upper case first letter

Save

Powered by Weblate 4.13 About Weblate Legal Contact Documentation Donate to Weblate

Podpowiedź: Component discovery add-on uses *Wewnętrzne adresy URL Weblate*. It's a convenient way to share VCS setup between multiple components. Linked components use the local repository of the main component set up by filling `weblate://project/main-component` into the *Repozytorium kodu źródłowego* field (in *Manage* ↓ *Settings* ↓ *Version control system*) of each respective component. This saves time with configuration and system resources too.

Zobacz także:

Znaczniki szablonu

Edycja zbiorcza

Nowe w wersji 3.11.

ID dodatku
`weblate.flags.bulk`

Konfiguracja

`q`	Zapytanie	
`state`	Stan do ustawienia	Dostępne opcje: `-1` – Nie zmieniaj `10` – Wymaga edycji `20` – Przetłumaczone `30` – Zatwierdzone
`add_flags`	Flagi tłumaczeń do dodania	
`remove_flags`	Flagi tłumaczeń do usunięcia	
`add_labels`	Etykiety do dodania	
`remove_labels`	Etykiety do usunięcia	

Triggers
aktualizacja komponentów

Zbiorcza edycja flag, etykiet lub stanów ciągów.

Automate labeling by starting out with the search query `NOT has:label` and add labels till all strings have all required labels. Other automated operations for Weblate metadata can also be done.

Przykłady:

Tabela 5: Label new strings automatically

Zapytanie wyszukiwania	`NOT has:label`
Etykiety do dodania	*ostatnie*

Tabela 6: Marking all Pliki metadanych sklepu z aplikacjami changelog strings read-only

Zapytanie wyszukiwania	`language:en AND key:changelogs/`
Flagi tłumaczeń do dodania	`read-only`

Zobacz także:

Edycja zbiorcza, *Dostosowywanie zachowania za pomocą flag*, labels

Oznacz niezmienione tłumaczenia jako „Wymaga edycji”

Nowe w wersji 3.1.

ID dodatku
`weblate.flags.same_edit`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
unit post-create

Za każdym razem, gdy nowy przetłumaczalny ciąg jest importowany z VCS i pasuje do ciągu źródłowego, jest oznaczany jako wymagający edycji w Weblate. Szczególnie przydatne w przypadku formatów plików zawierających ciągi źródłowe nieprzetłumaczonych ciągów.

Podpowiedź: You might also want to tighthen the *Niezmienione tłumaczenie* check by adding `strict-same` flag to *Flagi tłumaczeń*.

Zobacz także:

Stan tłumaczenia

Oznacz nowe ciągi źródłowe jako „Wymaga edycji”

ID dodatku
`weblate.flags.source_edit`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
unit post-create

Za każdym razem, gdy nowy źródłowy ciąg znaków jest importowany z VCS, jest oznaczany jako wymagający edycji w Weblate. W ten sposób można łatwo filtrować i edytować ciągi źródłowe napisane przez twórców oprogramowania.

Zobacz także:

Stan tłumaczenia

Oznacz nowe tłumaczenia jako „Wymaga edycji”

ID dodatku
`weblate.flags.target_edit`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
unit post-create

Za każdym razem, gdy zaimportowany zostanie nowy łańcuch do przetłumaczenia z VCS, zostanie oznaczony jako wymagający edycji w Weblate. W ten sposób możesz łatwo filtrować i edytować tłumaczenia stworzone przez programistów.

Zobacz także:

Stan tłumaczenia

Generator statystyk

ID dodatku
`weblate.generate.generate`

Konfiguracja

`filename`	Nazwa wygenerowanego pliku	
`template`	Zawartość wygenerowanego pliku	

Triggers
przed commitem na repozytorium

Generuje plik zawierający szczegółowe informacje o stanie tłumaczenia.

Możesz używać szablonów Django zarówno w nazwie pliku, jak i treści, sprawdź *Znaczniki szablonu* żeby poznać szczegóły szablonów.

Na przykład wygenerowanie pliku podsumowania dla każdego tłumaczenia:

Nazwa wygenerowanego pliku
`locale/{{ language_code }}.json`

Zawartość

```
{
   "language": "{{ language_code }}",
   "strings": "{{ stats.all }}",
   "translated": "{{ stats.translated }}",
   "last_changed": "{{ stats.last_changed }}",
   "last_author": "{{ stats.last_author }}",
}
```

Zobacz także:

Znaczniki szablonu

Wypełnij wstępnie tłumaczenie źródłem

Nowe w wersji 4.11.

ID dodatku
`weblate.generate.prefill`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
aktualizacja komponentów, codziennie

Wypełnia ciągi tłumaczeń ciągami źródłowymi.

All untranslated strings in the component will be filled with the source string, and marked as needing edit. Use this when you can not have empty strings in the translation files.

Generowanie pseudolokalizacji

Nowe w wersji 4.5.

ID dodatku
`weblate.generate.pseudolocale`

Konfiguracja

`source`	Ciągi źródłowe	
`target`	Tłumaczenie docelowe	Wszystkie ciągi w tym tłumaczeniu zostaną nadpisane
`prefix`	Stały prefiks ciągu	
`var_prefix`	Zmienny prefiks ciągu	
`suffix`	Stały sufiks ciągu	
`var_suffix`	Zmienny sufiks ciągu	
`var_multiplier`	Mnożnik części zmiennej	Ile razy powtarzać część zmienną w zależności od długości ciągu źródłowego.

Triggers
aktualizacja komponentów, codziennie

Generuje tłumaczenie poprzez automatyczne dodanie prefiksu i sufiksu do ciągów źródłowych.

Pseudolocales are useful to find strings that are not prepared for localization. This is done by altering all translatable source strings to make it easy to spot unaltered strings when running the application in the pseudolocale language.

Finding strings whose localized counterparts might not fit the layout is also possible.

Using the variable parts makes it possible to look for strings which might not fit into the user interface after the localization - it extends the text based on the source string length. The variable parts are repeated by length of the text multiplied by the multiplier. For example `Hello world` with variable suffix _ and variable multiplier of 1 becomes `Hello world`_____ - the suffix is repeated once for each character in the source string.

Ciągi zostaną wygenerowane według następującego wzorca:

Fixed string prefix Variable string prefix Source string Variable string suffix Fixed string suffix

---

Podpowiedź: You can use real languages for testing, but there are dedicated pseudolocales available in Weblate - *en_XA* and *ar_XB*.

Podpowiedź: You can use this add-on to start translation to a new locale of an existing language or similar language. Once you add the translation to the component, follow to the add-on. *Example:* If you have *fr* and want to start *fr_CA* translation, simply set *fr* as the source, *fr_CA* as the target, and leave the prefix and suffix blank.

Uninstall the add-on once you have the new translation filled to prevent Weblate from changing the translations made after the copying.

Komentarz współautora

ID dodatku
`weblate.gettext.authors`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
przed commitem na repozytorium

Zaktualizuj część komentarz w nagłówku pliku PO, aby zawierał nazwę autora i lata wkładu.

The PO file header will look like this:

```
# 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.
```

Zaktualizuj zmienną ALL_LINGUAS w pliku „konfiguracyjnym”

ID dodatku
`weblate.gettext.configure`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
po dodaniu repozytorium, codziennie

Updates the ALL_LINGUAS variable in `configure`, `configure.in` or any `configure.ac` files, when a new translation is added.

Dostosuj wyjście gettext

ID dodatku
`weblate.gettext.customize`

Konfiguracja

`width`	Zawijanie długich linii	Domyślnie gettext zawija linie przy 77 znakach i nowych liniach. Z parametrem –no-wrap, zawija jest wykonywane tylko przy przejściu do nowych linii. Dostępne opcje: `77` – Zawijaj linie przy 77 znakach i przy nowych liniach `65535` – Zawijaj linie tylko przy nowych liniach `-1` – Bez zawijania linii

Triggers
po załadowaniu do pamięci

Umożliwia dostosowanie zachowania wyjścia gettext, na przykład zawijanie wierszy.

Oferuje następujące opcje:

- Zawijaj linie przy 77 znakach i przy nowych liniach
- Zawijaj linie tylko przy nowych liniach
- Bez zawijania linii

Informacja: Domyślnie gettext zawija linie przy 77 znakach i nowych liniach. Z parametrem `--no-wrap` zawijanie jest wykonywane tylko przy przejściu do nowych linii.

Aktualizuj plik LINGUAS

ID dodatku
`weblate.gettext.linguas`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
po dodaniu repozytorium, codziennie

Aktualizuje plik LINGUAS, gdy dodawane jest nowe tłumaczenie.

Generuj pliki MO

ID dodatku
`weblate.gettext.mo`

Konfiguracja

`path`	Ścieżka wygenerowanego pliku MO	Jeśli nie zostanie określony, zostanie użyta lokalizacja pliku PO.

Triggers
przed commitem na repozytorium

Automatycznie generuje plik MO dla każdego zmienionego pliku PO.

The location of the generated MO file can be customized and the field for it uses *Znaczniki szablonu*.

Zaktualizuj pliki PO, aby dopasować POT (msgmerge)

ID dodatku
`weblate.gettext.msgmerge`

Konfiguracja

`previous`	Zachowaj poprzednie msgids przetłumaczonych ciągów	
`no_location`	Usuwanie lokalizacji przetłumaczonych ciągów	
`fuzzy`	Użyj dopasowania rozmytego	

Triggers
po aktualizacji repozytorium

Aktualizuje wszystkie pliki PO (zgodnie z konfiguracją *Maska pliku*) tak, aby pasowały do pliku POT (zgodnie z konfiguracją *Szablon dla nowych tłumaczeń*) przy użyciu **`msgmerge`**.

Wyzwalany za każdym razem, gdy nowe zmiany są pobierane z repozytorium nadrzędnego. Większość opcji wiersza polecenia msgmerge można ustawić za pomocą konfiguracji dodatku.

Zobacz także:

Does Weblate update translation files besides translations?

Zesquashowane commity na Git

ID dodatku
`weblate.git.squash`

Konfiguracja

`squash`	Squashowanie commitów	Dostępne opcje: `all` – Wszystkie commity w jeden `language` – Według języka `file` – Per file `author` – Per author
`append_trailers`	Dołącz zakończenia do opisu zesquashowanego commita	Linie końcowe to linie, które wyglądają podobnie do nagłówków wiadomości e-mail RFC 822, na końcu części opisu commita, takie jak „Współtworzone przez: …”.
`commit_message`	Opis commita	Ten opis commita będzie użyty zamiast połączonych opisów commitów z zesquashowanych commitów.

Triggers
po commicie na repozytorium

Zesquashuj commity na Git przed wysłaniem zmian.

Git commits can be squashed prior to pushing changes in one of the following modes:

- Wszystkie commity w jeden
- Według języka
- Według pliku
- Według autora

Oryginalne opisy commitów są zachowywane, ale autorstwo zostaje utracone, chyba że wybrano dla autora opcję :guilabel:`Na autora`lub dostosowano opis commita, aby go uwzględnić.

Oryginalne opisy commitów można opcjonalnie zastąpić niestandardowym opisem commita.

Trailers (commit lines like `Co-authored-by: …`) can optionally be removed from the original commit messages and appended to the end of the squashed commit message. This also generates proper `Co-authored-by:` credit for every translator.

Dostosuj wyjście JSON

ID dodatku
`weblate.json.customize`

Konfiguracja

`sort_keys`	Sortuj klucze JSON	
`indent`	Wcięcie JSON	
`style`	Styl wcięć JSON	Dostępne opcje: `spaces` – Spaces `tabs` – Tabs

Triggers
po załadowaniu do pamięci

Pozwala na dostosowanie zachowania wyjścia JSON, na przykład wcięcia lub sortowanie.

Format the Java properties file

ID dodatku
`weblate.properties.sort`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
przed commitem na repozytorium

Formats and sorts the Java properties file.

- Consolidates newlines to Unix ones.
- Uppercase formatting of Unicode escape sequences (in case they are present).
- Strips blank lines and comments.
- Sorts the strings by the keys.
- Drops duplicate strings.

Usuwanie skasowanych komentarzy

Nowe w wersji 3.7.

ID dodatku
`weblate.removal.comments`

Konfiguracja

`age`	Dni do zachowania	

Triggers
codziennie

Ustaw przedział czasowy na usunięcie komentarzy.

Może to być przydatne do usuwania starych komentarzy, które mogły stać się nieaktualne. Używaj z rozwagą, ponieważ starzenie się komentarzy nie oznacza, że straciły one na znaczeniu.

Usuwanie nieaktualnych sugestii

Nowe w wersji 3.7.

ID dodatku
`weblate.removal.suggestions`

Konfiguracja

`age`	Dni do zachowania	
`votes`	Próg głosowania	Próg do usunięcia. To pole nie ma wpływu na wyłączone głosowanie.

Triggers
codziennie

Ustaw przedział czasowy na usunięcie sugestii.

Can be very useful in connection with suggestion voting (see *Ocena koleżeńska*) to remove suggestions which don't receive enough positive votes in a given timeframe.

Zaktualizuj pliki RESX

Nowe w wersji 3.9.

ID dodatku
`weblate.resx.update`

Konfiguracja
Ten dodatek nie ma konfiguracji.

Triggers
po aktualizacji repozytorium

Zaktualizuj wszystkie pliki tłumaczeń, aby pasowały do jednojęzycznego pliku podstawowego. Nieużywane ciągi są usuwane, a nowe są dodawane jako kopie ciągu źródłowego.

Podpowiedź: Use *Wyczyść pliki tłumaczenia* if you only want to remove stale translation keys.

Zobacz także:

Does Weblate update translation files besides translations?

Dostosuj wyjście YAML

Nowe w wersji 3.10.2.

ID dodatku
`weblate.yaml.customize`

Konfiguracja

`indent`	Wcięcie YAML	
`width`	Zawijanie długich linii	Dostępne opcje: `80` – Zawijaj linie przy 80 znakach `100` – Zawijaj linie przy 100 znakach `120` – Zawijaj linie przy 120 znakach `180` – Zawijaj linie przy 180 znakach `65535` – Bez zawijania linii
`line_break`	Podział wierszy	Dostępne opcje: `dos` – DOS (\r\n) `unix` – UNIX (\n) `mac` – MAC (\r)

Triggers
po załadowaniu do pamięci

Umożliwia dostosowanie zachowania wyjściowego YAML, na przykład długość linii lub nowe linie.

2.14.2 Dostosowywanie listy dodatków

The list of add-ons is configured by `WEBLATE_ADDONS`. To add another add-on, simply include the absolute class name in this setting.

2.14.3 Pisanie dodatku

You can write your own add-ons too, create a subclass of `weblate.addons.base.BaseAddon` to define the add-on metadata, and then implement a callback to do the processing.

Zobacz także:

Tworzenie dodatków

2.14.4 Wykonywanie skryptów z dodatku

Add-ons can also be used to execute external scripts. This used to be integrated in Weblate, but now you have to write some code to wrap your script with an add-on.

```
#
# Copyright © 2012-2022 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 descrption
    verbose = _("Execute script before commit")
    description = _("This add-on executes a script.")

    # Script to execute
    script = "/bin/true"
    # File to add in commit (for pre commit event)
    # does not have to be set
    add_file = "po/{{ language_code }}.po"
```

Aby uzyskać instrukcje instalacji, zobacz *Custom quality checks, add-ons and auto-fixes*.

The script is executed with the current directory set to the root of the VCS repository for any given component.

Dodatkowo dostępne są następujące zmienne środowiskowe:

WL_VCS
Używany system kontroli wersji.

WL_REPO
Adres URL repozytorium nadrzędnego.

WL_PATH
Ścieżka bezwzględna do repozytorium VCS.

WL_BRANCH
Nowe w wersji 2.11.

Gałąź repozytorium skonfigurowana w bieżącym komponencie.

WL_FILEMASK
Maska pliku dla bieżącego komponentu.

WL_TEMPLATE
Nazwa pliku szablonu dla tłumaczeń jednojęzycznych (może być pusta).

WL_NEW_BASE
Nowe w wersji 2.14.

Nazwa pliku użytego do tworzenia nowych tłumaczeń (może być pusta).

WL_FILE_FORMAT
Format pliku używany w bieżącym komponencie.

WL_LANGUAGE
Language of currently processed translation (not available for component-level hooks).

WL_PREVIOUS_HEAD
Previous HEAD after update (only available after running the post-update hook).

WL_COMPONENT_SLUG
Nowe w wersji 3.9.

Component slug used to construct URL.

WL_PROJECT_SLUG
Nowe w wersji 3.9.

Project slug used to construct URL.

WL_COMPONENT_NAME
Nowe w wersji 3.9.

Nazwa komponentu.

WL_PROJECT_NAME
Nowe w wersji 3.9.

Nazwa projektu.

WL_COMPONENT_URL
Nowe w wersji 3.9.

URL komponentu.

WL_ENGAGE_URL

Nowe w wersji 3.9.

Adres URL zaangażowania projektu.

Zobacz także:

Konfiguracja komponentu

Przetwarzanie repozytorium po aktualizacji

Can be used to update translation files when the VCS upstream source changes. To achieve this, please remember Weblate only sees files committed to the VCS, so you need to commit changes as a part of the script.

For example with Gulp you can do it using following code:

```
#! /bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Przetwarzanie tłumaczeń przed scommitowaniem

Use the commit script to automatically change a translation before it is committed to the repository.

It is passed as a single parameter consisting of the filename of a current translation.

2.15 Pamięć tłumaczeniowa

Nowe w wersji 2.20.

Weblate comes with a built-in translation memory consisting of the following:

- Manually imported translation memory (see *Interfejs użytkownika*).
- Automatically stored translations performed in Weblate (depending on *Zakresy pamięci tłumaczeniowej*).
- Automatycznie importowane wcześniejsze tłumaczenia.

Content in the translation memory can be applied one of two ways:

- Manually, *Automatyczne sugestie* view while translating.
- Automatically, by translating strings using *Tłumaczenie automatyczne*, or *Tłumaczenie automatyczne* add-on.

For installation tips, see *Pamięć tłumaczeniowa Weblate*, which is turned on by default.

2.15.1 Zakresy pamięci tłumaczeniowej

Nowe w wersji 3.2: In earlier versions translation memory could be only loaded from a file corresponding to the current imported translation memory scope.

The translation memory scopes are there to allow both privacy and sharing of translations, to suit the desired behavior.

Importowana pamięć tłumaczeniowa

Importing arbitrary translation memory data using the `import_memory` command makes memory content available to all users and projects.

Pamięć tłumaczeniowa na użytkownika

Stores all user translations automatically in the personal translation memory of each respective user.

Pamięć tłumaczeniowa na projekt

All translations within a project are automatically stored in a project translation memory only available for this project.

Współużytkowana pamięć tłumaczeniowa

All translation within projects with shared translation memory turned on are stored in a shared translation memory available to all projects.

Please consider carefully whether to turn this feature on for shared Weblate installations, as it can have severe implications:

- The translations can be used by anybody else.
- This might lead to disclosing secret information.

2.15.2 Zarządzanie pamięcią tłumaczeniową

Interfejs użytkownika

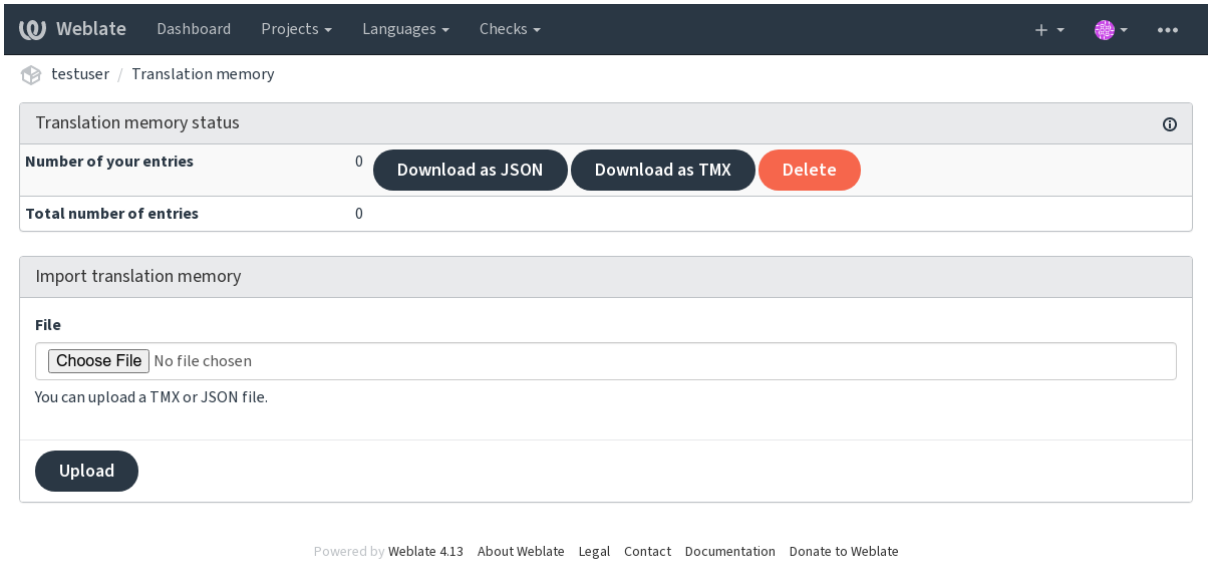
Nowe w wersji 3.2.

In the basic user interface you can manage per user and per project translation memories. It can be used to download, wipe or import translation memory.

Podpowiedź: Translation memory in JSON can be imported into Weblate, TMX is provided for interoperability with other tools.

Zobacz także:

Schemat pamięci tłumaczeniowej Weblate



Interfejs zarządzania

There are several management commands to manipulate the translation memory content. These operate on the translation memory as whole, unfiltered by scopes (unless requested by parameters):

dump_memory
Eksportuje pamięć do formatu JSON

import_memory
Imports TMX or JSON files into the translation memory

2.16 Konfiguracja

All settings are stored in `settings.py` (as is usual for Django).

Informacja: After changing any of these settings, you need to restart Weblate - both WSGI and Celery processes.

In case it is run as `mod_wsgi`, you need to restart Apache to reload the configuration.

Zobacz także:

Please also check Django's documentation for parameters configuring Django itself.

2.16.1 AKISMET_API_KEY

Weblate can use Akismet to check incoming anonymous suggestions for spam. Visit akismet.com to purchase an API key and associate it with a site.

2.16.2 ANONYMOUS_USER_NAME

Username of users that are not signed in.

Zobacz także:

Kontrola dostępu

2.16.3 AUDITLOG_EXPIRY

Nowe w wersji 3.6.

How many days Weblate should keep audit logs, which contain info about account activity.

Domyślnie 180 dni.

2.16.4 AUTH_LOCK_ATTEMPTS

Nowe w wersji 2.14.

Maximum number of failed authentication attempts before rate limiting is applied.

This is currently applied in the following locations:

- Sign in. Deletes the account password, preventing the user from signing in without requesting a new password.
- Password reset. Prevents new e-mails from being sent, avoiding spamming users with too many password reset attempts.

Domyślna wartość 10.

Zobacz także:

Ograniczenie szybkości

2.16.5 AUTO_UPDATE

Nowe w wersji 3.2.

Zmienione w wersji 3.11: The original on/off option was changed to differentiate which strings are accepted.

Updates all repositories on a daily basis.

Podpowiedź: Useful if you are not using *Hooki powiadomień* to update Weblate repositories automatically.

Informacja: On/off options exist in addition to string selection for backward compatibility.

Dostępne opcje:

`"none"`
Brak codziennych aktualizacji.

`"zdalne"` również `fałszywe`
Tylko zdalne aktualizacje.

`"pełne''` również `Prawdziwe`
Update remotes and merge working copy.

Informacja: This requires that *Zadania w tle korzystające z Celery* is working, and will take effect after it is restarted.

2.16.6 AVATAR_URL_PREFIX

Prefix for constructing avatar URLs as: `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. The following services are known to work:

Gravatar (default), as per https://gravatar.com/

```
AVATAR_URL_PREFIX = 'https://www.gravatar.com/'
```

Libravatar, zgodnie z https://www.libravatar.org/

```
AVATAR_URL_PREFIX = 'https://www.libravatar.org/'
```

Zobacz także:

Pamięć podręczna awatarów, `ENABLE_AVATARS`, *Awatary*

2.16.7 AUTH_TOKEN_VALID

Nowe w wersji 2.14.

How long the authentication token and temporary password from password reset e-mails is valid for. Set in number of seconds, defaulting to 172800 (2 days).

2.16.8 AUTH_PASSWORD_DAYS

Nowe w wersji 2.15.

How many days using the same password should be allowed.

Informacja: Password changes made prior to Weblate 2.15 will not be accounted for in this policy.

Domyślnie 180 dni.

2.16.9 AUTOFIX_LIST

List of automatic fixes to apply when saving a string.

Informacja: Provide a fully-qualified path to the Python class that implementing the autofixer interface.

Dostępne poprawki:

`weblate.trans.autofixes.whitespace.SameBookendingWhitespace`
Matches whitespace at the start and end of the string to the source.

`weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis`
Replaces trailing dots (…) if the source string has a corresponding ellipsis (…).

`weblate.trans.autofixes.chars.RemoveZeroSpace`
Removes zero-width space characters if the source does not contain any.

`weblate.trans.autofixes.chars.RemoveControlChars`
Removes control characters if the source does not contain any.

`weblate.trans.autofixes.html.BleachHTML`
Removes unsafe HTML markup from strings flagged as `safe-html` (see *Niebezpieczny HTML*).

You can select which ones to use:

```
AUTOFIX_LIST = (
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
)
```

Zobacz także:

Automatyczne korekty, *Niestandardowe automatyczne korekty*

2.16.10 BACKGROUND_TASKS

Nowe w wersji 4.5.2.

Defines how often lengthy maintenance tasks should be triggered for a component.

Right now this controls:

- Dodatek *Tłumaczenie automatyczne*
- *Kontrole i korekty* recalculation

Możliwe opcje:

- `monthly` (this is the default)
- `weekly`
- `daily`
- `never`

Informacja: Increasing the frequency is not recommended when Weblate contains thousands of components.

2.16.11 BASE_DIR

Base directory where Weblate sources are located. Used to derive several other paths by default:

- *DATA_DIR*

Default value: Top level directory of Weblate sources.

2.16.12 BASIC_LANGUAGES

Nowe w wersji 4.4.

List of languages to offer users for starting new translation. When not specified built-in list is used which includes all commonly used languages, but without country specific variants.

This only limits non privileged users to add unwanted languages. The project admins are still presented with full selection of languages defined in Weblate.

Informacja: This does not define new languages for Weblate, it only filters existing ones in the database.

Przykład:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Zobacz także:

Definicje języków

2.16.13 BORG_EXTRA_ARGS

Nowe w wersji 4.9.

You can pass additional arguments to **borg create** when built-in backups are triggered.

Przykład:

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/"]
```

Zobacz także:

Tworzenie kopii zapasowych i przenoszenie weblate, borg create

2.16.14 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Customize `Content-Security-Policy` header for Weblate. The header is automatically generated based on enabled integrations with third-party services (Matomo, Google Analytics, Sentry, …).

All these default to empty list.

Przykład:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Zobacz także:

Polityka bezpieczeństwa treści, Content Security Policy (CSP)

2.16.15 CHECK_LIST

List of quality checks to perform on a translation.

Informacja: Provide a fully-qualified path to the Python class implementing the check interface.

Adjust the list of checks to include ones relevant to you.

All built-in *Kontrole jakości* are turned on by default, from where you can change these settings. By default they are commented out in *Przykładowa konfiguracja* so that default values are used. New checks then carried out for each new Weblate version.

You can turn off all checks:

```
CHECK_LIST = ()
```

You can turn on only a few:

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

Informacja: Changing this setting only affects newly changed translations, existing checks will still be stored in the database. To also apply changes to the stored translations, run `updatechecks`.

Zobacz także:

Kontrole jakości, *Dostosowywanie zachowania za pomocą flag*

2.16.16 COMMENT_CLEANUP_DAYS

Nowe w wersji 3.6.

Delete comments after a given number of days. Defaults to `None`, meaning no deletion at all.

2.16.17 COMMIT_PENDING_HOURS

Nowe w wersji 2.10.

Number of hours between committing pending changes by way of the background task.

Zobacz także:

Konfiguracja komponentu, *Wiek zmian do scommitowania*, *Uruchamianie zadań konserwacyjnych*, `commit_pending`

2.16.18 CONTACT_FORM

Nowe w wersji 4.6.

Configures how e-mail from the contact form is being sent. Choose a configuration that matches your mail server configuration.

"reply-to"
The sender is used in as `Reply-To`, this is the default behaviour.

"from"
The sender is used in as `From`. Your mail server needs to allow sending such e-mails.

2.16.19 DATA_DIR

The folder Weblate stores all data in. It contains links to VCS repositories, a fulltext index and various configuration files for external tools.

Zwykle istnieją następujące podkatalogi:

home
Home directory used for invoking scripts.

ssh
Klucze SSH i konfiguracja.

static
Default location for static Django files, specified by `STATIC_ROOT`. See *Obsługa plików statycznych*.

The Docker container uses a separate volume for this, see *Woluminy kontenerów platformy Docker*.

media
Default location for Django media files, specified by `MEDIA_ROOT`. Contains uploaded screenshots, see *Kontekst wizualny dla ciągów*.

vcs
Repozytoria kontroli wersji dla tłumaczeń.

backups
Daily backup data, please check *Zrzucone dane do kopii zapasowych* for details.

celery
Dane harmonogramu Celery, patrz: *Zadania w tle korzystające z Celery*.

fonts:
User-uploaded fonts, see *Zarządzanie czcionkami*.

Informacja: This directory has to be writable by Weblate. Running it as uWSGI means the `www-data` user should have write access to it.

The easiest way to achieve this is to make the user the owner of the directory:

```
sudo chown www-data:www-data -R $DATA_DIR
```

Wartość domyślna to `$BASE_DIR/data`.

Zobacz także:

BASE_DIR, *Uprawnienia systemu plików*, *Tworzenie kopii zapasowych i przenoszenie weblate*

2.16.20 DATABASE_BACKUP

Nowe w wersji 3.1.

Whether the database backups should be stored as plain text, compressed or skipped. The authorized values are:

- `"plain"`
- `"compressed"`
- `"none"`

Zobacz także:

Tworzenie kopii zapasowych i przenoszenie weblate

2.16.21 DEFAULT_ACCESS_CONTROL

Nowe w wersji 3.3.

The default access control setting for new projects:

0
Publiczny

1
Chroniony

100
Prywatny

200
Niestandardowy

Use *Custom* if you are managing ACL manually, which means not relying on the internal Weblate management.

Zobacz także:

Ustawienia kontroli dostępu do projektu, *Kontrola dostępu*

2.16.22 DEFAULT_AUTO_WATCH

Nowe w wersji 4.5.

Configures whether *Automatically watch projects on contribution* should be turned on for new users. Defaults to `True`.

Zobacz także:

Powiadomienia

2.16.23 DEFAULT_RESTRICTED_COMPONENT

Nowe w wersji 4.1.

Domyślna wartość ograniczenia komponentu.

Zobacz także:

Ograniczony dostęp, *Zakres grup*

2.16.24 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DEFAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DEFAULT_MERGE_MESSAGE

Default commit messages for different operations, please check *Konfiguracja komponentu* for details.

Zobacz także:

Znaczniki szablonu, *Konfiguracja komponentu*, *Commituj, dodawaj, usuwaj, scalaj, dodawaj i scalaj żądania z wiadomościami*

2.16.25 DEFAULT_ADDONS

Default add-ons to install on every created component.

Informacja: This setting affects only newly created components.

Przykład:

```
DEFAULT_ADDONS = {
    # Add-on with no parameters
    "weblate.flags.target_edit": {},
    # Add-on with parameters
    "weblate.autotranslate.autotranslate": {
        "mode": "suggest",
        "filter_type": "todo",
        "auto_source": "mt",
        "component": "",
        "engines": ["weblate-translation-memory"],
        "threshold": "80",
    },
}
```

Zobacz także:

`install_addon`, *Dodatki*, `WEBLATE_ADDONS`

2.16.26 DEFAULT_COMMITER_EMAIL

Nowe w wersji 2.4.

Committer e-mail address defaulting to `noreply@weblate.org`.

Zobacz także:

DEFAULT_COMMITER_NAME

2.16.27 DEFAULT_COMMITER_NAME

Nowe w wersji 2.4.

Domyślna nazwa commitującego to `Weblate`.

Zobacz także:

DEFAULT_COMMITER_EMAIL

2.16.28 DEFAULT_LANGUAGE

Nowe w wersji 4.3.2.

Default source language to use for example in *Język źródłowy*.

Defaults to *en*. The matching language object needs to exist in the database.

Zobacz także:

Definicje języków, *Język źródłowy*

2.16.29 DEFAULT_MERGE_STYLE

Nowe w wersji 3.4.

Merge style for any new components.

- *rebase* - domyślne
- *merge*

Zobacz także:

Konfiguracja komponentu, *Styl scalania*

2.16.30 DEFAULT_SHARED_TM

Nowe w wersji 3.2.

Configures default value of *Użyj wspólnej pamięci tłumaczeniowej* and *Przyczynia się do wspólnej pamięci tłumaczeń*.

2.16.31 DEFAULT_TRANSLATION_PROPAGATION

Nowe w wersji 2.5.

Default setting for translation propagation, defaults to `True`.

Zobacz także:

Konfiguracja komponentu, *Zezwól na propagację tłumaczenia*

2.16.32 DEFAULT_PULL_MESSAGE

Configures the default title and message for pull requests.

2.16.33 ENABLE_AVATARS

Whether to turn on Gravatar-based avatars for users. By default this is on.

Avatars are fetched and cached on the server, lowering the risk of leaking private info, speeding up the user experience.

Zobacz także:

Pamięć podręczna awatarów, `AVATAR_URL_PREFIX`, *Awatary*

2.16.34 ENABLE_HOOKS

Whether to enable anonymous remote hooks.

Zobacz także:

Hooki powiadomień

2.16.35 ENABLE_HTTPS

Whether to send links to Weblate as HTTPS or HTTP. This setting affects sent e-mails and generated absolute URLs.

In the default configuration this is also used for several Django settings related to HTTPS - it enables secure cookies, toggles HSTS or enables redirection to HTTPS URL.

The HTTPS redirection might be problematic in some cases and you might hit issue with infinite redirection in case you are using a reverse proxy doing SSL termination which does not correctly pass protocol headers to Django. Please tweak your reverse proxy configuration to emit `X-Forwarded-Proto` or `Forwarded` headers or configure `SECURE_PROXY_SSL_HEADER` to let Django correctly detect the SSL status.

Zobacz także:

`SESSION_COOKIE_SECURE`, `CSRF_COOKIE_SECURE`, `SECURE_SSL_REDIRECT`, `SECURE_PROXY_SSL_HEADER` *Ustaw prawidłową domenę strony*

2.16.36 ENABLE_SHARING

Turn on/off the *Share* menu so users can share translation progress on social networks.

2.16.37 GET_HELP_URL

Nowe w wersji 4.5.2.

URL where support for your Weblate instance can be found.

2.16.38 GITEA_CREDENTIALS

Nowe w wersji 4.12.

List for credentials for Gitea servers.

Podpowiedź: Use this in case you want Weblate to interact with more of them, for single Gitea endpoint stick with `GITEA_USERNAME` and `GITEA_TOKEN`.

```
GITEA_CREDENTIALS = {
    "try.gitea.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitea.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.39 GITEA_USERNAME

Nowe w wersji 4.12.

Nazwa użytkownika Gitea używana do wysyłania żądań wciągnięcia w celu aktualizacji tłumaczeń.

Zobacz także:

`GITEA_CREDENTIALS`, *Gitea pull requests*

2.16.40 GITEA_TOKEN

Nowe w wersji 4.12.

Osobisty token dostępu Gitea używany do wykonywania wywołań interfejsu API w celu wysyłania żądań wciągnięcia aktualizacji tłumaczeń.

Zobacz także:

`GITEA_CREDENTIALS`, *Gitea pull requests*, Tworzenie osobistego tokena dostępu Gitea

2.16.41 GITLAB_CREDENTIALS

Nowe w wersji 4.3.

List for credentials for GitLab servers.

Podpowiedź: Use this in case you want Weblate to interact with more of them, for single GitLab endpoint stick with `GITLAB_USERNAME` and `GITLAB_TOKEN`.

```
GITLAB_CREDENTIALS = {
    "gitlab.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitlab.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.42 GITLAB_USERNAME

GitLab username used to send merge requests for translation updates.

Zobacz także:

`GITLAB_CREDENTIALS`, *GitLab merge requests*

2.16.43 GITLAB_TOKEN

Nowe w wersji 4.3.

Osobisty token dostępu GitLab używany do wykonywania wywołań interfejsu API w celu wysyłania żądań scalenia aktualizacji tłumaczeń.

Zobacz także:

`GITLAB_CREDENTIALS`, *GitLab merge requests*, GitLab: Personal access token

2.16.44 GITHUB_CREDENTIALS

Nowe w wersji 4.3.

List for credentials for GitHub servers.

Podpowiedź: Use this in case you want Weblate to interact with more of them, for single GitHub endpoint stick with `GITHUB_USERNAME` and `GITHUB_TOKEN`.

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "github.example.com": {
        "username": "weblate",
        "token": "another-api-token",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    },
}
```

2.16.45 GITHUB_USERNAME

GitHub username used to send pull requests for translation updates.

Zobacz także:

`GITHUB_CREDENTIALS`, *GitHub pull requests*

2.16.46 GITHUB_TOKEN

Nowe w wersji 4.3.

GitHub personal access token used to make API calls to send pull requests for translation updates.

Zobacz także:

`GITHUB_CREDENTIALS`, *GitHub pull requests*, Tworzenie osobistego tokena dostępu GitHub

2.16.47 GOOGLE_ANALYTICS_ID

Google Analytics ID to turn on monitoring of Weblate using Google Analytics.

2.16.48 HIDE_REPO_CREDENTIALS

Hide repository credentials from the web interface. In case you have repository URL with user and password, Weblate will hide it when related info is shown to users.

For example instead of `https://user:password@git.example.com/repo.git` it will show just `https://git.example.com/repo.git`. It tries to clean up VCS error messages too in a similar manner.

Informacja: This is turned on by default.

2.16.49 HIDE_VERSION

Nowe w wersji 4.3.1.

Hides version information from unauthenticated users. This also makes all documentation links point to latest version instead of the documentation matching currently installed version.

Hiding version is recommended security practice in some corporations, but it doesn't prevent attacker to figure out version by probing the behavior.

Informacja: Ta wartość jest domyślnie wyłączona.

2.16.50 INTERLEDGER_PAYMENT_POINTERS

Nowe w wersji 4.12.1.

List of Interledger Payment Pointers (ILPs) for Web Monetization.

If multiple are specified, probabilistic revenue sharing is achieved by selecting one randomly.

Please check <https://webmonetization.org/> for more details.

Podpowiedź: The default value lets users fund Weblate itself.

2.16.51 IP_BEHIND_REVERSE_PROXY

Nowe w wersji 2.14.

Indicates whether Weblate is running behind a reverse proxy.

If set to `True`, Weblate gets IP address from a header defined by *IP_PROXY_HEADER*.

Ostrzeżenie: Ensure you are actually using a reverse proxy and that it sets this header, otherwise users will be able to fake the IP address.

Informacja: This is not on by default.

Zobacz także:

Działanie za odwrotnym proxy, *Ograniczenie szybkości*, *IP_PROXY_HEADER*, *IP_PROXY_OFFSET*

2.16.52 IP_PROXY_HEADER

Nowe w wersji 2.14.

Indicates which header Weblate should obtain the IP address from when *IP_BEHIND_REVERSE_PROXY* is turned on.

Domyślna wartość to `HTTP_X_FORWARDED_FOR`.

Zobacz także:

Działanie za odwrotnym proxy, *Ograniczenie szybkości*, `SECURE_PROXY_SSL_HEADER`, *IP_BEHIND_REVERSE_PROXY*, *IP_PROXY_OFFSET*

2.16.53 IP_PROXY_OFFSET

Nowe w wersji 2.14.

Indicates which part of *IP_PROXY_HEADER* is used as client IP address.

Depending on your setup, this header might consist of several IP addresses, (for example `X-Forwarded-For: a, b, client-ip`) and you can configure which address from the header is used as client IP address here.

Ostrzeżenie: Setting this affects the security of your installation, you should only configure it to use trusted proxies for determining IP address.

Wartość domyślna to 0.

Zobacz także:

Działanie za odwrotnym proxy, *Ograniczenie szybkości*, SECURE_PROXY_SSL_HEADER, *IP_BEHIND_REVERSE_PROXY*, *IP_PROXY_HEADER*

2.16.54 LEGAL_URL

Nowe w wersji 3.5.

URL where your Weblate instance shows its legal documents.

Podpowiedź: Useful if you host your legal documents outside Weblate for embedding them inside Weblate, please check *Informacje prawne* for details.

Przykład:

```
LEGAL_URL = "https://weblate.org/terms/"
```

Zobacz także:

PRIVACY_URL

2.16.55 LICENSE_EXTRA

Additional licenses to include in the license choices.

Informacja: Each license definition should be tuple of its short name, a long name and an URL.

Na przykład:

```
LICENSE_EXTRA = [
    (
        "AGPL-3.0",
        "GNU Affero General Public License v3.0",
        "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
    ),
]
```

2.16.56 LICENSE_FILTER

Zmienione w wersji 4.3: Setting this to blank value now disables license alert.

Filter list of licenses to show. This also disables the license alert when set to empty.

Informacja: This filter uses the short license names.

Na przykład:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

Following disables the license alert:

```
LICENSE_FILTER = set()
```

Zobacz także:

alerts

2.16.57 LICENSE_REQUIRED

Defines whether the license attribute in *Konfiguracja komponentu* is required.

Informacja: Ta opcja jest domyślnie wyłączona.

2.16.58 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Whether the length of a given translation should be limited. The restriction is the length of the source string × 10 characters.

Podpowiedź: Set this to `False` to allow longer translations (up to 10,000 characters) irrespective of source string length.

Informacja: Wartość domyślna to `True`.

2.16.59 LOCALIZE_CDN_URL i LOCALIZE_CDN_PATH

These settings configure the *Lokalizacja JavaScript CDN* add-on. *LOCALIZE_CDN_URL* defines root URL where the localization CDN is available and *LOCALIZE_CDN_PATH* defines path where Weblate should store generated files which will be served at the *LOCALIZE_CDN_URL*.

Podpowiedź: On Hosted Weblate, this uses `https://weblate-cdn.com/`.

Zobacz także:

Lokalizacja JavaScript CDN

2.16.60 LOGIN_REQUIRED_URLS

A list of URLs you want to require signing in. (Besides the standard rules built into Weblate).

Podpowiedź: This allows you to password protect a whole installation using:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

Podpowiedź: It is desirable to lock down API access as well, as shown in the above example.

Zobacz także:

REQUIRE_LOGIN

2.16.61 LOGIN_REQUIRED_URLS_EXCEPTIONS

List of exceptions for *LOGIN_REQUIRED_URLS*. If not specified, users are allowed to access the sign in page.

Some of exceptions you might want to include:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r"/accounts/(.*)$",  # Required for sign in
    r"/static/(.*)$",  # Required for development mode
    r"/widgets/(.*)$",  # Allowing public access to widgets
    r"/data/(.*)$",  # Allowing public access to data exports
    r"/hooks/(.*)$",  # Allowing public access to notification hooks
    r"/api/(.*)$",  # Allowing access to API
    r"/js/i18n/$",  # JavaScript localization
)
```

2.16.62 MATOMO_SITE_ID

ID of a site in Matomo (formerly Piwik) you want to track.

Informacja: This integration does not support the Matomo Tag Manager.

Zobacz także:

MATOMO_URL

2.16.63 MATOMO_URL

Full URL (including trailing slash) of a Matomo (formerly Piwik) installation you want to use to track Weblate use. Please check <https://matomo.org/> for more details.

Podpowiedź: This integration does not support the Matomo Tag Manager.

Na przykład:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Zobacz także:

MATOMO_SITE_ID

2.16.64 NEARBY_MESSAGES

How many strings to show around the currently translated string. This is just a default value, users can adjust this in *Profil użytkownika*.

2.16.65 DEFAULT_PAGE_LIMIT

Nowe w wersji 4.7.

Default number of elements to display when pagination is active.

2.16.66 PAGURE_CREDENTIALS

Nowe w wersji 4.3.2.

List for credentials for Pagure servers.

Podpowiedź: Use this in case you want Weblate to interact with more of them, for single Pagure endpoint stick with *PAGURE_USERNAME* and *PAGURE_TOKEN*.

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.67 PAGURE_USERNAME

Nowe w wersji 4.3.2.

Nazwa użytkownika Pagure używana do wysyłania żądań scalenia w celu aktualizacji tłumaczeń.

Zobacz także:

PAGURE_CREDENTIALS, *Pagure merge requests*

2.16.68 PAGURE_TOKEN

Nowe w wersji 4.3.2.

Osobisty token dostępu Pagure używany do wykonywania wywołań interfejsu API w celu wysyłania żądań scalenia aktualizacji tłumaczeń.

Zobacz także:

PAGURE_CREDENTIALS, *Pagure merge requests*, Pagure API

2.16.69 PRIVACY_URL

Nowe w wersji 4.8.1.

URL where your Weblate instance shows its privacy policy.

Podpowiedź: Useful if you host your legal documents outside Weblate for embedding them inside Weblate, please check *Informacje prawne* for details.

Przykład:

```
PRIVACY_URL = "https://weblate.org/terms/"
```

Zobacz także:

LEGAL_URL

2.16.70 RATELIMIT_ATTEMPTS

Nowe w wersji 3.2.

Maximum number of authentication attempts before rate limiting is applied.

Wartość domyślna to 5.

Zobacz także:

Ograniczenie szybkości, *RATELIMIT_WINDOW*, *RATELIMIT_LOCKOUT*

2.16.71 RATELIMIT_WINDOW

Nowe w wersji 3.2.

How long authentication is accepted after rate limiting applies.

An amount of seconds defaulting to 300 (5 minutes).

Zobacz także:

Ograniczenie szybkości, *RATELIMIT_ATTEMPTS*, *RATELIMIT_LOCKOUT*

2.16.72 RATELIMIT_LOCKOUT

Nowe w wersji 3.2.

How long authentication is locked after rate limiting applies.

An amount of seconds defaulting to 600 (10 minutes).

Zobacz także:

Ograniczenie szybkości, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*

2.16.73 REGISTRATION_ALLOW_BACKENDS

Nowe w wersji 4.1.

List of authentication backends to allow registration from. This only limits new registrations, users can still authenticate and add authentication using all configured authentication backends.

It is recommended to keep *REGISTRATION_OPEN* enabled while limiting registration backends, otherwise users will be able to register, but Weblate will not show links to register in the user interface.

Przykład:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

Podpowiedź: The backend names match names used in URL for authentication.

Zobacz także:

REGISTRATION_OPEN, *Uwierzytelnienie*

2.16.74 REGISTRATION_CAPTCHA

A value of either `True` or `False` indicating whether registration of new accounts is protected by CAPTCHA. This setting is optional, and a default of `True` will be assumed if it is not supplied.

If turned on, a CAPTCHA is added to all pages where a users enters their e-mail address:

- Rejestracja nowego konta.
- Odzyskanie hasła.
- Dodanie e-maila do konta.
- Contact form for users that are not signed in.

2.16.75 REGISTRATION_EMAIL_MATCH

Nowe w wersji 2.17.

Allows you to filter which e-mail addresses can register.

Defaults to `.*`, which allows any e-mail address to be registered.

You can use it to restrict registration to a single e-mail domain:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.76 REGISTRATION_OPEN

Whether registration of new accounts is currently permitted. This optional setting can remain the default `True`, or changed to `False`.

This setting affects built-in authentication by e-mail address or through the Python Social Auth (you can whitelist certain back-ends using *REGISTRATION_ALLOW_BACKENDS*).

Informacja: If using third-party authentication methods such as *Uwierzytelnianie LDAP*, it just hides the registration form, but new users might still be able to sign in and create accounts.

Zobacz także:

REGISTRATION_ALLOW_BACKENDS, *REGISTRATION_EMAIL_MATCH*, *Uwierzytelnienie*

2.16.77 REPOSITORY_ALERT_THRESHOLD

Nowe w wersji 4.0.2.

Threshold for triggering an alert for outdated repositories, or ones that contain too many changes. Defaults to 25.

Zobacz także:

alerts

2.16.78 REQUIRE_LOGIN

Nowe w wersji 4.1.

This enables *LOGIN_REQUIRED_URLS* and configures REST framework to require authentication for all API endpoints.

Informacja: This is implemented in the *Przykładowa konfiguracja*. For Docker, use *WEBLATE_REQUIRE_LOGIN*.

2.16.79 SENTRY_DSN

Nowe w wersji 3.9.

Sentry DSN to use for *Zbieranie raportów o błędach*.

Zobacz także:

Django integration for Sentry

2.16.80 SESSION_COOKIE_AGE_AUTHENTICATED

Nowe w wersji 4.3.

Set session expiry for authenticated users. This complements `SESSION_COOKIE_AGE` which is used for unauthenticated users.

Zobacz także:

`SESSION_COOKIE_AGE`

2.16.81 SIMPLIFY_LANGUAGES

Use simple language codes for default language/country combinations. For example an `fr_FR` translation will use the `fr` language code. This is usually the desired behavior, as it simplifies listing languages for these default combinations.

Turn this off if you want to different translations for each variant.

2.16.82 SITE_DOMAIN

Configures site domain. This is necessary to produce correct absolute links in many scopes (for example activation e-mails, notifications or RSS feeds).

In case Weblate is running on non-standard port, include it here as well.

Przykłady:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

Informacja: This setting should only contain the domain name. For configuring protocol, (enabling and enforcing HTTPS) use *ENABLE_HTTPS* and for changing URL, use *URL_PREFIX*.

Podpowiedź: On a Docker container, the site domain is configured through *WEBLATE_ALLOWED_HOSTS*.

Zobacz także:

Ustaw prawidłową domenę strony, *Konfiguracja dozwolonych hostów*, *Prawidłowa konfiguracja HTTPS* *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

2.16.83 SITE_TITLE

Site title to be used for the website and sent e-mails.

2.16.84 SPECIAL_CHARS

Additional characters to include in the visual keyboard, *Klawiatura wizualna*.

Wartość domyślna to:

```
SPECIAL_CHARS = ("\t", "\n", "\u00a0", "…")
```

2.16.85 SINGLE_PROJECT

Nowe w wersji 3.8.

Redirects users directly to a project or component instead of showing the dashboard. You can either set it to `True` and in this case it only works in case there is actually only single project in Weblate. Alternatively set the project slug, and it will redirect unconditionally to this project.

Zmienione w wersji 3.11: The setting now also accepts a project slug, to force displaying that single project.

Przykład:

```
SINGLE_PROJECT = "test"
```

2.16.86 SSH_EXTRA_ARGS

Nowe w wersji 4.9.

Allows to add custom parameters when Weblate is invoking SSH. This is useful when connecting to servers using legacy encryption or other non-standard features.

For example when SSH connection in Weblate fails with *Unable to negotiate with legacyhost: no matching key exchange method found. Their offer: diffie-hellman-group1-sha1*, you can enable that using:

```
SSH_EXTRA_ARGS = "-oKexAlgorithms=+diffie-hellman-group1-sha1"
```

Podpowiedź: The string is evaluated by shell, so make sure to quote any whitespace and special characters.

Zobacz także:

OpenSSH Legacy Options

2.16.87 STATUS_URL

The URL where your Weblate instance reports its status.

2.16.88 SUGGESTION_CLEANUP_DAYS

Nowe w wersji 3.2.1.

Automatically deletes suggestions after a given number of days. Defaults to `None`, meaning no deletions.

2.16.89 UPDATE_LANGUAGES

Nowe w wersji 4.3.2.

Controls whether languages database should be updated when running database migration and is enabled by default. This setting has no effect on invocation of `setuplang`.

Ostrzeżenie: The languages display might become inconsistent with this. Weblate language definitions extend over time and it will not display language code for the defined languages.

Zobacz także:

Wbudowane definicje języków

2.16.90 URL_PREFIX

This setting allows you to run Weblate under some path (otherwise it relies on being run from the webserver root).

Informacja: Aby użyć tego ustawienia, musisz również skonfigurować serwer tak, aby usuwał ten prefiks. Na przykład w przypadku WSGI można to osiągnąć, ustawiając `` WSGIScriptAlias``.

Podpowiedź: Prefiks powinien zaczynać się od `/`.

Przykład:

```
URL_PREFIX = "/translations"
```

Informacja: This setting does not work with Django's built-in server, you would have to adjust `urls.py` to contain this prefix.

2.16.91 VCS_BACKENDS

Configuration of available VCS backends.

Informacja: Weblate tries to use all supported back-ends you have the tools for.

Podpowiedź: You can limit choices or add custom VCS back-ends by using this.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

Zobacz także:

Integracja kontroli wersji

2.16.92 VCS_CLONE_DEPTH

Nowe w wersji 3.10.2.

Configures how deep cloning of repositories Weblate should do.

Informacja: Currently this is only supported in *Git*. By default Weblate does shallow clones of the repositories to make cloning faster and save disk space. Depending on your usage (for example when using custom *Dodatki*), you might want to increase the depth or turn off shallow clones completely by setting this to 0.

Podpowiedź: In case you get `fatal: protocol error: expected old/new/ref, got 'shallow <commit hash>'` error when pushing from Weblate, turn off shallow clones completely by setting:

```
VCS_CLONE_DEPTH = 0
```

2.16.93 WEBLATE_ADDONS

Lista dodatków dostępnych do użycia. Aby z nich skorzystać, muszą być włączone dla danego komponentu tłumaczeniowego. Domyślnie obejmuje to wszystkie wbudowane dodatki, podczas rozszerzania listy prawdopodobnie będziesz chciał pozostawić włączone istniejące, na przykład:

```
WEBLATE_ADDONS = (
    # Built-in add-ons
    "weblate.addons.gettext.GenerateMoAddon",
    "weblate.addons.gettext.UpdateLinguasAddon",
    "weblate.addons.gettext.UpdateConfigureAddon",
    "weblate.addons.gettext.MsgmergeAddon",
    "weblate.addons.gettext.GettextCustomizeAddon",
    "weblate.addons.gettext.GettextAuthorComments",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    "weblate.addons.cleanup.CleanupAddon",
    "weblate.addons.consistency.LangaugeConsistencyAddon",
    "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",
    # Add-on you want to include
    "weblate.addons.example.ExampleAddon",
)
```

Informacja: Removing the add-on from the list does not uninstall it from the components. Weblate will crash in that case. Please uninstall add-on from all components prior to removing it from this list.

Zobacz także:

Dodatki, `DEFAULT_ADDONS`

2.16.94 WEBLATE_EXPORTERS

Nowe w wersji 4.2.

Lista istniejących eksporterów, oferujących pobieranie tłumaczeń i słowników w różnych formatach.

Zobacz także:

:ref:»formaty»

2.16.95 WEBLATE_FORMATS

Nowe w wersji 3.0.

Lista formatów plików dostępnych do użycia.

Informacja: Domyślna lista ma już wspólne formaty.

Zobacz także:

:ref:»formaty»

2.16.96 WEBLATE_MACHINERY

Nowe w wersji 4.13.

Lista usług maszynowych dostępnych do użycia.

Zobacz także:

Konfigurowanie automatycznych sugestii

2.16.97 WEBLATE_GPG_IDENTITY

Nowe w wersji 3.1.

Tożsamość używana przez Weblate do podpisywania commitów w Git, na przykład:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

The Weblate GPG keyring is searched for a matching key (`home/.gnupg` under *DATA_DIR*). If not found, a key is generated, please check *Signing Git commits with GnuPG* for more details.

Zobacz także:

Signing Git commits with GnuPG

2.16.98 WEBSITE_REQUIRED

Defines whether *Strona internetowa projektu* has to be specified when creating a project. Turned on by default as that suits public server setups.

2.17 Przykładowa konfiguracja

The following example is shipped as `weblate/settings_example.py` with Weblate:

```
#
# Copyright © 2012-2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#


import os
import platform
from logging.handlers import SysLogHandler

# Title of site to use
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
SITE_TITLE = "Weblate"

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

#
# Django settings for Weblate project.
#

DEBUG = True

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password.
        "PASSWORD": "",
        # Set to empty string for localhost.
        "HOST": "127.0.0.1",
        # Set to empty string for default.
        "PORT": "",
        # Customizations for databases.
        "OPTIONS": {
            # In case of using an older MySQL server,
            # which has MyISAM as a default storage
            # "init_command": "SET storage_engine=INNODB",
            # Uncomment for MySQL older than 5.7:
            # "init_command": "SET sql_mode='STRICT_TRANS_TABLES'",
            # Set emoji capable charset for MySQL:
            # "charset": "utf8mb4",
            # Change connection timeout in case you get MySQL gone away error:
            # "connect_timeout": 28800,
        },
        # Persistent connections
        "CONN_MAX_AGE": 0,
        # Disable server-side cursors, might be needed with pgbouncer
        "DISABLE_SERVER_SIDE_CURSORS": False,
    }
}

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Data directory
DATA_DIR = os.path.join(BASE_DIR, "data")
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name
# although not all choices may be available on all operating systems.
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = "UTC"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

LANGUAGES = (
    ("ar", "ةيبرعلا"),
    ("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", "□□□"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "□□□"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("th", "□□□"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "□□□□"),
    ("zh-hant", "□□□□"),
)

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
# to load the internationalization machinery.
USE_I18N = True

# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
USE_L10N = True

# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True

# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(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 = ""

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",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
                "weblate.trans.context_processors.weblate_context",
            ],
        },
        "APP_DIRS": True,
    }
]


# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_USERNAME = None
GITLAB_TOKEN = None

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth2",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

SOCIAL_AUTH_GITHUB_ORG_KEY = ""
SOCIAL_AUTH_GITHUB_ORG_SECRET = ""
SOCIAL_AUTH_GITHUB_ORG_NAME = ""

SOCIAL_AUTH_GITHUB_TEAM_KEY = ""
SOCIAL_AUTH_GITHUB_TEAM_SECRET = ""
SOCIAL_AUTH_GITHUB_TEAM_ID = ""

SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    "social_core.pipeline.social_auth.social_details",
    "social_core.pipeline.social_auth.social_uid",
    "social_core.pipeline.social_auth.auth_allowed",
    "social_core.pipeline.social_auth.social_user",
    "weblate.accounts.pipeline.store_params",
    "weblate.accounts.pipeline.verify_open",
    "social_core.pipeline.user.get_username",
    "weblate.accounts.pipeline.require_email",
    "social_core.pipeline.mail.mail_validation",
    "weblate.accounts.pipeline.revoke_mail_code",
    "weblate.accounts.pipeline.ensure_valid",
    "weblate.accounts.pipeline.remove_account",
    "social_core.pipeline.social_auth.associate_by_email",
    "weblate.accounts.pipeline.reauthenticate",
    "weblate.accounts.pipeline.verify_username",
    "social_core.pipeline.user.create_user",
    "social_core.pipeline.social_auth.associate_user",
    "social_core.pipeline.social_auth.load_extra_data",
    "weblate.accounts.pipeline.cleanup_next",
    "weblate.accounts.pipeline.user_full_name",
    "weblate.accounts.pipeline.store_email",
    "weblate.accounts.pipeline.notify_connect",
    "weblate.accounts.pipeline.password_reset",
)
SOCIAL_AUTH_DISCONNECT_PIPELINE = (
    "social_core.pipeline.disconnect.allowed_to_disconnect",
    "social_core.pipeline.disconnect.get_entries",
    "social_core.pipeline.disconnect.revoke_tokens",
    "weblate.accounts.pipeline.cycle_session",
    "weblate.accounts.pipeline.adjust_primary_mail",
    "weblate.accounts.pipeline.notify_disconnect",
    "social_core.pipeline.disconnect.disconnect",
    "weblate.accounts.pipeline.cleanup_next",
)

# Custom authentication strategy
SOCIAL_AUTH_STRATEGY = "weblate.accounts.strategy.WeblateStrategy"

# Raise exceptions so that we can handle them later
SOCIAL_AUTH_RAISE_EXCEPTIONS = True

SOCIAL_AUTH_EMAIL_VALIDATION_FUNCTION = "weblate.accounts.pipeline.send_validation"
SOCIAL_AUTH_EMAIL_VALIDATION_URL = f"{URL_PREFIX}/accounts/email-sent/"
SOCIAL_AUTH_LOGIN_ERROR_URL = f"{URL_PREFIX}/accounts/login/"
SOCIAL_AUTH_EMAIL_FORM_URL = f"{URL_PREFIX}/accounts/email/"
SOCIAL_AUTH_NEW_ASSOCIATION_REDIRECT_URL = f"{URL_PREFIX}/accounts/profile/#account
↪"
SOCIAL_AUTH_PROTECTED_USER_FIELDS = ("email",)
SOCIAL_AUTH_SLUGIFY_USERNAMES = True
SOCIAL_AUTH_SLUGIFY_FUNCTION = "weblate.accounts.pipeline.slugify_username"

# Password validation configuration
AUTH_PASSWORD_VALIDATORS = [
    {
        "NAME": "django.contrib.auth.password_validation.
↪UserAttributeSimilarityValidator"  # noqa: E501, pylint: disable=line-too-long
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.CharsPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.PastPasswordsValidator"},
    # Optional password strength validation by django-zxcvbn-password
    # {
    #     "NAME": "zxcvbn_password.ZXCVBNValidator",
    #     "OPTIONS": {
    #         "min_score": 3,
    #         "user_attributes": ("username", "email", "full_name")
    #     }
    # },
]

# Password hashing (prefer Argon)
PASSWORD_HASHERS = [
    "django.contrib.auth.hashers.Argon2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher",
    "django.contrib.auth.hashers.BCryptSHA256PasswordHasher",
]

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
    "social_django.middleware.SocialAuthExceptionMiddleware",
    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
    "weblate.wladmin.middleware.ManageMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",
    "weblate.screenshots",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
    "weblate.gitexport",
    # Standard Django modules
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "django.contrib.admin.apps.SimpleAdminConfig",
    "django.contrib.admindocs",
    "django.contrib.sitemaps",
    "django.contrib.humanize",
    # Third party Django modules
    "social_django",
    "crispy_forms",
    "compressor",
    "rest_framework",
    "rest_framework.authtoken",
    "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 OSError:
        HAVE_SYSLOG = False

if DEBUG or not HAVE_SYSLOG:
    DEFAULT_LOG = "console"
else:
    DEFAULT_LOG = "syslog"
DEFAULT_LOGLEVEL = "DEBUG" if DEBUG else "INFO"

# A sample logging configuration. The only tangible logging
# performed by this configuration is to send an email to
# the site admins on every HTTP 500 error when DEBUG=False.
# See http://docs.djangoproject.com/en/stable/topics/logging for
# more details on how to customize your logging configuration.
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
LOGGING = {
    "version": 1,
    "disable_existing_loggers": True,
    "filters": {"require_debug_false": {"()": "django.utils.log.RequireDebugFalse"}
↪},
    "formatters": {
        "syslog": {"format": "weblate[%(process)d]: %(levelname)s %(message)s"},
        "simple": {"format": "[%(asctime)s: %(levelname)s/%(process)s] %(message)s
↪"},
        "logfile": {"format": "%(asctime)s %(levelname)s %(message)s"},
        "django.server": {
            "()": "django.utils.log.ServerFormatter",
            "format": "[%(server_time)s] %(message)s",
        },
    },
    "handlers": {
        "mail_admins": {
            "level": "ERROR",
            "filters": ["require_debug_false"],
            "class": "django.utils.log.AdminEmailHandler",
            "include_html": True,
        },
        "console": {
            "level": "DEBUG",
            "class": "logging.StreamHandler",
            "formatter": "simple",
        },
        "django.server": {
            "level": "INFO",
            "class": "logging.StreamHandler",
            "formatter": "django.server",
        },
        "syslog": {
            "level": "DEBUG",
            "class": "logging.handlers.SysLogHandler",
            "formatter": "syslog",
            "address": "/dev/log",
            "facility": SysLogHandler.LOG_LOCAL2,
        },
        # 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
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
        # "django.db.backends": {
        #     "handlers": [DEFAULT_LOG],
        #     "level": "DEBUG",
        # },
        "weblate": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # Logging VCS operations
        "weblate.vcs": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # Python Social Auth
        "social": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # Django Authentication Using LDAP
        "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # SAML IdP
        "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    },
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    #     "weblate.machinery.apertium.ApertiumAPYTranslation",
    #     "weblate.machinery.baidu.BaiduTranslation",
    #     "weblate.machinery.deepl.DeepLTranslation",
    #     "weblate.machinery.glosbe.GlosbeTranslation",
    #     "weblate.machinery.google.GoogleTranslation",
    #     "weblate.machinery.googlev3.GoogleV3Translation",
    #     "weblate.machinery.libretranslate.LibreTranslateTranslation",
    #     "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    #     "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    #     "weblate.machinery.modernmt.ModernMTTranslation",
    #     "weblate.machinery.mymemory.MyMemoryTranslation",
    #     "weblate.machinery.netease.NeteaseSightTranslation",
    #     "weblate.machinery.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

# LibreTranslate
MT_LIBRETRANSLATE_API_URL = None
MT_LIBRETRANSLATE_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

# ModernMT
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
MT_MODERNMT_KEY = None

# MyMemory identification email, see
# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
MT_GOOGLE_CREDENTIALS = None
MT_GOOGLE_PROJECT = None

# Baidu app key and secret
MT_BAIDU_ID = None
MT_BAIDU_SECRET = None

# Youdao Zhiyun app key and secret
MT_YOUDAO_ID = None
MT_YOUDAO_SECRET = None

# Netease Sight (Jianwai) app key and secret
MT_NETEASE_KEY = None
MT_NETEASE_SECRET = None

# API key for Yandex Translate API
MT_YANDEX_KEY = None

# tmserver URL
MT_TMSERVER = None

# SAP Translation Hub
MT_SAP_BASE_URL = None
MT_SAP_SANDBOX_APIKEY = None
MT_SAP_USERNAME = None
MT_SAP_PASSWORD = None
MT_SAP_USE_MT = True

# Use HTTPS when creating redirect URLs for social authentication, see
# documentation for more details:
# https://python-social-auth-docs.readthedocs.io/en/latest/configuration/settings.
↪html#processing-redirects-and-urlopen
SOCIAL_AUTH_REDIRECT_IS_HTTPS = ENABLE_HTTPS

# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view
CSRF_FAILURE_VIEW = "weblate.trans.views.error.csrf_failure"
SESSION_COOKIE_SECURE = ENABLE_HTTPS
SESSION_COOKIE_HTTPONLY = True
# SSL redirect
SECURE_SSL_REDIRECT = ENABLE_HTTPS
# Sent referrrer only for same origin links
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",)  # Allowing HTTP access to health check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
SESSION_COOKIE_SAMESITE = "Lax"
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer
# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
SIMPLIFY_LANGUAGES = True

# Render forms using bootstrap
CRISPY_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
# CHECK_LIST = (
#     "weblate.checks.same.SameCheck",
#     "weblate.checks.chars.BeginNewlineCheck",
#     "weblate.checks.chars.EndNewlineCheck",
#     "weblate.checks.chars.BeginSpaceCheck",
#     "weblate.checks.chars.EndSpaceCheck",
#     "weblate.checks.chars.DoubleSpaceCheck",
#     "weblate.checks.chars.EndStopCheck",
#     "weblate.checks.chars.EndColonCheck",
#     "weblate.checks.chars.EndQuestionCheck",
#     "weblate.checks.chars.EndExclamationCheck",
#     "weblate.checks.chars.EndEllipsisCheck",
#     "weblate.checks.chars.EndSemicolonCheck",
#     "weblate.checks.chars.MaxLengthCheck",
#     "weblate.checks.chars.KashidaCheck",
#     "weblate.checks.chars.PunctuationSpacingCheck",
#     "weblate.checks.format.PythonFormatCheck",
#     "weblate.checks.format.PythonBraceFormatCheck",
#     "weblate.checks.format.PHPFormatCheck",
#     "weblate.checks.format.CFormatCheck",
#     "weblate.checks.format.PerlFormatCheck",
#     "weblate.checks.format.JavaScriptFormatCheck",
#     "weblate.checks.format.LuaFormatCheck",
#     "weblate.checks.format.ObjectPascalFormatCheck",
#     "weblate.checks.format.SchemeFormatCheck",
#     "weblate.checks.format.CSharpFormatCheck",
#     "weblate.checks.format.JavaFormatCheck",
#     "weblate.checks.format.JavaMessageFormatCheck",
#     "weblate.checks.format.PercentPlaceholdersCheck",
#     "weblate.checks.format.VueFormattingCheck",
#     "weblate.checks.format.I18NextInterpolationCheck",
#     "weblate.checks.format.ESTemplateLiteralsCheck",
#     "weblate.checks.angularjs.AngularJSInterpolationCheck",
#     "weblate.checks.icu.ICUMessageFormatCheck",
#     "weblate.checks.icu.ICUSourceCheck",
#     "weblate.checks.qt.QtFormatCheck",
#     "weblate.checks.qt.QtPluralCheck",
#     "weblate.checks.ruby.RubyFormatCheck",
#     "weblate.checks.consistency.PluralsCheck",
#     "weblate.checks.consistency.SamePluralsCheck",
#     "weblate.checks.consistency.ConsistencyCheck",
#     "weblate.checks.consistency.TranslatedCheck",
#     "weblate.checks.chars.EscapedNewlineCountingCheck",
#     "weblate.checks.chars.NewLineCountCheck",
#     "weblate.checks.markup.BBCodeCheck",
#     "weblate.checks.chars.ZeroWidthSpaceCheck",
#     "weblate.checks.render.MaxSizeCheck",
#     "weblate.checks.markup.XMLValidityCheck",
#     "weblate.checks.markup.XMLTagsCheck",
#     "weblate.checks.markup.MarkdownRefLinkCheck",
#     "weblate.checks.markup.MarkdownLinkCheck",
#     "weblate.checks.markup.MarkdownSyntaxCheck",
#     "weblate.checks.markup.URLCheck",
#     "weblate.checks.markup.SafeHTMLCheck",
#     "weblate.checks.placeholders.PlaceholderCheck",
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
#     "weblate.checks.placeholders.RegexCheck",
#     "weblate.checks.duplicate.DuplicateCheck",
#     "weblate.checks.source.OptionalPluralCheck",
#     "weblate.checks.source.EllipsisCheck",
#     "weblate.checks.source.MultipleFailingCheck",
#     "weblate.checks.source.LongUntranslatedCheck",
#     "weblate.checks.format.MultipleUnnamedFormatsCheck",
#     "weblate.checks.glossary.GlossaryCheck",
# )

# List of automatic fixups
# AUTOFIX_LIST = (
#     "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
#     "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
#     "weblate.trans.autofixes.chars.RemoveZeroSpace",
#     "weblate.trans.autofixes.chars.RemoveControlChars",
# )

# List of enabled addons
# WEBLATE_ADDONS = (
#     "weblate.addons.gettext.GenerateMoAddon",
#     "weblate.addons.gettext.UpdateLinguasAddon",
#     "weblate.addons.gettext.UpdateConfigureAddon",
#     "weblate.addons.gettext.MsgmergeAddon",
#     "weblate.addons.gettext.GettextCustomizeAddon",
#     "weblate.addons.gettext.GettextAuthorComments",
#     "weblate.addons.cleanup.CleanupAddon",
#     "weblate.addons.cleanup.RemoveBlankAddon",
#     "weblate.addons.consistency.LangaugeConsistencyAddon",
#     "weblate.addons.discovery.DiscoveryAddon",
#     "weblate.addons.autotranslate.AutoTranslateAddon",
#     "weblate.addons.flags.SourceEditAddon",
#     "weblate.addons.flags.TargetEditAddon",
#     "weblate.addons.flags.SameEditAddon",
#     "weblate.addons.flags.BulkEditAddon",
#     "weblate.addons.generate.GenerateFileAddon",
#     "weblate.addons.generate.PseudolocaleAddon",
#     "weblate.addons.generate.PrefillAddon",
#     "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",
# )

# E-mail address that error messages come from.
SERVER_EMAIL = "noreply@example.com"

# Default email address to use for various automated correspondence from
# the site managers. Used for registration emails.
DEFAULT_FROM_EMAIL = "noreply@example.com"

# List of URLs your site is supposed to serve
ALLOWED_HOSTS = ["*"]

# Configuration for caching
CACHES = {
    "default": {
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/1",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # "LOCATION": "unix:///var/run/redis/redis.sock?db=1",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
            # If you set password here, adjust CELERY_BROKER_URL as well
            "PASSWORD": None,
            "CONNECTION_POOL_KWARGS": {},
        },
        "KEY_PREFIX": "weblate",
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 86400,
        "OPTIONS": {"MAX_ENTRIES": 1000},
    },
}

# Store sessions in cache
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
# Store messages in session
MESSAGE_STORAGE = "django.contrib.messages.storage.session.SessionStorage"

# REST framework settings for API
REST_FRAMEWORK = {
    # Use Django's standard `django.contrib.auth` permissions,
    # or allow read-only access for unauthenticated users.
    "DEFAULT_PERMISSION_CLASSES": [
        # Require authentication for login required sites
        "rest_framework.permissions.IsAuthenticated"
        if REQUIRE_LOGIN
        else "rest_framework.permissions.IsAuthenticatedOrReadOnly"
    ],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework.authentication.TokenAuthentication",
        "weblate.api.authentication.BearerAuthentication",
        "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},
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI":␣
↪False},
]

# Require login for all URLs
if REQUIRE_LOGIN:
    LOGIN_REQUIRED_URLS = (r"/(.*)$",)

# In such case you will want to include some of the exceptions
# LOGIN_REQUIRED_URLS_EXCEPTIONS = (
#    rf"{URL_PREFIX}/accounts/(.*)$",  # Required for login
#    rf"{URL_PREFIX}/admin/login/(.*)$",  # Required for admin login
#    rf"{URL_PREFIX}/static/(.*)$",  # Required for development mode
#    rf"{URL_PREFIX}/widgets/(.*)$",  # Allowing public access to widgets
#    rf"{URL_PREFIX}/data/(.*)$",  # Allowing public access to data exports
#    rf"{URL_PREFIX}/hooks/(.*)$",  # Allowing public access to notification hooks
#    rf"{URL_PREFIX}/healthz/$",  # Allowing public access to health check
#    rf"{URL_PREFIX}/api/(.*)$",  # Allowing access to API
#    rf"{URL_PREFIX}/js/i18n/$",  # JavaScript localization
#    rf"{URL_PREFIX}/contact/$",  # Optional for contact form
#    rf"{URL_PREFIX}/legal/(.*)$",  # Optional for legal app
#    rf"{URL_PREFIX}/avatar/(.*)$",  # Optional for avatars
# )

# Silence some of the Django system checks
SILENCED_SYSTEM_CHECKS = [
    # We have modified django.contrib.auth.middleware.AuthenticationMiddleware
    # as weblate.accounts.middleware.AuthenticationMiddleware
    "admin.E408"
]

# Celery worker configuration for testing
# CELERY_TASK_ALWAYS_EAGER = True
# CELERY_BROKER_URL = "memory://"
# CELERY_TASK_EAGER_PROPAGATES = True
# Celery worker configuration for production
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

# Celery settings, it is not recommended to change these
CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_SCHEDULE_FILENAME = os.path.join(DATA_DIR, "celery", "beat-schedule")
CELERY_TASK_ROUTES = {
    "weblate.trans.tasks.auto_translate*": {"queue": "translate"},
    "weblate.accounts.tasks.notify_*": {"queue": "notify"},
    "weblate.accounts.tasks.send_mails": {"queue": "notify"},
    "weblate.utils.tasks.settings_backup": {"queue": "backup"},
    "weblate.utils.tasks.database_backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup_service": {"queue": "backup"},
    "weblate.memory.tasks.*": {"queue": "memory"},
}

# Enable plain database backups
DATABASE_BACKUP = "plain"

# Enable auto updating
AUTO_UPDATE = False

# PGP commits signing
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
WEBLATE_GPG_IDENTITY = None

# Third party services integration
MATOMO_SITE_ID = None
MATOMO_URL = None
GOOGLE_ANALYTICS_ID = None
SENTRY_DSN = None
SENTRY_ENVIRONMENT = SITE_DOMAIN
AKISMET_API_KEY = None
```

2.18 Polecenia zarządzania

Informacja: Running management commands under a different user than the one running your webserver can result in files getting wrong permissions, please check *Uprawnienia systemu plików* for more details.

You will find basic management commands (available as `./manage.py` in the Django sources, or as an extended set in a script called **`weblate`** installable atop Weblate).

2.18.1 Wywoływanie poleceń zarządzania

As mentioned before, invocation depends on how you installed Weblate.

If using virtualenv for Weblate, you can either specify the full path to **`weblate`**, or activate the virtualenv prior to invoking it:

```
# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

If you are using source code directly (either from a tarball or Git checkout), the management script is `./manage.py` available in the Weblate sources. To run it:

```
python ./manage.py list_versions
```

If you've installed Weblate using the pip or pip3 installer, or by using the `./setup.py` script, the **`weblate`** is installed to your path (or virtualenv path), from where you can use it to control Weblate:

```
weblate list_versions
```

For the Docker image, the script is installed like above, and you can run it using **`docker exec`**:

```
docker exec --user weblate <container> weblate list_versions
```

For **`docker-compose`** the process is similar, you just have to use **`docker-compose exec`**:

```
docker-compose exec --user weblate weblate weblate list_versions
```

In case you need to pass it a file, you can temporary add a volume:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /tmp/
↪users.json
```

Zobacz także:

Instalowanie za pomocą Dockera, *Instalacja na Debianie i Ubuntu*, *Instalacja na SUSE i openSUSE*, *Installing on RedHat, Fedora and CentOS*, *Instalacja ze źródeł*

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

Nowe w wersji 2.5.

Imports a translation from the file to use as a suggestion for the given translation. It skips duplicated translations; only different ones are added.

--author `USER@EXAMPLE.COM`

E-mail of author for the suggestions. This user has to exist prior to importing (you can create one in the admin interface if needed).

Przykład:

```
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>

Nowe w wersji 2.5.

Zmienione w wersji 4.6: Dodano parametr dla trybu tłumaczenia.

Performs automatic translation based on other component translations.

--source `PROJECT/COMPONENT`

Specifies the component to use as source available for translation. If not specified all components in the project are used.

--user `USERNAME`

Specify username listed as author of the translations. „Anonymous user” is used if not specified.

--overwrite

Czy zastąpić istniejące tłumaczenia.

--inconsistent

Whether to overwrite existing translations that are inconsistent (see *Niespójność*).

--add

Automatically add language if a given translation does not exist.

--mt `MT`

Use machine translation instead of other components as machine translations.

--threshold `THRESHOLD`

Similarity threshold for machine translation, defaults to 80.

--mode `MODE`

Specify translation mode, default is `translate` but `fuzzy` or `suggest` can be used.

Przykład:

```
weblate auto_translate --user nijel --inconsistent --source weblate/application␣
↪weblate website cs
```

Zobacz także:

Tłumaczenie automatyczne

2.18.4 celery_queues

weblate celery_queues

Nowe w wersji 3.7.

Displays length of Celery task queues.

2.18.5 checkgit

weblate checkgit <project|project/component>

Prints current state of the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

2.18.6 commitgit

weblate commitgit <project|project/component>

Commits any possible pending changes to the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

2.18.7 commit_pending

weblate commit_pending <project|project/component>

Commits pending changes older than a given age.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

--age `HOURS`

Age in hours for committing. If not specified the value configured in *Konfiguracja komponentu* is used.

Informacja: This is automatically performed in the background by Weblate, so there no real need to invoke this manually, besides forcing an earlier commit than specified by *Konfiguracja komponentu*.

Zobacz także:

Uruchamianie zadań konserwacyjnych, `COMMIT_PENDING_HOURS`

2.18.8 cleanuptrans

weblate cleanuptrans

Cleans up orphaned checks and translation suggestions. There is normally no need to run this manually, as the cleanups happen automatically in the background.

Zobacz także:

Uruchamianie zadań konserwacyjnych

2.18.9 cleanup_ssh_keys

weblate cleanup_ssh_keys

Nowe w wersji 4.9.1.

Performs cleanup of stored SSH host keys:

- Removes deprecated RSA keys for GitHub which might cause issues connecting to GitHub.
- Removes duplicate entries in host keys.

Zobacz także:

Repozytoria SSH

2.18.10 createadmin

weblate createadmin

Creates an `admin` account with a random password, unless it is specified.

--password `PASSWORD`
: Provides a password on the command-line, to not generate a random one.

--no-password
: Do not set password, this can be useful with *–update*.

--username `USERNAME`
: Use the given name instead of `admin`.

--email `USER@EXAMPLE.COM`
: Określ adres e-mail administratora.

--name
: Określ nazwę administratora (widoczną).

--update
: Update the existing user (you can use this to change passwords).

Zmienione w wersji 2.9: Added parameters `--username`, `--email`, `--name` and `--update`.

2.18.11 dump_memory

weblate dump_memory

Nowe w wersji 2.20.

Export a JSON file containing Weblate Translation Memory content.

Zobacz także:

Pamięć tłumaczeniowa, *Schemat pamięci tłumaczeniowej Weblate*

2.18.12 dumpuserdata

weblate dumpuserdata <file.json>

Dumps userdata to a file for later use by `importuserdata`.

Podpowiedź: This comes in handy when migrating or merging Weblate instances.

2.18.13 import_demo

weblate import_demo

Nowe w wersji 4.1.

Creates a demo project with components based on <https://github.com/WeblateOrg/demo>. Make sure the celery tasks are running before running this command.

This can be useful when developing Weblate.

2.18.14 import_json

weblate import_json <json-file>

Nowe w wersji 2.7.

Batch import of components based on JSON data.

The imported JSON file structure pretty much corresponds to the component object (see `GET /api/components/(string:project)/(string:component)/`). You have to include the `name` and `filemask` fields.

--project PROJECT

Specifies where the components will be imported from.

--main-component COMPONENT

Use the given VCS repository from this component for all of them.

--ignore

Pomiń (już) zaimportowane komponenty.

--update

Zaktualizuj (już) zaimportowane komponenty.

Zmienione w wersji 2.9: The parameters `--ignore` and `--update` are there to deal with already imported components.

Przykład pliku 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"
  }
]
```

Zobacz także:

import_memory

2.18.15 import_memory

weblate import_memory <file>

Nowe w wersji 2.20.

Imports a TMX or JSON file into the Weblate translation memory.

--language-map LANGMAP

Allows mapping languages in the TMX to the Weblate translation memory. The language codes are mapped after normalization usually done by Weblate.

`--language-map en_US:en` will for example import all `en_US` strings as `en` ones.

This can be useful in case your TMX file locales happen not to match what you use in Weblate.

Zobacz także:

Pamięć tłumaczeniowa, *Schemat pamięci tłumaczeniowej Weblate*

2.18.16 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

Zmienione w wersji 3.0: The import_project command is now based on the *Wykrycie komponentów* add-on, leading to some changes in behavior and what parameters are accepted.

Batch imports components into project based on the file mask.

<project> names an existing project, into which the components are to be imported.

The *<gitrepo>* defines the Git repository URL to use, and *<branch>* signifies the Git branch. To import additional translation components from an existing Weblate component, use a *weblate://<project>/<component>* URL for the *<gitrepo>*.

The *<filemask>* defines file discovery for the repository. It can be either be made simple using wildcards, or it can use the full power of regular expressions.

The simple matching uses `**` for component name and `*` for language, for example: `**/*.po`

The regular expression has to contain groups named *component* and *language*. For example: `(?P<language>[^/]*)/(?P<component>[^-/]*)\.po`

The import matches existing components based on files and adds the ones that do not exist. It does not change already existing ones.

--name-template TEMPLATE

Customize the name of a component using Django template syntax.

For example: `Documentation: {{ component }}`

--base-file-template TEMPLATE

Customize the base file for monolingual translations.

Na przykład: `{{ component }}/res/values/string.xml`

--new-base-template TEMPLATE

Customize the base file for addition of new translations.

Na przykład: `{{ component }}/ts/en.ts`

--file-format FORMAT

You can also specify the file format to use (see *Obsługiwane formaty plików*), the default is auto-detection.

--language-regex REGEX

You can specify language filtering (see *Konfiguracja komponentu*) with this parameter. It has to be a valid regular expression.

--main-component

You can specify which component will be chosen as the main one—the one actually containing the VCS repository.

--license NAME

Specify the overall, project or component translation license.

--license-url URL

Specify the URL where the translation license is to be found.

--vcs NAME

In case you need to specify which version control system to use, you can do it here. The default version control is Git.

To give you some examples, let's try importing two projects.

First The Debian Handbook translations, where each language has separate a folder with the translations of each chapter:

```
weblate import_project \
    debian-handbook \
    git://anonscm.debian.org/debian-handbook/debian-handbook.git \
    squeeze/master \
    '*/**.po'
```

Then the Tanaguru tool, where the file format needs be specified, along with the base file template, and how all components and translations are located in single folder:

```
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
```

More complex example of parsing of filenames to get the correct component and language out of a filename like `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$'
```

Filtrowanie tylko tłumaczeń w wybranym języku:

```
./manage import_project \
    --language-regex '^(cs|sk)$' \
    weblate \
    https://github.com/WeblateOrg/weblate.git \
    'weblate/locale/*/LC_MESSAGES/**.po'
```

Importowanie dokumentacji Sphinx podzielonej na wiele plików:

```
$ weblate import_project --name-template 'Documentation: %s' \
    --file-format po \
    project https://github.com/project/docs.git master \
    'docs/locale/*/LC_MESSAGES/**.po'
```

Importing Sphinx documentation split to multiple files and directories:

```
$ 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'
```

Zobacz także:

More detailed examples can be found in the starting chapter, alternatively you might want to use *import_json*.

2.18.17 importuserdata

weblate importuserdata <file.json>

Imports user data from a file created by *dumpuserdata*

2.18.18 importusers

weblate importusers --check <file.json>

Imports users from JSON dump of the Django auth_users database.

--check

With this option it will just check whether a given file can be imported and report possible conflicts arising from usernames or e-mails.

You can dump users from the existing Django installation using:

```
weblate dumpdata auth.User > users.json
```

2.18.19 install_addon

Nowe w wersji 3.2.

weblate install_addon --addon ADDON <project|project/component>

Installs an add-on to a set of components.

--addon ADDON
: Name of the add-on to install. For example `weblate.gettext.customize`.

--configuration CONFIG
: Konfiguracja dodatku zakodowana w formacie JSON.

--update
: Zaktualizuj istniejącą konfigurację dodatku.

You can either define which project or component to install the add-on in (for example `weblate/application`), or use `--all` to include all existing components.

To install *Dostosuj wyjście gettext* for all components:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --
↪update --all
```

Zobacz także:

Dodatki

2.18.20 list_languages

weblate list_languages <locale>

Lists supported languages in MediaWiki markup - language codes, English names and localized names.

This is used to generate <https://wiki.l10n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF>.

2.18.21 list_translators

weblate list_translators <project|project/component>

Lists translators by contributed language for the given project:

```
[French]
Jean Dupont <jean.dupont@example.com>
[English]
John Doe <jd@example.com>
```

--language-code
: List names by language code instead of language name.

You can either define which project or component to use (for example `weblate/application`), or use `--all` to list translators from all existing components.

2.18.22 list_versions

weblate list_versions

Lists all Weblate dependencies and their versions.

2.18.23 loadpo

weblate loadpo <project|project/component>

Reloads translations from disk (for example in case you have done some updates in the VCS repository).

--force

Force update, even if the files should be up-to-date.

--lang LANGUAGE

Limit processing to a single language.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Informacja: You seldom need to invoke this, Weblate will automatically load changed files for every VCS update. This is needed in case you manually changed an underlying Weblate VCS repository or in some special cases following an upgrade.

2.18.24 lock_translation

weblate lock_translation <project|project/component>

Prevents further translation of a component.

Podpowiedź: Useful in case you want to do some maintenance on the underlying repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Zobacz także:

unlock_translation

2.18.25 move_language

weblate move_language source target

Nowe w wersji 3.0.

Allows you to merge language content. This is useful when updating to a new version which contains aliases for previously unknown languages that have been created with the *(generated)* suffix. It moves all content from the *source* language to the *target* one.

Przykład:

```
weblate move_language cze cs
```

After moving the content, you should check whether there is anything left (this is subject to race conditions when somebody updates the repository meanwhile) and remove the *(generated)* language.

2.18.26 pushgit

weblate pushgit <project|project/component>

Pushes committed changes to the upstream VCS repository.

--force-commit

Force commits any pending changes, prior to pushing.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Informacja: Weblate pushes changes automatically if *Przesyłaj przy commitowaniu* in *Konfiguracja komponentu* is turned on, which is the default.

2.18.27 unlock_translation

weblate unlock_translation <project|project/component>

Unlocks a given component, making it available for translation.

Podpowiedź: Useful in case you want to do some maintenance on the underlying repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Zobacz także:

`lock_translation`

2.18.28 setupgroups

weblate setupgroups

Configures default groups and optionally assigns all users to that default group.

--no-privs-update

Turns off automatic updating of existing groups (only adds new ones).

--no-projects-update

Prevents automatic updates of groups for existing projects. This allows adding newly added groups to existing projects, see *Ustawienia kontroli dostępu do projektu*.

Zobacz także:

Lista uprawnień i wbudowanych ról

2.18.29 setuplang

weblate setuplang

Updates list of defined languages in Weblate.

--no-update

Turns off automatic updates of existing languages (only adds new ones).

2.18.30 updatechecks

weblate updatechecks <project|project/component>

Updates all checks for all strings.

Podpowiedź: Useful for upgrades which do major changes to checks.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

2.18.31 updategit

weblate updategit <project|project/component>

Fetches remote VCS repositories and updates the internal cache.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Informacja: Usually it is better to configure hooks in the repository to trigger *Hooki powiadomień*, instead of regular polling by `updategit`.

2.19 Ogłoszenia

Zmienione w wersji 4.0: In prior releases this feature was called whiteboard messages.

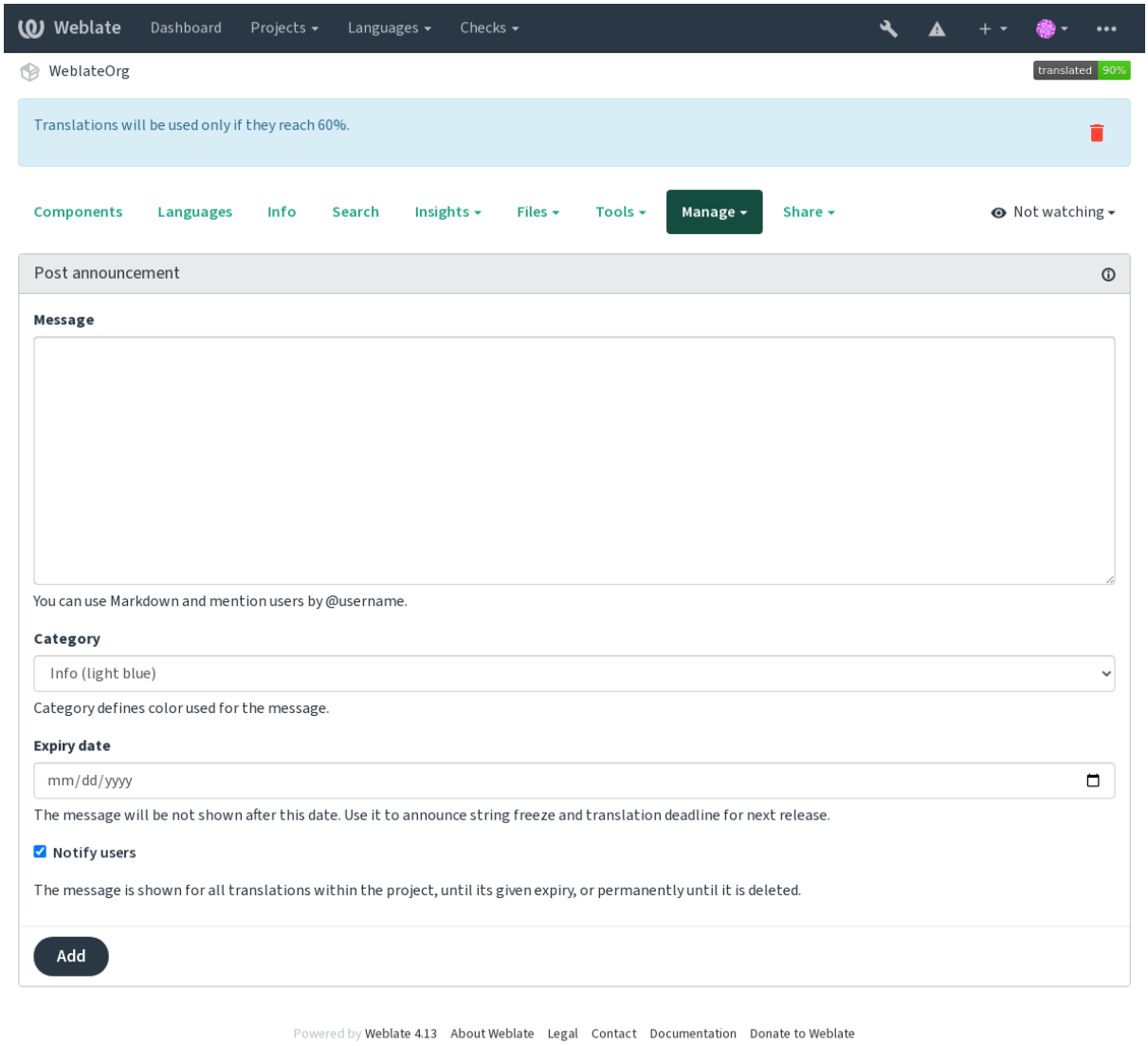
Provide info to your translators by posting announcements, site-wide, per project, component, or language.

Announce the purpose, deadlines, status, or specify targets for translation.

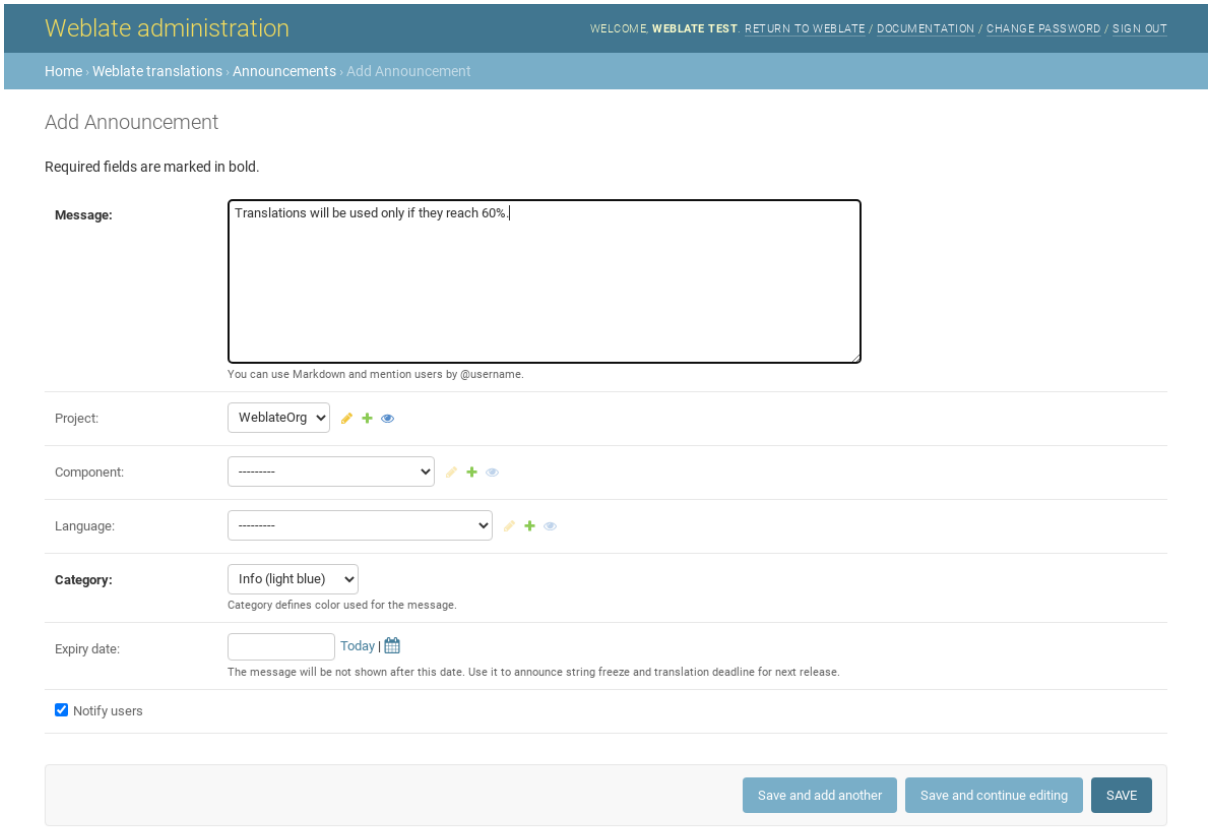
The users will receive notification on the announcements for watched projects (unless they opt out).

This can be useful for various things from announcing the purpose of the website to specifying targets for translations.

The announcements can posted on each level in the *Manage* menu, using *Post announcement*:



It can be also added using the admin interface:



The announcements are then shown based on their specified context:

Nie określono kontekstu

Wyświetlany na kokpicie (strona docelowa).

Określony projekt

Shown within the project, including all its components and translations.

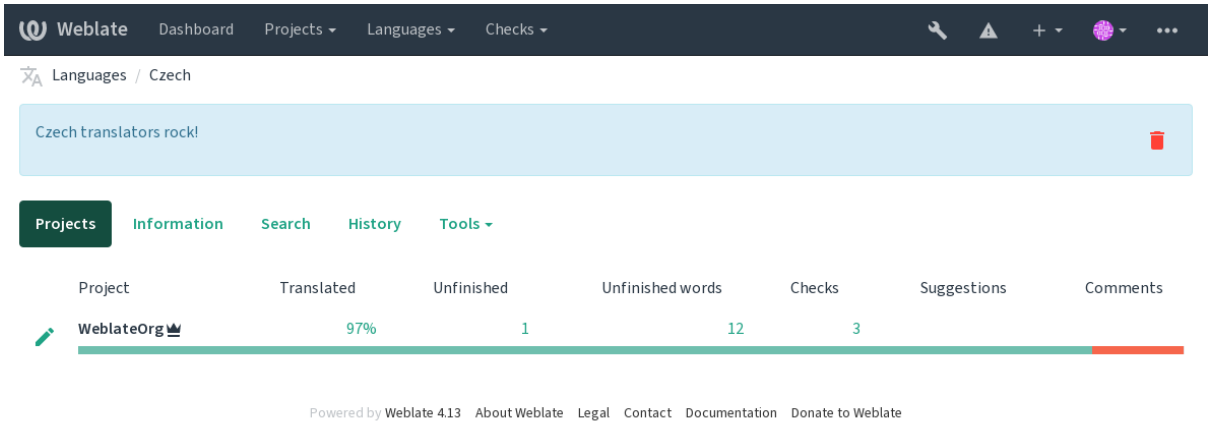
Określony komponent

Shown for a given component and all its translations.

Określony język

Shown on the language overview and all translations in that language.

This is how it looks on the language overview page:



2.20 Listy komponentów

Specify multiple lists of components to appear as options on the user dashboard, from which users can pick one as their default view. See *Kokpit* to learn more.

Zmienione w wersji 2.20: A status will be presented for each component list presented on the dashboard.

The names and content of component lists can be specified in the admin interface, in *Component lists* section. Each component list must have a name that is displayed to the user, and a slug representing it in the URL.

Zmienione w wersji 2.13: Change dashboard settings for anonymous users from the admin interface, altering what dashboard is presented to unauthenticated users.

2.20.1 Automatyczne listy komponentów

Nowe w wersji 2.13.

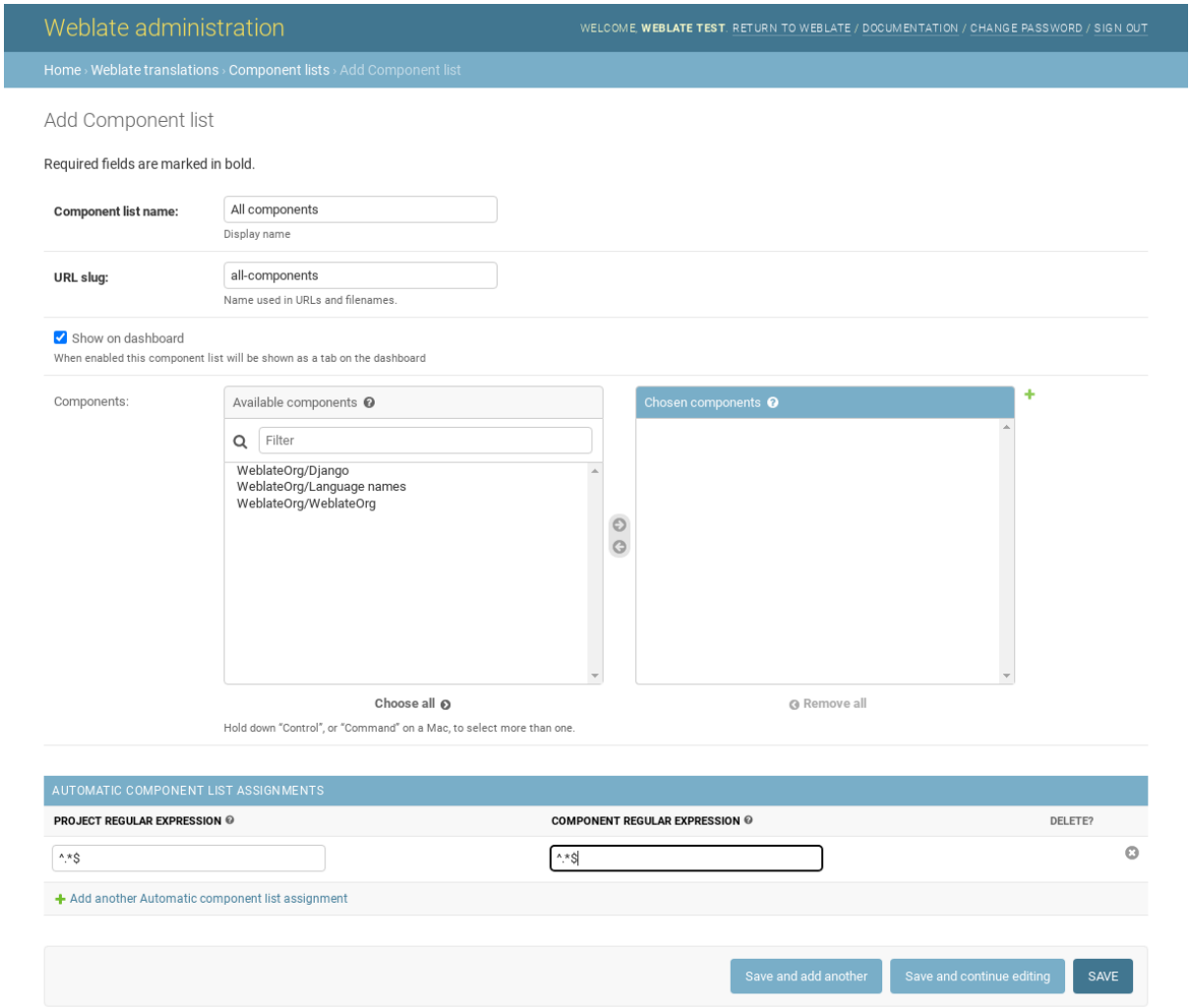
Add components to the list automatically based on their slug by creating *Automatic component list assignment* rules.

- Useful for maintaining component lists for large installations, or in case you want to have one component list with all components on your Weblate installation.

Podpowiedź: Make a component list containing all the components of your Weblate installation.

---

1. Define *Automatic component list assignment* with `^.*$` as regular expression in both the project and the component fields, as shown on this image:



2.21 Opcjonalne moduły Weblate

Several optional modules are available for your setup.

2.21.1 Eksporter Git

Nowe w wersji 2.10.

Provides you read-only access to the underlying Git repository using HTTP(S).

Instalacja

1. Add `weblate.gitexport` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.gitexport",)
```

2. Export existing repositories by migrating your database after installation:

```
weblate migrate
```

Zastosowanie

The module automatically hooks into Weblate and sets the exported repository URL in the *Konfiguracja komponentu*. The repositories are accessible under the `/git/` part of the Weblate URL, for example `https://example.org/git/weblate/main/`.

Repositories for publicly available projects can be cloned without authentication:

```
git clone 'https://example.org/git/weblate/main/'
```

Access to browse the repositories with restricted access (with *Private access control* or when `REQUIRE_LOGIN` is enabled) requires an API token which can be obtained in your *user profile*:

```
git clone 'https://user:KEY@example.org/git/weblate/main/'
```

Podpowiedź: By default members or *Users* group and anonymous user have access to the repositories for public projects via *Access repository* and *Power user* roles.

2.21.2 Rozliczenia

Nowe w wersji 2.4.

This is used on Hosted Weblate to define billing plans, track invoices and usage limits.

Instalacja

1. Add `weblate.billing` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.billing",)
```

2. Run the database migration to optionally install additional database structures for the module:

```
weblate migrate
```

Zastosowanie

After installation you can control billing in the admin interface. Users with billing enabled will get new *Billing* tab in their *Profil użytkownika*.

The billing module additionally allows project admins to create new projects and components without being superusers (see *Dodawanie projektów i komponentów tłumaczeniowych*). This is possible when following conditions are met:

- The billing is in its configured limits (any overusage results in blocking of project/component creation) and paid (if its price is non zero)
- The user is admin of existing project with billing or user is owner of billing (the latter is necessary when creating new billing for users to be able to import new projects).

Upon project creation user is able to choose which billing should be charged for the project in case he has access to more of them.

2.21.3 Informacje prawne

Nowe w wersji 2.15.

This is used on Hosted Weblate to provide required legal documents. It comes provided with blank documents, and you are expected to fill out the following templates in the documents:

legal/documents/tos.html
Dokument Warunków świadczenia usług

legal/documents/privacy.html
Dokument dotyczący polityki prywatności

legal/documents/summary.html
Krótkie podsumowanie warunków świadczenia usług i polityki prywatności

---

Informacja: Legal documents for the Hosted Weblate service are available in this Git repository <https://github.com/WeblateOrg/wllegal/tree/main/wllegal/templates/legal/documents>.

Most likely these will not be directly usable to you, but might come in handy as a starting point if adjusted to meet your needs.

Instalacja

1. Add `weblate.legal` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.legal",)

# Optional:

# Social auth pipeline to confirm TOS upon registration/subsequent sign in
SOCIAL_AUTH_PIPELINE += ("weblate.legal.pipeline.tos_confirm",)

# Middleware to enforce TOS confirmation of signed in users
MIDDLEWARE += [
    "weblate.legal.middleware.RequireTOSMiddleware",
]
```

2. Run the database migration to optionally install additional database structures for the module:

```
weblate migrate
```

3. Edit the legal documents in the `weblate/legal/templates/legal/` folder to match your service.

Zastosowanie

After installation and editing, the legal documents are shown in the Weblate UI.

2.21.4 Awatary

Avatars are downloaded and cached server-side to reduce information leaks to the sites serving them by default. The built-in support for fetching avatars from e-mails addresses configured for it can be turned off using `ENABLE_AVATARS`.

Weblate obecnie obsługuje:

- Gravatar
- Libravatar

Zobacz także:

Pamięć podręczna awatarów, `AVATAR_URL_PREFIX`, `ENABLE_AVATARS`

2.21.5 Ochrona przed spamem

You can protect against spamming by users by using the Akismet service.

1. Install the *akismet* Python module (this is already included in the official Docker image).
2. Uzyskaj klucz API Akismet.
3. Store it as `AKISMET_API_KEY` or `WEBLATE_AKISMET_API_KEY` in Docker.

Following content is sent to Akismet for checking:

- Sugestie od nieuwierzytelnionych użytkowników
- Opisy projektów i komponentów oraz linki

Informacja: This (among other things) relies on IP address of the client, please see *Działanie za odwrotnym proxy* for properly configuring that.

Zobacz także:

Działanie za odwrotnym proxy, `AKISMET_API_KEY`, `WEBLATE_AKISMET_API_KEY`

2.21.6 Signing Git commits with GnuPG

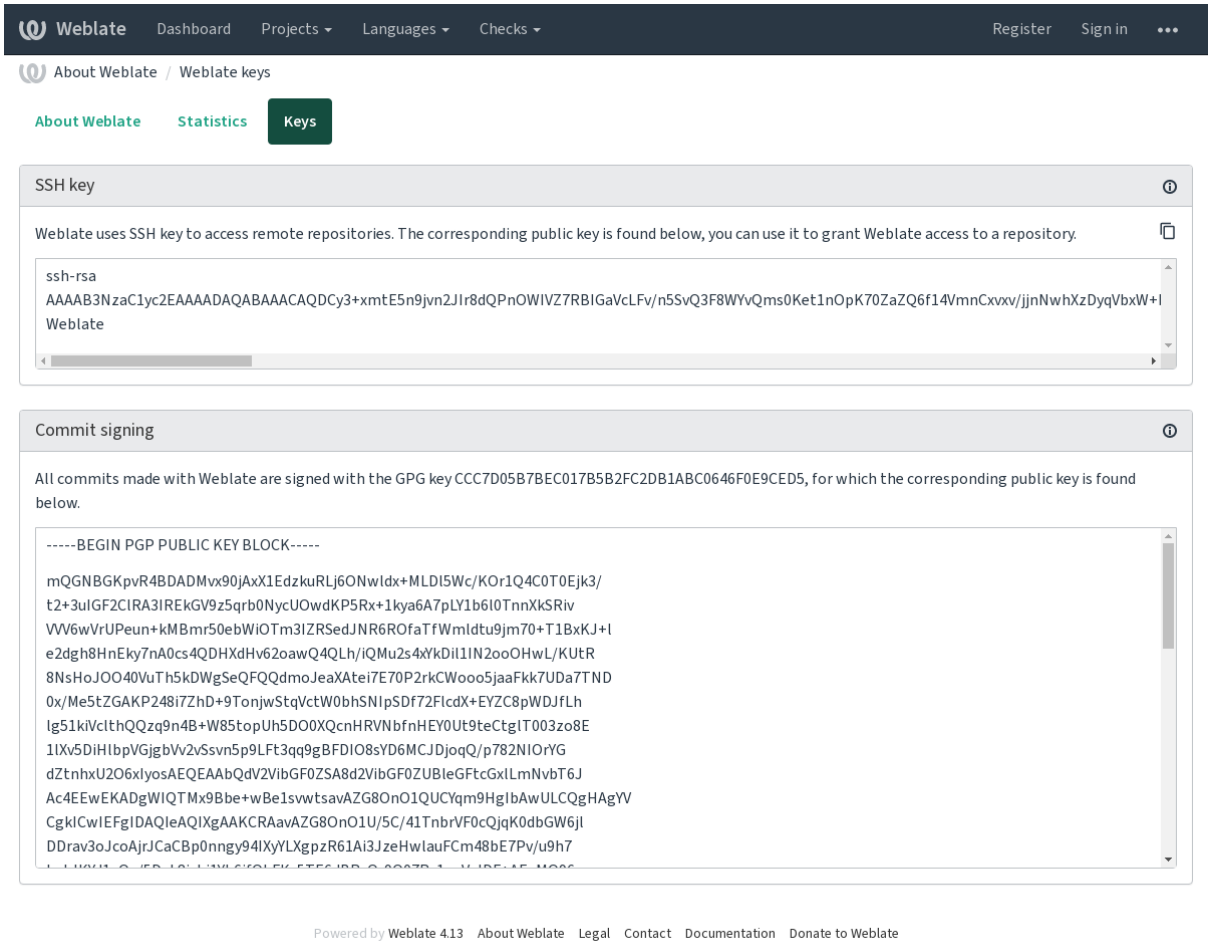
Nowe w wersji 3.1.

All commits can be signed by the GnuPG key of the Weblate instance.

1. Turn on `WEBLATE_GPG_IDENTITY`. (Weblate will generate a GnuPG key when needed and will use it to sign all translation commits.)

This feature needs GnuPG 2.1 or newer installed.

You can find the key in the `DATA_DIR` and the public key is shown on the „About” page:



2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking gpg.

Zobacz także:

WEBLATE_GPG_IDENTITY

2.21.7 Ograniczenie szybkości

Zmienione w wersji 3.2: The rate limiting now accepts more fine-grained configuration.

Zmienione w wersji 4.6: The rate limiting no longer applies to superusers.

Several operations in Weblate are rate limited. At most *RATELIMIT_ATTEMPTS* attempts are allowed within *RATELIMIT_WINDOW* seconds. The user is then blocked for *RATELIMIT_LOCKOUT*. There are also settings specific to scopes, for example `RATELIMIT_CONTACT_ATTEMPTS` or `RATELIMIT_TRANSLATE_ATTEMPTS`. The table below is a full list of available scopes.

The following operations are subject to rate limiting:

Nazwa	Zakres	Dozwolone próby	Okno ograniczeń szybkości	Okres blokady
Rejestracja	`REGISTRATION`	5	300	600
Wysyłanie wiadomości do administratorów	`MESSAGE`	5	300	600
Password authentication on sign in	`LOGIN`	5	300	600
Wyszukiwanie po całej stronie	`SEARCH`	6	60	60
Tłumaczenie	`TRANSLATE`	30	60	600
Dodawanie do słownika	`GLOSSARY`	30	60	600
Rozpoczynanie tłumaczenia na nowy język	`LANGUAGE`	2	300	600
Tworzenie nowego projektu	`PROJECT`	5	600	600

If a user fails to sign in `AUTH_LOCK_ATTEMPTS` times, password authentication will be turned off on the account until having gone through the process of having its password reset.

The settings can be also applied in the Docker container by adding `WEBLATE_` prefix to the setting name, for example `RATELIMIT_ATTEMPTS` becomes `WEBLATE_RATELIMIT_ATTEMPTS`.

The API has separate rate limiting settings, see *Ograniczanie szybkości interfejsu API*.

Zobacz także:

Ograniczenie szybkości, *Działanie za odwrotnym proxy*, *Ograniczanie szybkości interfejsu API*

2.21.8 Integracja z Fedora Messaging

Fedora Messaging is AMQP-based publisher for all changes happening in Weblate. You can hook additional services on changes happening in Weblate using this.

The Fedora Messaging integration is available as a separate Python module `weblate-fedora-messaging`. Please see <https://github.com/WeblateOrg/fedora_messaging/> for setup instructions.

2.22 Dostosowywanie Weblate

Extend and customize using Django and Python. Contribute your changes upstream so that everybody can benefit. This reduces your maintenance costs; code in Weblate is taken care of when changing internal interfaces or refactoring the code.

Ostrzeżenie: Neither internal interfaces nor templates are considered a stable API. Please review your own customizations for every upgrade, the interfaces or their semantics might change without notice.

Zobacz także:

Przyczyń się do rozwoju Weblate

2.22.1 Tworzenie modułu Pythona

If you are not familiar with Python, you might want to look into Python For Beginners, explaining the basics and pointing to further tutorials.

To write some custom Python code (called a module), a place to store it is needed, either in the system path (usually something like `/usr/lib/python3.7/site-packages/`) or in the Weblate directory, which is also added to the interpreter search path.

Better yet, turn your customization into a proper Python package:

1. Create a folder for your package (we will use *weblate_customization*).
2. Within it, create a `setup.py` file to describe the package:

```
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. Create a folder for the Python module (also called `weblate_customization`) for the customization code.
4. Within it, create a `__init__.py` file to make sure Python can import the module.
5. This package can now be installed using **`pip install -e`**. More info to be found in Editable installs.
6. Once installed, the module can be used in the Weblate configuration (for example `weblate_customization.checks.FooCheck`).

Your module structure should look like this:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

You can find an example of customizing Weblate at <https://github.com/WeblateOrg/customize-example>, it covers all the topics described below.

2.22.2 Zmiana logo

1. Create a simple Django app containing the static files you want to overwrite (see *Tworzenie modułu Pythona*).

   Branding appears in the following files:

   `icons/weblate.svg`
   Logo shown in the navigation bar.

   `logo-*.png`
   Web icons depending on screen resolution and web-browser.

   `favicon.ico`
   Web icon used by legacy browsers.

   `weblate-*.png`
   Avatars for bots or anonymous users. Some web-browsers use these as shortcut icons.

email-logo.png
Używany w powiadomieniach e-mail.

2. Dodaj do `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    # Add your customization as first
    "weblate_customization",
    # Weblate apps are here…
)
```

3. Run `weblate collectstatic --noinput`, to collect static files served to clients.

Zobacz także:

Jak zarządzać plikami statycznymi (np. obrazy, JavaScript, CSS), *Obsługa plików statycznych*

2.22.3 Custom quality checks, add-ons and auto-fixes

To install your code for *Niestandardowe automatyczne korekty*, *Pisanie własnych kontroli* or *Pisanie dodatku* in Weblate:

1. Place the files into your Python module containing the Weblate customization (see *Tworzenie modułu Pythona*).
2. Add its fully-qualified path to the Python class in the dedicated settings (*`WEBLATE_ADDONS`*, *`CHECK_LIST`* or *`AUTOFIX_LIST`*):

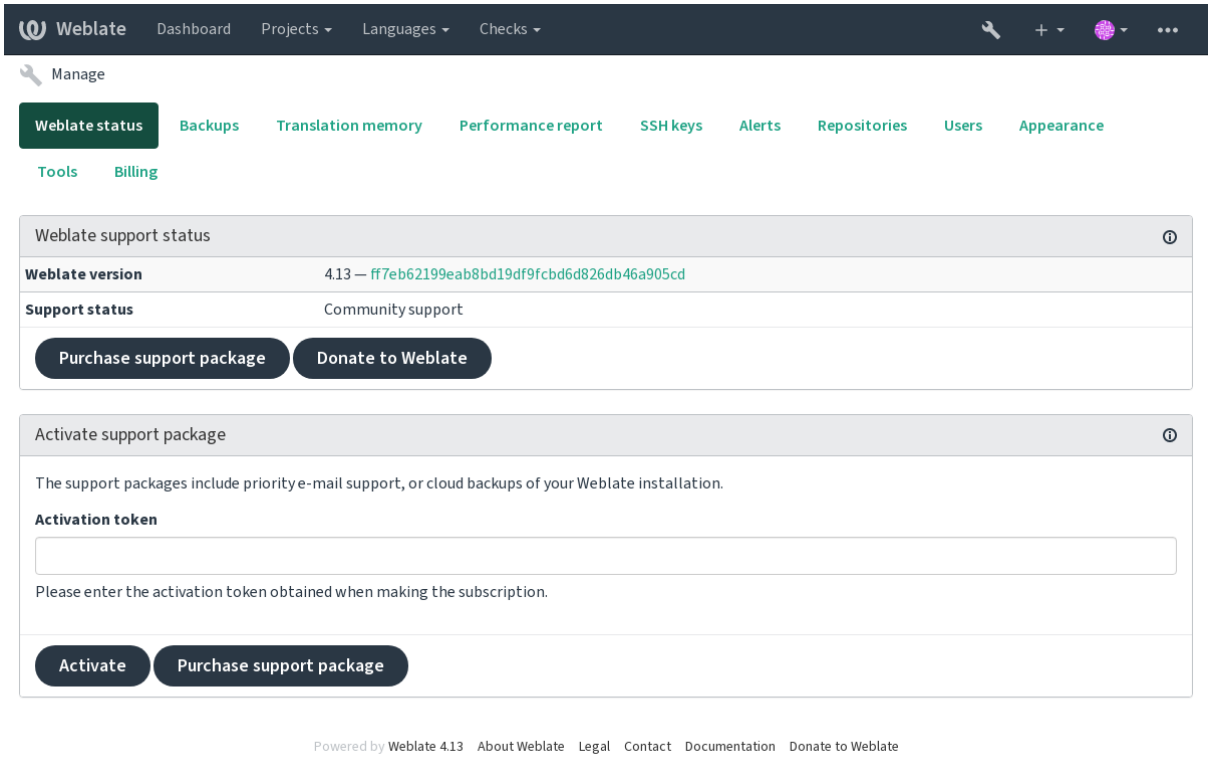
```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

# Add-ons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Zobacz także:

Niestandardowe automatyczne korekty, *Pisanie własnych kontroli*, *Pisanie dodatku*, *Wykonywanie skryptów z dodatku*

2.23 Interfejs zarządzania

The management interface offer administration settings under the `/manage/` URL. It is available for users signed in with admin privileges, accessible by using the wrench icon top right:



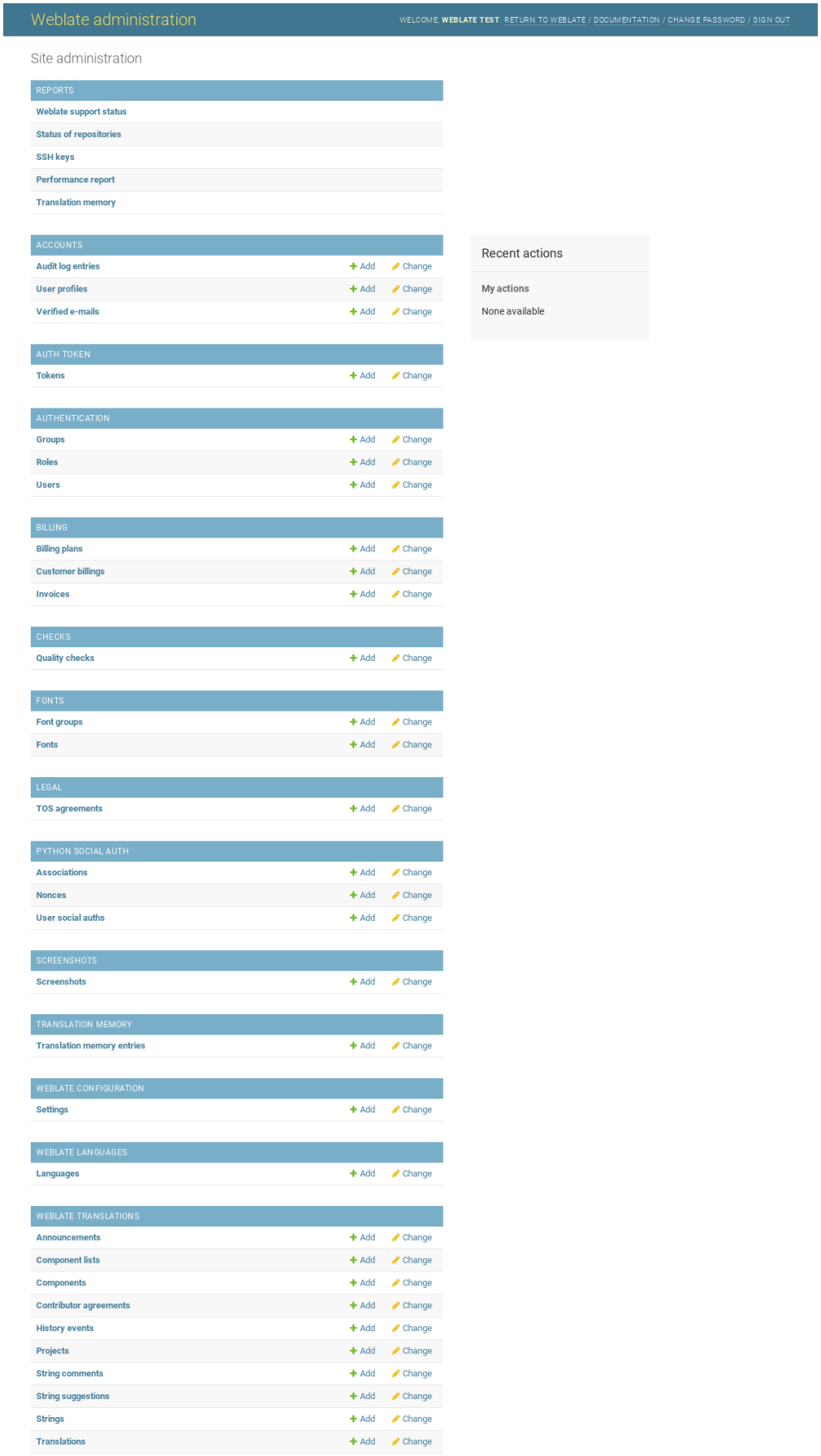
It includes basic overview of your Weblate:

- Support status, see *Uzyskaj wsparcie dla Weblate*
- Backups, see *Tworzenie kopii zapasowych i przenoszenie weblate*
- Współużytkowana pamięć tłumaczeniowa zobacz *Pamięć tłumaczeniowa*
- Performance report to review Weblate health and length of Celery queues
- Zarządzanie kluczami SSH zobacz *Repozytoria SSH*
- Alerts overview for all components, see alerts

2.23.1 Interfejs administracyjny Django

> **Ostrzeżenie:** Will be removed in the future, as its use is discouraged—most features can be managed directly in Weblate.

Here you can manage objects stored in the database, such as users, translations and other settings:


Weblate administration
WELCOME, WEBLATE TEST. RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Site administration
REPORTS
Weblate support status
Status of repositories
SSH keys
Performance report
Translation memory
ACCOUNTS
Audit log entries + Add Change
User profiles + Add Change
Verified e-mails + Add Change
AUTH TOKEN
Tokens + Add Change
AUTHENTICATION
Groups + Add Change
Roles + Add Change
Users + Add Change
BILLING
Billing plans + Add Change
Customer billings + Add Change
Invoices + Add Change
CHECKS
Quality checks + Add Change
FONTS
Font groups + Add Change
Fonts + Add Change
LEGAL
TOS agreements + Add Change
PYTHON SOCIAL AUTH
Associations + Add Change
Nonces + Add Change
User social auths + Add Change
SCREENSHOTS
Screenshots + Add Change
TRANSLATION MEMORY
Translation memory entries + Add Change
WEBLATE CONFIGURATION
Settings + Add Change
WEBLATE LANGUAGES
Languages + Add Change
WEBLATE TRANSLATIONS
Announcements + Add Change
Component lists + Add Change
Components + Add Change
Contributor agreements + Add Change
History events + Add Change
Projects + Add Change
String comments + Add Change
String suggestions + Add Change
Strings + Add Change
Translations + Add Change
Recent actions
My actions
None available

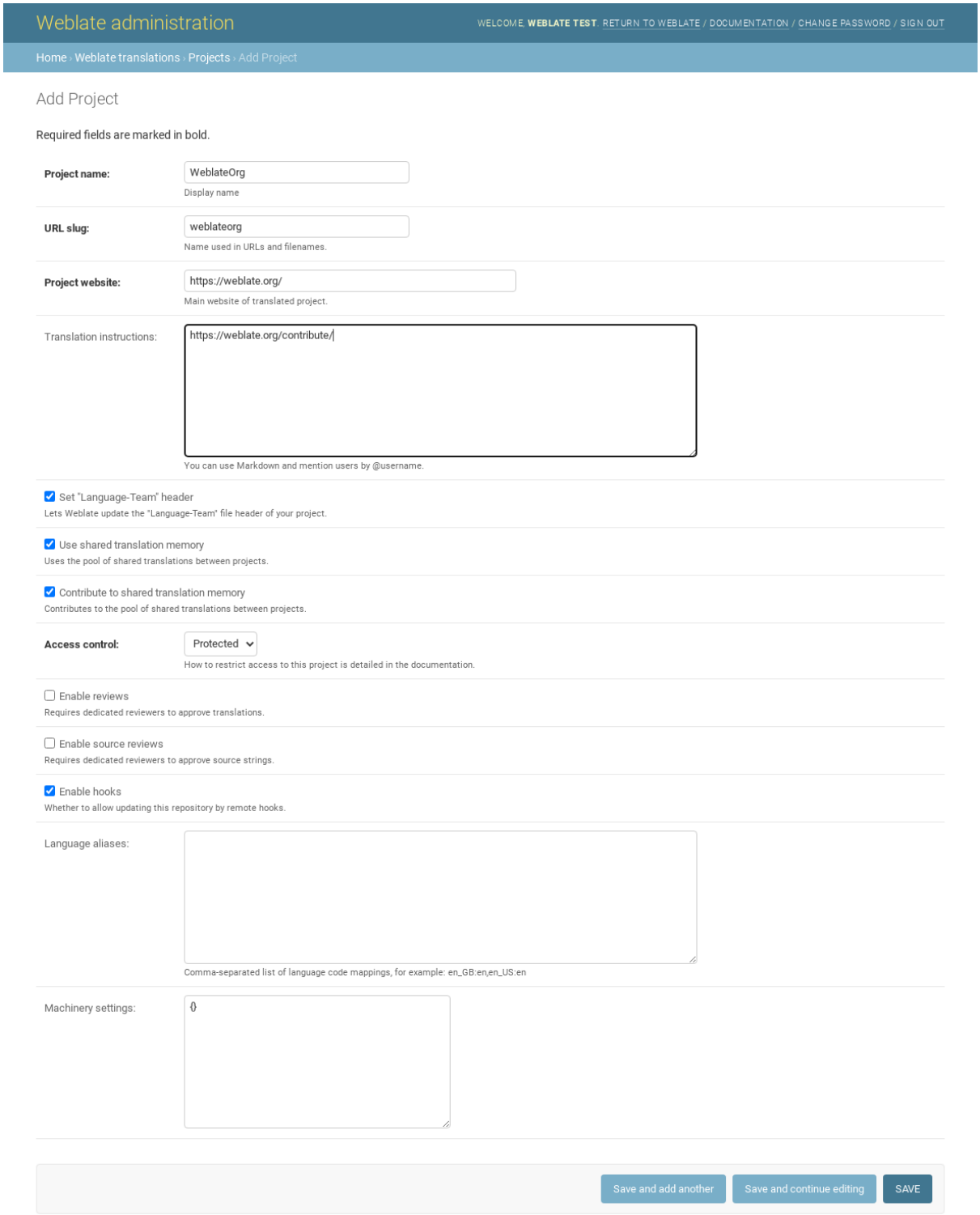
In the *Reports* section, you can check the status of your site, tweak it for *Instalacja produkcyjna*, or manage SSH keys used to access *Dostęp do repozytoriów*.

Manage database objects under any of the sections. The most interesting one is probably *Weblate translations*, where you can manage translatable projects, see *Konfiguracja projektu* and *Konfiguracja komponentu*.

Weblate languages holds language definitions, explained further in *Definicje języków*.

Dodawanie projektu

Adding a project serves as container for all components. Usually you create one project for one piece of software, or book (See *Konfiguracja projektu* for info on individual parameters):



Zobacz także:

Konfiguracja projektu

Komponenty dwujęzyczne

Once you have added a project, translation components can be added to it. (See *Konfiguracja komponentu* for info regarding individual parameters):

Weblate administration
Add Component
Required fields are marked in bold.
Component name:
Language names
URL slug:
language-names
Project:
WeblateOrg
Version control system:
Git
Source code repository:
https://github.com/WeblateOrg/demo.git
Repository push URL:
Repository browser:
Exported repository URL:
Source string bug reporting address:
Repository branch:
Push branch:
File mask:
weblate/langdata/locale/*/LC_MESSAGES/djs
Monolingual base language file:
Edit base file
Intermediate language file:
Template for new translations:
weblate/langdata/locale/django.pot
File format:
gettext PO file
Locked
Allow translation propagation
Turn on suggestions
Suggestion voting
Autoaccept suggestions:
0
Translation flags:
Enforced checks:
Translation license:
GNU General Public License v3.0 or later
Contributor agreement:
Adding new translation:
Create new language file
Language code style:
Default based on the file format
Manage strings
Merge style:
Rebase
Commit message when translating:
Translated using Weblate {{ language_name }}
Commit message when adding translation:
Added translation using Weblate {{ language_name }}
Commit message when removing translation:
Deleted translation using Weblate {{ language_name }}
Commit message when merging translation:
Merge branch '{{ component_remote_branch }}' into Weblate.
Commit message when add-on makes a change:
Update translation files
Merge request message:
Translations update from {{ site_title }}
Push on commit
Age of changes to commit:
24
Lock on error
Source language:
English
Language filter:
^(zsh|hu|x)$
Variants regular expression:
Priority:
Medium
Restricted component
Share in projects:
WeblateOrg
Use as a glossary
Glossary color:
Silver
Remote revision:
Local revision:
Save and add another
Save and continue editing
SAVE

Zobacz także:

Konfiguracja komponentu, *Formaty dwujęzyczne i jednojęzyczne*

Komponenty jednojęzyczne

For easier translation of these, provide a template file containing the mapping of message IDs to its respective source language (usually English). (See *Konfiguracja komponentu* for info regarding individual parameters):

Weblate administration
WELCOME WEBLATE TEST RETURN TO WEBLATE · DOCUMENTATION · CHANGE PASSWORD · SIGN OUT
Home · Weblate translations · Components · Add Component
Add Component
IMPORT SPIKE DOCUMENTATION
Required fields are marked in bold.
Component name:
Android
Display name
URL slug:
android
Name used in URLs and filenames.
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:
weblate://weblateorg/language-names
URL of a repository use weblate:{project}/component to share it with other component.
Repository push URL:
URL of a push repository, pushing is turned off if empty.
Repository browser:
Link to repository browser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders. You might want to strip leading directory by using {{filename|parentdir}}.
Exported repository URL:
URL of repository where users can fetch changes from Weblate.
Source string bug reporting address:
Email address for reports on errors in source strings. Leave empty for no e-mails.
Repository branch:
Repository branch to translate.
Push branch:
Branch for pushing changes, leave empty to use repository branch.
File mask:
app/src/main/res/values-*/strings.xml
Path of files to translate relative to repository root, use * instead of language code, for example: po*.po or locale/*/LC_MESSAGES/django.po.
Monolingual base language file:
app/src/main/res/values/strings.xml
Filename of translation base file, containing all strings and their source; it is recommended for monolingual translation formats.
Edit base file
Whether users will be able to edit the base file for monolingual translations.
Intermediate language file:
Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.
Template for new translations:
Filename of file used for creating new translations. For gettext choose .pot file.
File format:
Android Sitting Resource
Locked
Locked 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
Users can only vote for suggestions and can't make direct translations.
Autoaccept suggestions:
0
Automatically accept suggestions with this number of votes, use 0 to turn it off.
Translation flags:
Additional comma-separated flags to influence Weblate behavior.
Enforced checks:
{}
List of checks which can not be ignored.
Translation license:
MIT License
Contributor agreement:
User agreement which needs to be approved before a user can translate this component.
Adding new translation:
Create new language file
How to handle requests for creating new translations.
Language code style:
Default based on the file format
Customize language code used to generate the filename for translations created by Weblate.
Manage strings
Enables adding and removing strings straight from Weblate. If your strings are extracted from the source code or managed externally you probably want to keep it disabled.
Merge style:
Rebase
Define whether Weblate should merge the upstream repository or rebase changes onto 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_remote_branch }}' into Weblate.
You can use template language for various info, please consult the documentation for more details.
Commit message when add-on makes a change:
Update translation files
Updated by "{{ addon_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.
Merge request message:
Translations update from {{ site_title }}
Translations update from {{ site_title }}{{{ site_url }}} for {{ project_name }}/{{ component_name }}{{}}{{}}
{% if component_linked_child% %}
Rules includes following components:
{% for linked in component_linked_child% %}
* {{ linked.project_name }}/{{ linked.name }}{{ linked.url }}
{% endfor %}
{% endif %}
You can use template language for various info, please consult the documentation for more details.
Push on commit
Whether the repository should be pushed upstream on every commit.
Age of changes to commit:
24
Time in hours after which any pending changes will be committed to the VCS.
Lock on error
Whether the component should be locked on repository errors.
Source language:
English
Language used for source strings in all components.
Language filter:
^[^]+$
Regular expression used to filter translation files when scanning for file mask.
Variants 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.
Share in projects:
WeblateOrg
Choose additional projects where this component will be listed. Hold down "Control", or "Command" on a Mac, to select more than one.
Use as a glossary
Glossary color:
Silver
Remote revision:
Local revision:
Save and add another
Save and continue editing
SAVE

Zobacz także:

Konfiguracja komponentu, *Formaty dwujęzyczne i jednojęzyczne*

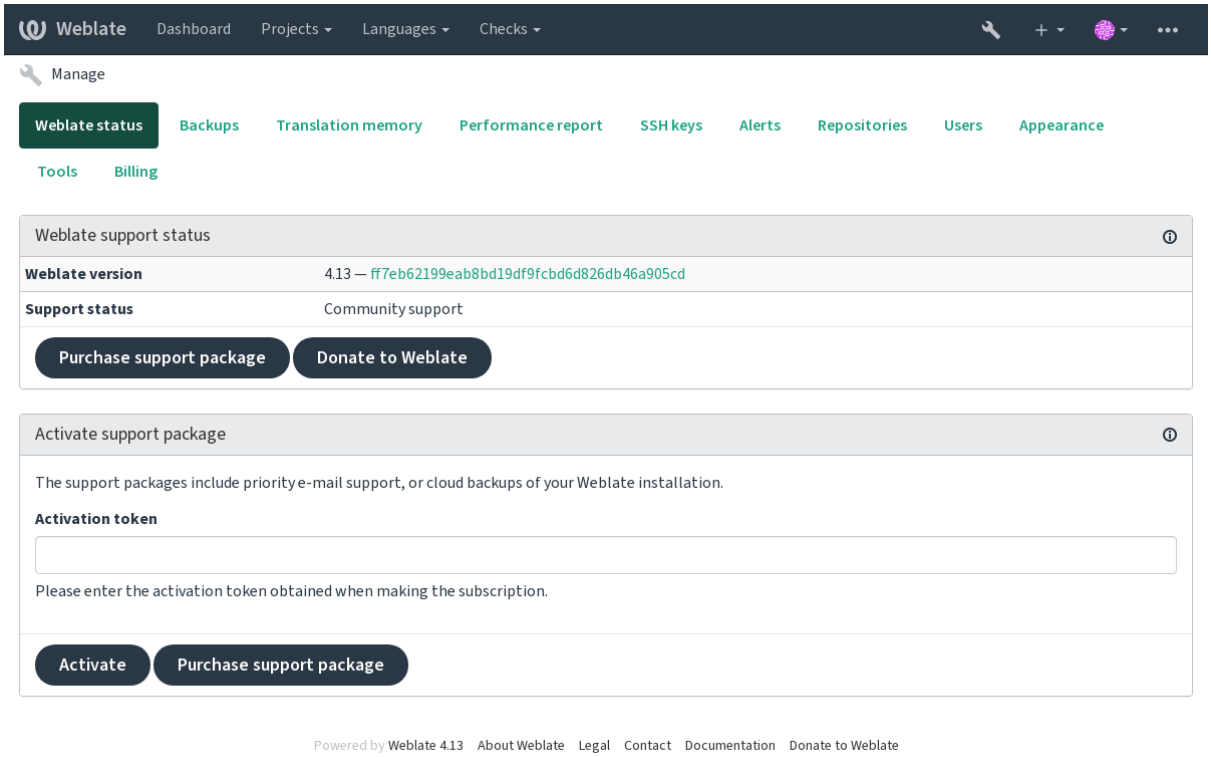
2.24 Uzyskaj wsparcie dla Weblate

Weblate is copylefted libre software with community support. Subscribers receive priority support at no extra charge. Prepaid help packages are available for everyone. You can find more info about current support offerings at <https://weblate.org/support/>.

2.24.1 Integracja wsparcia

Nowe w wersji 3.8.

Purchased support packages can optionally be integrated into your Weblate subscription management interface, from where you will find a link to it. Basic instance details about your installation are also reported back to Weblate this way.



2.24.2 Data przesłania do Weblate

- URL where your Weblate instance is configured
- Tytuł Twojej strony
- The Weblate version you are running
- Tallies of some objects in your Weblate database (projects, components, languages, source strings and users)
- The public SSH key of your instance

Additionally, when *Odkryj Weblate* is turned on:

- List of public projects (name, URL and website)

Żadne inne dane nie są przesłane.

2.24.3 Usługi integracyjne

- See if your support package is still valid
- *Weblate provisioned backup storage*
- *Odkryj Weblate*

Podpowiedź: Purchased support packages are already activated upon purchase, and can be used without integrating them.

2.24.4 Odkryj Weblate

Nowe w wersji 4.5.2.

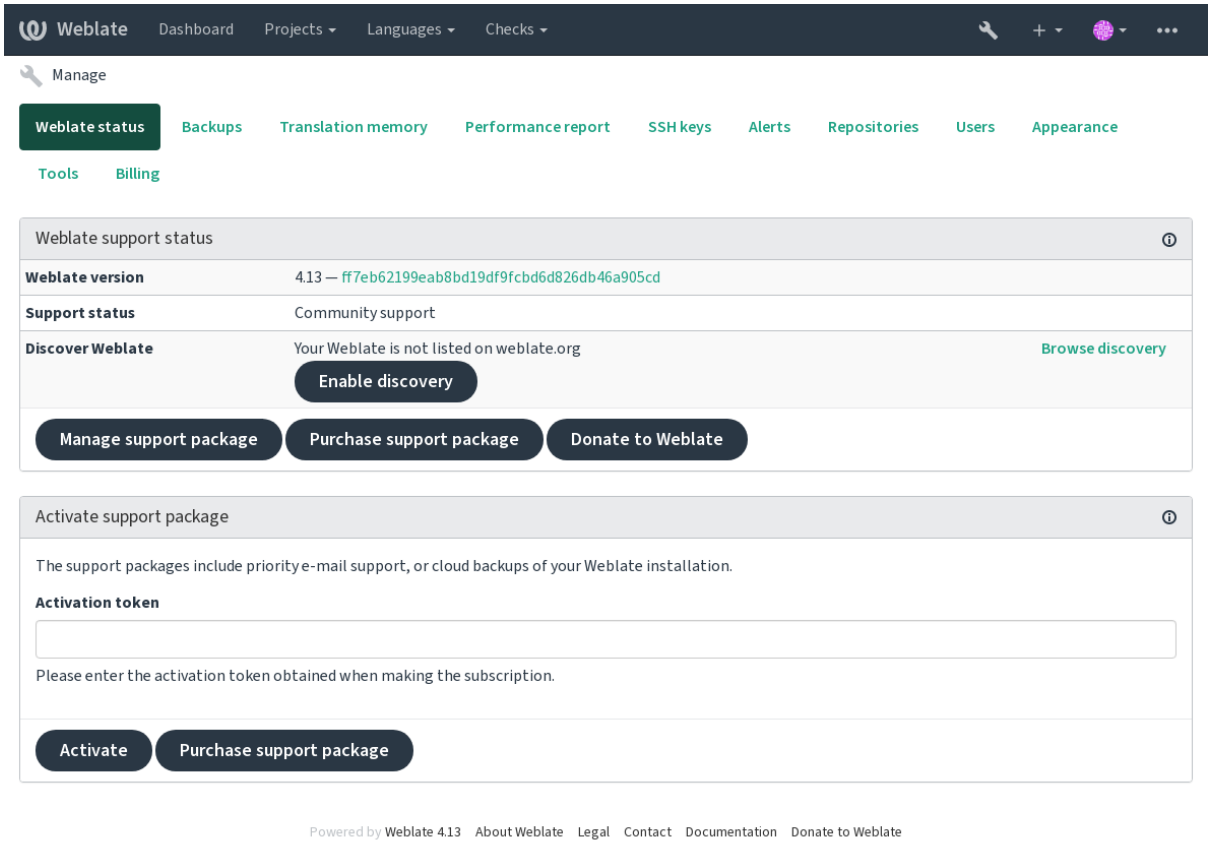
Informacja: Ta funkcja jest obecnie we wczesnej wersji beta.

Discover Weblate is an opt-in service that makes it easier for users to find Weblate servers and communities. Users can browse registered services on <https://weblate.org/discover/>, and find there projects to contribute.

Umieszczanie na liście

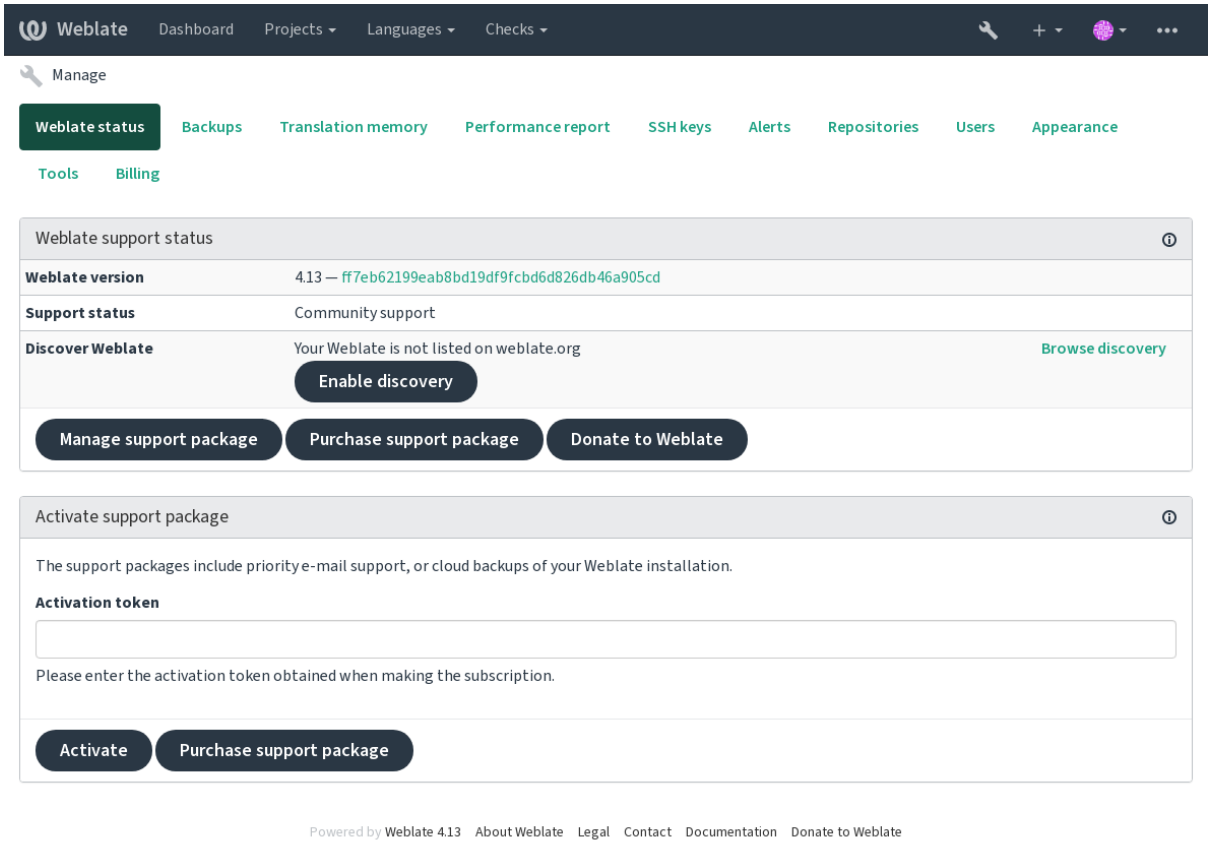
Podpowiedź: Participating in Discover Weblate makes Weblate submit some information about your server, please see *Data przesłania do Weblate*.

To list your server with an active support subscription (see *Integracja wsparcia*) in Discover Weblate all you need to do is turn this on in the management panel:



Dodawanie serwera bez abonamentu do listy Odkryj Weblate:

1. Zarejestruj się na <https://weblate.org/user/>
2. Zarejestruj swój serwer Weblate w bazie danych Odkrywania pod adresem <https://weblate.org/subscription/discovery/>
3. Potwierdź aktywację usługi w Weblate i włącz listę odkrywania na stronie zarządzania Weblate za pomocą przycisku *Włącz odkrywanie* :



Dostosowywanie wpisu

Możesz dostosować wpis, podając tekst i obraz (570 × 260 pikseli) pod adresem <https://weblate.org/user/>.

2.25 Dokumenty prawne

Informacja: 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 and other export controls

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 Kontrole szyfrowania w USA

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
: Zobacz https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

GnuPG
: Opcjonalnie używane przez Weblate

Git
: Opcjonalnie używane przez Weblate

curl
: Używany przez Git

OpenSSL
: Used by Python and cURL

The strength of encryption keys depends 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:

- Accessing remote servers using secure protocols (HTTPS)
- Generating signatures for code commits (PGP)

Zobacz także:

Export Controls (EAR) on Open Source Software

ROZDZIAŁ 3

Dokumenty współautora

3.1 Przyczyń się do rozwoju Weblate

There are dozens of ways to improve Weblate. You can choose the one you feel comfortable with, be it coding, graphics design, documentation, sponsorship, or an idea:

- *Zgłaszanie problemów w Weblate*
- *Rozpoczęcie wspierania rozwoju kodu Weblate*
- *Współtworzenie modułów Weblate*
- *Tłumaczenie Weblate*
- *Przyczyń się do rozwoju dokumentacji Weblate*
- *Dyskusje o Weblate*
- *Finansowanie rozwoju Weblate*

3.1.1 Tłumaczenie Weblate

Weblate jest stale **`translated<https://hosted.weblate.org/>`_** przy użyciu samego Weblate. Zapraszam do wzięcia udziału w staraniach o udostępnienie Weblate w jak największej liczbie języków. Przybliża to Weblate do swoich użytkowników!

If you find a possible mistake in the source string, you can mark it with a comment in the Weblate editor. This way, it can be discussed and corrected. If you're certain, you can also click on the link in the *Source string location* section and submit a PR with your correction.

3.1.2 Przyczyń się do rozwoju dokumentacji Weblate

You are welcome to improve the documentation page of your choice. Do it easily by clicking the *Edit on GitHub* button in the top-right corner of the page.

Please respect these guidelines while writing:

1. Don't remove part of the documentation if it's valid.
2. Use clear and easily-understandable language. You are writing tech docs, not a poem. Not all docs readers are native speakers, be thoughtful.
3. Don't be afraid to ask if you are not certain. If you have to ask about some feature while editing, don't change its docs before you have the answer. This means: You change or ask. Don't do both at the same time.
4. Verify your changes by performing described actions while following the docs.
5. Send PR with changes in small chunks to make it easier and quicker to review and merge.
6. If you want to rewrite and change the structure of a big article, do it in two steps:
    1. Rewrite
    2. Once the rewrite is reviewed, polished, and merged, change the structure of the paragraphs in another PR.

Podpowiedź: You can translate the docs.

3.1.3 Rozszerzanie wbudowanych definicji językowych

The language definitions are in the weblate-language-data repository.

You are welcome to add missing language definitions to `languages.csv`, other files are generated from that file.

3.1.4 Dyskusje o Weblate

If you have an idea and not sure if it's suitable for an issue, don't worry. You can join the community in GitHub discussions.

3.1.5 Finansowanie rozwoju Weblate

You can boost Weblate's development on the donate page. Funds collected there are used to enable gratis hosting for libre software projects and further development of Weblate. Please check the donate page for options, such as funding goals and the rewards you get as a proud funder.

Supporters who have funded Weblate

List of Weblate supporters:

- Yashiro Ccs
- Cheng-Chia Tseng
- Timon Reinhard
- Cassidy James
- Loic Dachary
- Marozed

- https://freedombox.org/
- GNU Solidario (GNU Health)
- BallotReady
- Richard Nespithal
- MyExpenses.Mobi

Do you want to be in the list? Please see options on the Donate to Weblate.

3.2 Rozpoczęcie wspierania rozwoju kodu Weblate

Understand the Weblate source code by going through *Kod źródłowy Weblate*, *Frontend Weblate* and *Wewnętrzna struktura Weblate*.

3.2.1 Rozpoczęcie pracy z kodem

Familiarize yourself with the Weblate codebase, by having a go at the bugs labelled good first issue.

3.2.2 Uruchamianie Weblate lokalnie

The most comfortable approach to get started with Weblate development is to follow *Instalacja ze źródeł*. It will get you a virtualenv with editable Weblate sources.

1. Sklonuj kod źródłowy Weblate:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Utwórz virtualenv:

```
virtualenv .venv
.venv/bin/activate
```

3. Install Weblate (for this you need some system dependencies, see *Instalacja ze źródeł*):

```
pip install -e .
```

3. Install all dependencies useful for development:

```
pip install -r requirements-dev.txt
```

4. Start a development server:

```
weblate runserver
```

5. Depending on your configuration, you might also want to start Celery workers:

```
./weblate/examples/celery start
```

6. To run a test (see *Testy lokalne* for more details):

```
. scripts/test-database
./manage.py test
```

Zobacz także:

Instalacja ze źródeł

3.2.3 Running Weblate locally in Docker

If you have Docker and docker-compose installed, you can spin up the development environment by simply running:

```
./rundev.sh
```

It will create a development Docker image and start it. Weblate is running on <http://127.0.0.1:8080/> and you can sign in as the user `admin` using `admin` as the password. The new installation is empty, so you might want to continue with *Dodawanie projektów i komponentów tłumaczeniowych*.

The `Dockerfile` and `docker-compose.yml` for this are located in the `dev-docker` directory.

The script also accepts some parameters, to execute tests, run it with the `test` parameter and then specify any `test` parameters, for example running only tests in the `weblate.machine` module:

```
./rundev.sh test --failfast weblate.machine
```

Informacja: Be careful that your Docker containers are up and running before running the tests. You can check that by running the `docker ps` command.

Wyświetlanie logów:

```
./rundev.sh logs
```

To stop the background containers, run:

```
./rundev.sh stop
```

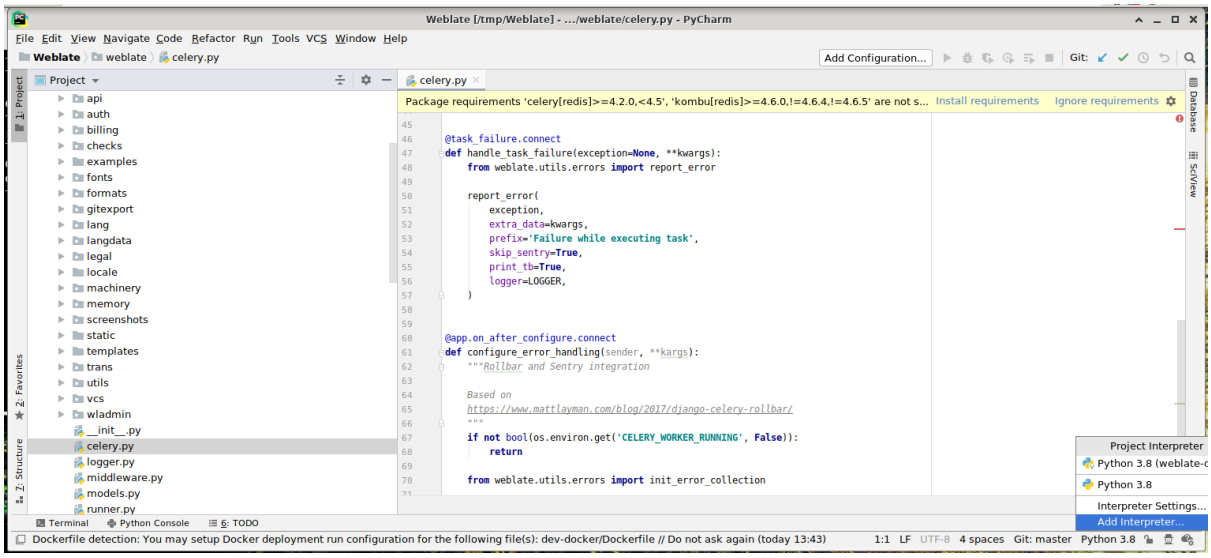
Running the script without arguments will re-create the Docker container and restart it.

Informacja: This is not a suitable setup for production, as it includes several hacks which are insecure, but they make development easier.

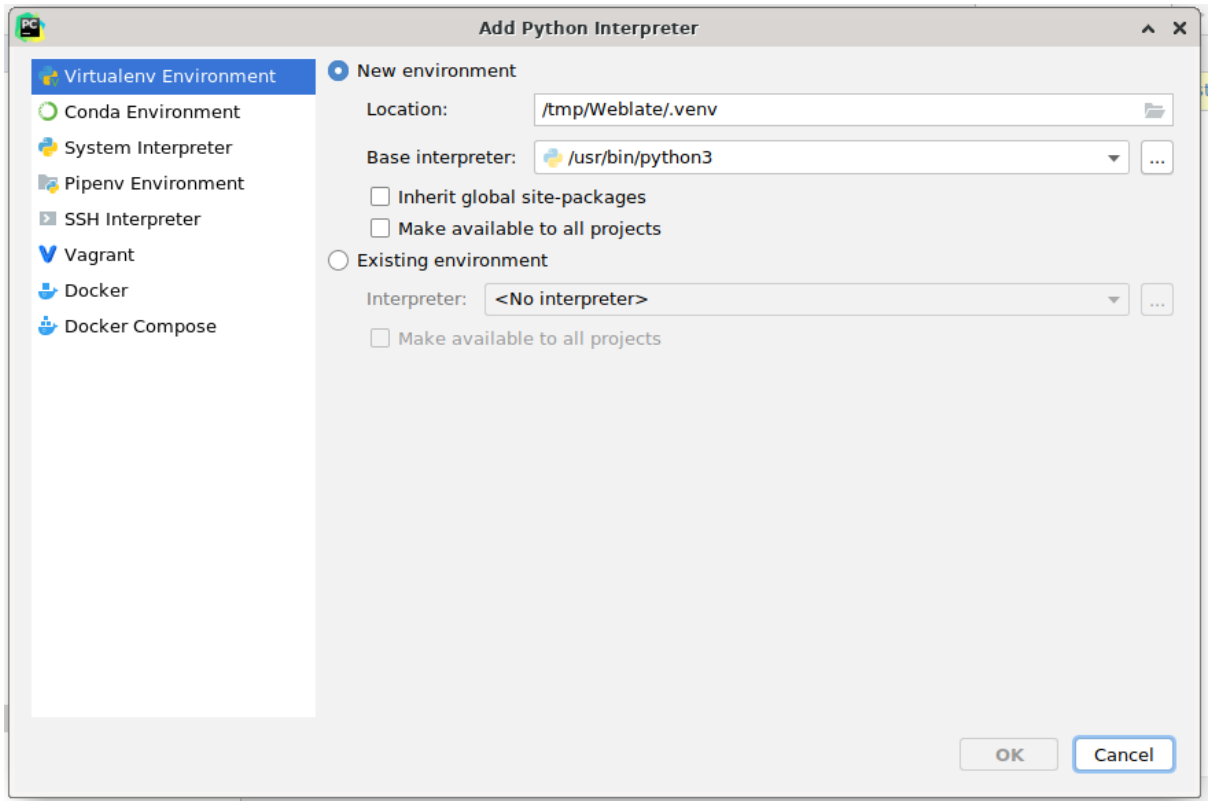
3.2.4 Kodowanie Weblate z PyCharm

PyCharm is a known IDE for Python, here are some guidelines to help you set up your Weblate project in it.

Considering you have just cloned the GitHub repository to a folder, just open it with PyCharm. Once the IDE is open, the first step is to specify the interpreter you want to use:

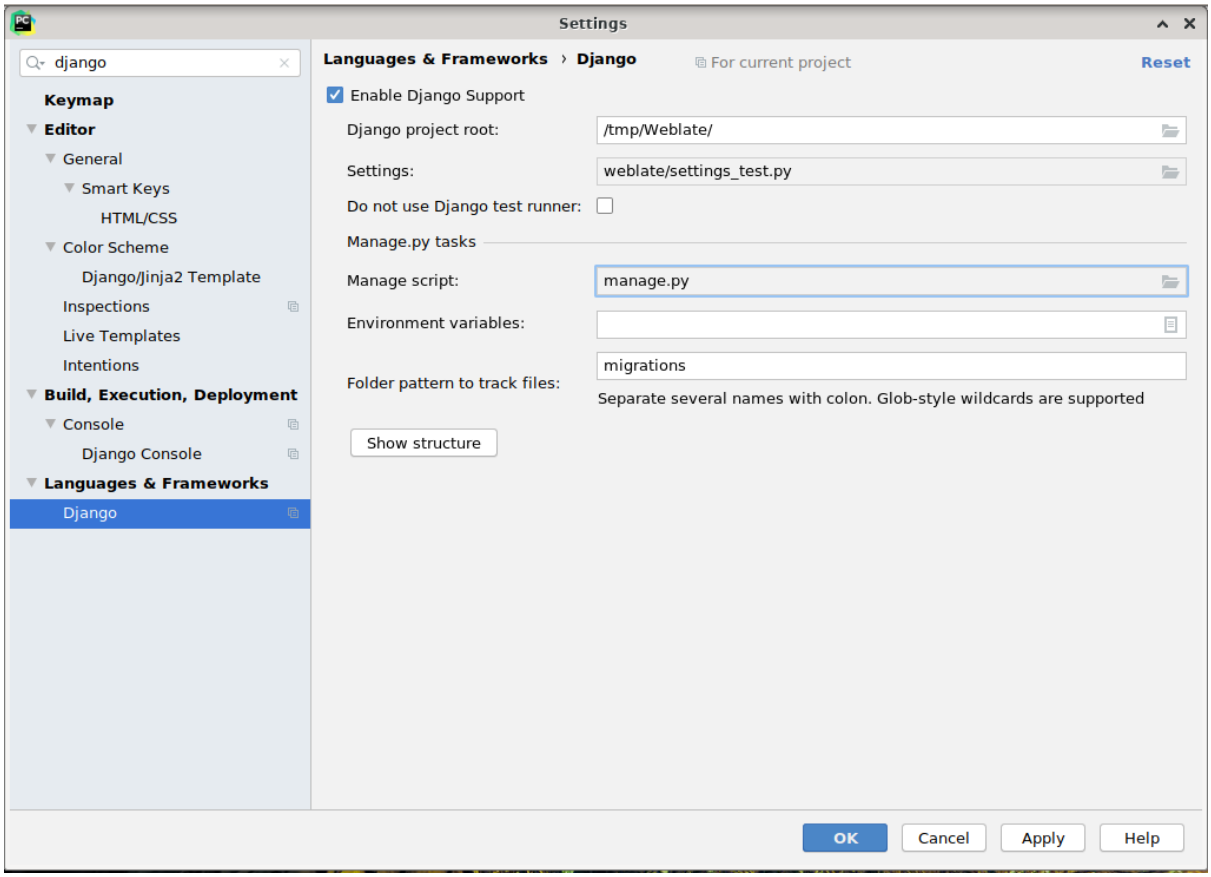


You can either choose to let PyCharm create the virtualenv for you, or select an already existing one:



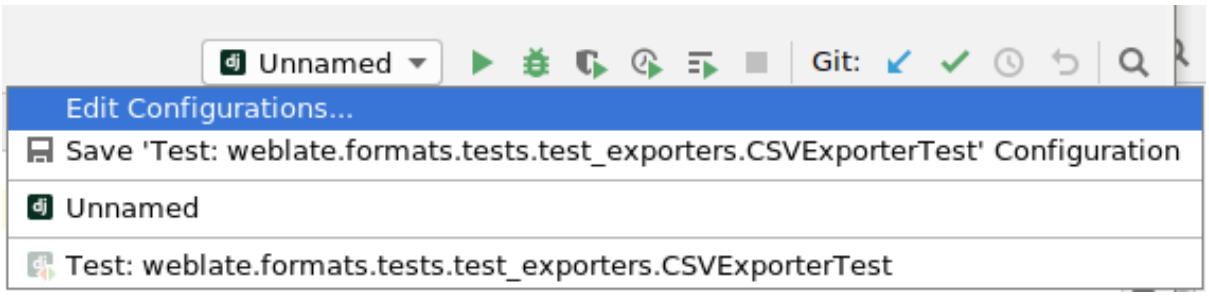
Don't forget to install the dependencies once the interpreter is set: Either through the console (the console from the IDE will directly use your virtualenv by default), or through the interface when you get a warning about missing dependencies.

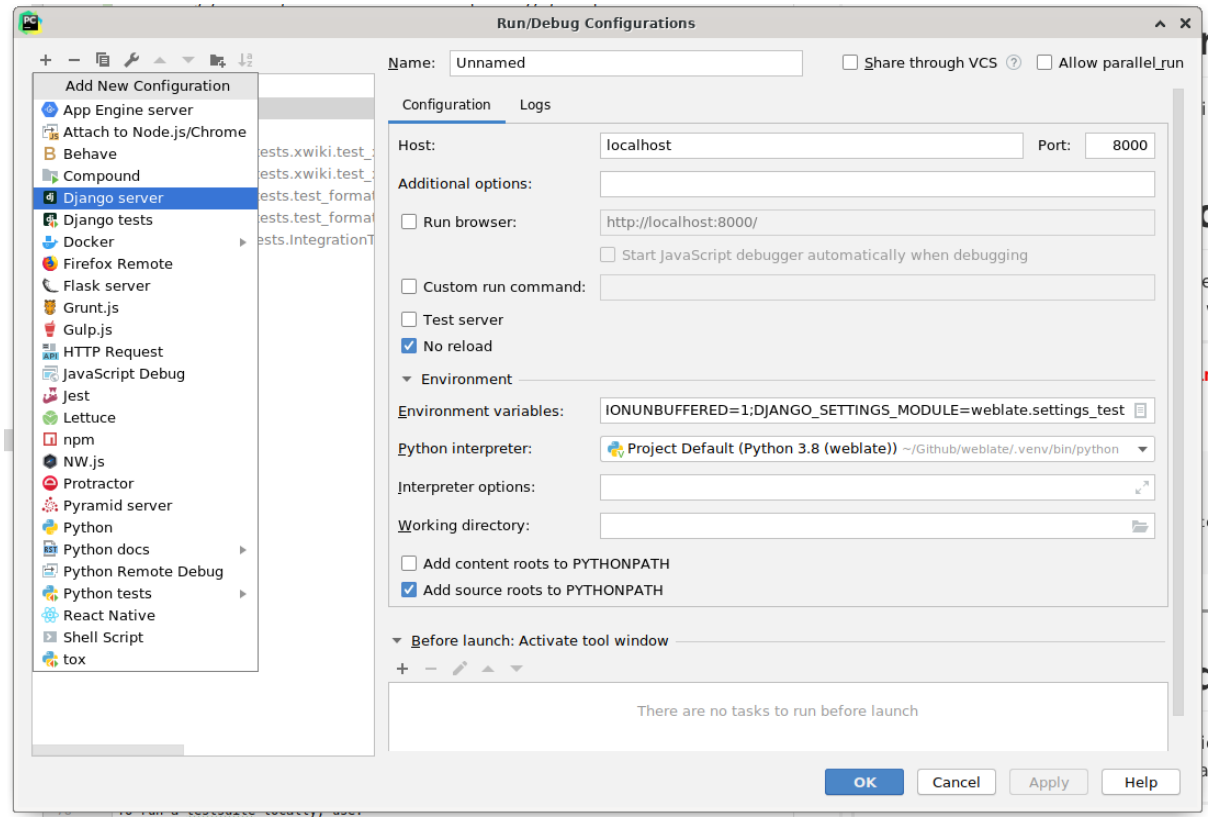
The second step is to set the right info to use Django natively inside PyCharm: The idea is to be able to immediately trigger the unit tests in the IDE. For that you need to specify the root path of the Django project and the path to its settings:



Be careful, the *Django project root* is the actual root of the repository, not the Weblate sub-directory. About the settings, you could use the `weblate/settings_test.py` from the repository, but you could create your own setting and set it there.

The last step is to run the server and to put breakpoints in the code to be able to debug it. This is done by creating a new *Django Server* configuration:





Podpowiedź: Be careful with the property called *No reload*: It prevents the server from being reloaded live if you modify files. This allows the existing debugger breakpoints to persist, when they normally would be discarded upon reloading the server.

3.2.5 Bootstrapping your devel instance

You might want to use `import_demo` to create demo translations and `createadmin` to make an admin user.

3.3 Kod źródłowy Weblate

Weblate is developed on GitHub. You are welcome to fork the code and open pull requests. Patches in any other form are welcome too.

Zobacz także:

Check out *Wewnętrzna struktura Weblate* to see how Weblate looks from inside.

3.3.1 Wytyczne dotyczące kodowania

Any code for Weblate should be written with Security by Design Principles in mind.

Any code should come with documentation explaining the behavior. Don't forget documenting methods, complex code blocks, or user visible features.

Any new code should utilize **PEP 484** type hints. We're not checking this in our CI yet as existing code does not yet include them.

3.3.2 Coding standard and linting the code

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 its configuration is placed in `setup.cfg`.

The easiest approach to enforce all this is to install pre-commit. The 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.

You can also trigger check manually, to check all files run:

```
pre-commit run --all
```

3.4 Debugowanie Weblate

Bugs can behave as application crashes or as various misbehavior. You are welcome to collect info on any such issue and submit it to the issue tracker.

3.4.1 Tryb diagnozowania błędów

Turning on debug mode will make the exceptions show in the web browser. This is useful to debug issues in the web interface, but not suitable for a production environment because it has performance consequences and might leak private data.

In a production environment, use `ADMINS` to receive e-mails containing error reports, or configure error collection using a third-party service.

Zobacz także:

Wyłączenie trybu debugowania, *Prawidłowe konfigurowanie administratorów*, *Zbieranie raportów o błędach*

3.4.2 Dzienniki logówweblate

Weblate can produce detailed logs of what is going on in the background. In the default configuration it uses syslog and that makes the log appear either in `/var/log/messages` or `/var/log/syslog` (depending on your syslog daemon configuration).

The Celery process (see *Zadania w tle korzystające z Celery*) usually produces its own logs as well. The example system-wide setups logs to several files under `/var/log/celery/`.

Docker containers log to their output (as per usual in the Docker world), so you can look at the logs using `docker-compose logs`.

Zobacz także:

Przykładowa konfiguracja zawiera `LOGGING` konfiguracja.

3.4.3 Not processing background tasks

A lot of things are done in the background by Celery workers. If things like sending out e-mails or component removal does not work, there might a related issue.

Rzeczy do sprawdzenia w tym przypadku:

- Sprawdź, czy proces Celery jest uruchomiony, zobacz *Zadania w tle korzystające z Celery*
- Check the Celery queue status, either in *Interfejs zarządzania*, or using `celery_queues`
- Look in the Celery logs for errors (see *Dzienniki logówweblate*)

3.4.4 Nie otrzymuję e-maili od Weblate

You can verify whether outgoing e-mail is working correctly by using the `sendtestemail` management command (see *Wywoływanie poleceń zarządzania* for instructions on how to invoke it in different environments) or by using *Interfejs zarządzania* under the *Tools* tab.

These send e-mails directly, so this verifies that your SMTP configuration is correct (see *Konfigurowanie poczty wychodzącej*). Most of the e-mails from Weblate are however sent in the background and there might be some issues with Celery involved as well, please see *Not processing background tasks* for debugging that.

3.4.5 Analiza awarii aplikacji

In case the application crashes, it is useful to collect as much info about the crash as possible. This can be achieved by using third-party services which can collect such info automatically. You can find info on how to set this up in *Zbieranie raportów o błędach*.

3.4.6 Ciche błędy

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 *Zbieranie raportów o błędach* helps you to notice such failures easier.

3.4.7 Problemy z wydajnością

In case Weblate performs badly in some scenario, 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 along with *Zbieranie raportów o błędach* and get pinpointed and detailed tracebacks in the error collection tool.

In case the slow performance is linked to the database, you can also enable logging of all database queries using following configuration after enabling `DEBUG`:

```
LOGGING["loggers"]["django.db.backends"] = {"handlers": ["console"], "level":
↪"DEBUG"}
```

3.5 Wewnętrzna struktura Weblate

Informacja: Ten rozdział zawiera podstawowe podsumowanie wewnętrznej struktury Weblate.

Weblate derives most of its code structure from, and is based on Django.

3.5.1 Struktura katalogów

Quick overview of directory structure of Weblate main repository:

docs
Source code for this documentation, which can be built using Sphinx.

dev-docker
Docker code to run development server, see *Running Weblate locally in Docker*.

weblate
Source code of Weblate as a Django application, see *Wewnętrzna struktura Weblate*.

weblate/static
Client files (CSS, Javascript and images), see *Frontend Weblate*.

3.5.2 Moduły

Weblate consists of several Django applications (some optional, see *Opcjonalne moduły Weblate*):

`accounts`

User account, profiles and notifications.

`addons`

Add-ons to tweak Weblate behavior, see *Dodatki*.

`api`

API based on Django REST framework.

`auth`

Uwierzytelnianie i uprawnienia.

`billing`

Opcjonalny moduł *Rozliczenia*.

`checks`

Translation string *Kontrole jakości* module.

`fonts`

Moduł kontroli renderowania czcionek.

`formats`

File format abstraction layer based on translate-toolkit.

`gitexport`

Opcjonalny moduł *Eksporter Git*.

`lang`

Module defining language and plural models.

`legal`

Opcjonalny moduł *Informacje prawne*.

`machinery`

Integration of machine translation services.

`memory`

Wbudowana pamięć tłumaczeń, zobacz *Pamięć tłumaczeniowa*.

`screenshots`

Screenshots management and OCR module.

`trans`

Główny moduł obsługujący tłumaczenia.

`utils`

Różne narzędzia pomocnicze.

`vcs`

Version control system abstraction.

`wladmin`

Dostosowywanie interfejsu administratora Django.

3.6 Tworzenie dodatków

Dodatki are way to customize localization workflow in Weblate.

class `weblate.addons.base.`**`BaseAddon`**(*storage=None*)

Base class for Weblate add-ons.

classmethod can_install(*component*, *user*)

Sprawdź, czy dodatek jest kompatybilny z danym komponentem.

configure(*settings*)

Zapisz konfigurację.

daily(*component*)

Hook wyzwalany codziennie.

classmethod get_add_form(*user*, *component*, ***kwargs*)

Zwróć formularz konfiguracji dla dodania nowego dodatku.

get_settings_form(*user*, ***kwargs*)

Zwróć formularz konfiguracji dla tego dodatku.

post_add(*translation*)

Hook wyzwalany po dodaniu nowego tłumaczenia.

post_commit(*component*)

Hook wyzwalany po scommitowaniu zmian do repozytorium.

post_push(*component*)

Hook wyzwalany po wysłaniu zmian na repozytorium.

post_update(*component*, *previous_head: str*, *skip_push: bool*)

Hook triggered after repository is updated from upstream.

Parametry

- **previous_head** (*str*) – HEAD of the repository prior to update, can be blank on initial clone.
- **skip_push** (*bool*) – Whether the add-on operation should skip pushing changes upstream. Usually you can pass this to underlying methods as `commit_and_push` or `commit_pending`.

pre_commit(*translation*, *author*)

Hook wyzwalany przed scommitowaniem zmian do repozytorium.

pre_push(*component*)

Hook triggered before repository is pushed upstream.

pre_update(*component*)

Hook triggered before repository is updated from upstream.

save_state()

Zapisz informacje o stanie dodatku.

store_post_load(*translation*, *store*)

Hook wyzwalany po przeparsowaniu pliku.

It receives an instance of a file format class as a argument.

This is useful to modify file format class parameters, for example adjust how the file will be saved.

unit_pre_create(*unit*)

Hook triggered before new unit is created.

Oto przykładowy dodatek:

```
#
# Copyright © 2012-2022 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 add-on should receive
    events = (EVENT_PRE_COMMIT,)
    # Add-on unique identifier
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example add-on")
    # Detailed add-on description
    description = _("This add-on does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

3.7 Frontend Weblate

The frontend is currently built using Bootstrap, jQuery and few third party libraries.

3.7.1 Obsługiwane przeglądarki

Weblate supports the latest, stable releases of all major browsers and platforms.

Alternative browsers which use the latest version of WebKit, Blink, or Gecko, whether directly or via the platform's web view API, are not explicitly supported. However, Weblate should (in most cases) display and function correctly in these browsers as well.

Older browsers might work, but some features might be limited.

3.7.2 Zarządzanie zależnościami

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. The Weblate specific code should be placed directly in `weblate/static` or feature specific subdirectories (for example `weblate/static/editor`).

Adding new third-party library typically consists of:

```
# Add a yarn package
yarn --cwd scripts/yarn add PACKAGE
# Edit the script to copy package to the static folder
edit scripts/yarn-update
# Run the update script
./scripts/yarn-update
# Add files to git
git add .
```

3.7.3 Styl kodowania

Weblate relies on Prettier for the code formatting for both JavaScript and CSS files.

We also use ESLint to check the JavaScript code.

3.7.4 Lokalizacja

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 = ngettext('literal for the singular case',
        'literal for the plural case', object_count);

fmts = ngettext('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'
```

Zobacz także:

Translation topic in the Django documentation

3.7.5 Ikony

Weblate currently uses material design icons. In case you are looking for new symbol, check Material Design Icons or Material Design Resources.

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.

3.8 Zgłaszanie problemów w Weblate

Weblate issue tracker jest hostowany na GitHubie.

Feel welcome to report any issues you have, or suggest improvement for Weblate there. There are various templates prepared to comfortably guide you through the issue report.

If what you have found is a security issue in Weblate, please consult the *Problemy bezpieczeństwa* section below.

If you are not sure about your bug report or feature request, you can try *Dyskusje o Weblate*.

3.8.1 Problemy bezpieczeństwa

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. Once you submit it there, community has limited but enough time to solve the incident.

Alternatively, report to 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. You can choose to encrypt it using this PGP key *3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D*. You can also get the PGP key from Keybase.

Informacja: 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.

Oto niektóre z nich:

- Django
- Django REST framework

- Python Social Auth

3.9 Pakiet testowy Weblate i ciągła integracja

Testsuites exist for most of the current code, increase coverage by adding testcases for any new functionality, and verify that it works.

3.9.1 Ciągła integracja

Current test results can be found on GitHub Actions and coverage is reported on Codecov.

There are several jobs to verify different aspects:

- Testy jednostkowe
- Documentation build and external links
- Migration testing from all supported releases
- Linting kodu
- 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=${1:-postgresql}

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

The simple execution can look like:

```
. scripts/test-database
./ci/run-migrate
./ci/run-test
./ci/run-docs
```

3.9.2 Testy lokalne

To run a testsuite locally, use:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

Podpowiedź: 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 *Konfiguracja bazy danych dla Weblate* for setup instructions.

The `weblate/settings_test.py` is used in CI environment as well (see *Ciągła integracja*) 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=${1:-postgresql}

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

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
```

Podpowiedź: The tests can also be executed inside developer docker container, see *Running Weblate locally in Docker*.

Zobacz także:

See Testing in Django for more info on running and writing tests for Django.

3.10 Schematy danych

Weblate uses JSON Schema to define layout of external JSON files.

3.10.1 Schemat pamięci tłumaczeniowej Weblate

https://weblate.org/schemas/weblate-memory.schema.json			
Typ	*tablica*		
menty	*The Translation Memory Item*		
	Typ	*obiekt*	
	właściwości		
	• **category**	*Kategoria ciągu*	
		1 is global, 2 is shared, 10000000+ are project specific, 20000000+ are user specific	
		Typ	*liczba całkowita*
		przykłady	1
		minimum	0
		omyślnie	1
	• **pochodzenie**	*Pochodzenie ciągu*	
		Nazwa pliku lub komponentu	
		Typ	*ciąg*
		przykłady	test.tmx
			projekt/komponent
		omyślnie	

ciąg dalszy na następnej stronie

Tabela 1 – kontynuacja poprzedniej strony

	• **źródło**	*Źródłowy ciąg znaków*	
		Typ	*ciąg*
		przykłady	Witaj
		minimalna długość	1
		omyślnie	
	• **źródło_język**	*Język źródłowy*	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		Typ	*ciąg*
		przykłady	en
		wzór	^[^]+$
		omyślnie	
	• **cel**	*Ciąg docelowy*	
		Typ	*ciąg*
		przykłady	Ahoj
		minimalna długość	1
		omyślnie	
	• **cel_język**	*Język docelowy*	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		Typ	*ciąg*
		przykłady	cs
		wzór	^[^]+$
		omyślnie	
	dodatkowe właściwości	Fałsz	
definicje			

Zobacz także:

Pamięć tłumaczeniowa, `dump_memory`, `import_memory`

3.10.2 Eksport danych użytkowników Weblate

https://weblate.org/schemas/weblate-userdata.schema.json			
Typ	*obiekt*		
właściwości			
• **podstawowy**	*Podstawowe*		
	Typ	*obiekt*	
	właściwości		
	• **nazwa użytkownika**	*Nazwa użytkownika*	
		Typ	*ciąg*
		przykłady	administrator
		omyślnie	
	• **imię_nazwisko**	*Imię i nazwisko*	
		Typ	*ciąg*
		przykłady	Administrator Weblate
		omyślnie	
	• **e-mail**	*Adres e-mail*	
		Typ	*ciąg*
		przykłady	noreply@example.com
		omyślnie	
	• **data_dołączenia**	*Data dołączenia*	
		Typ	*ciąg*
		przykłady	2019-11-18T18:53:54.862Z
		omyślnie	
• **profil**	*Profil*		
	Typ	*obiekt*	

ciąg dalszy na następnej stronie

Tabela 2 – kontynuacja poprzedniej strony

	właściwości			
	• **ęzyk**	**Język**		
		Typ	*ciąg*	
		przykłady	cs	
		wzór	^.*$	
		omyślnie		
	• **zaproponowane**	*Liczba zaproponowanych ciągów*		
		Typ	*liczba całkowita*	
		przykłady	1	
		omyślnie	0	
	• **przetłumaczone**	Liczba przetłumaczonych ciągów*		
		Typ	*liczba całkowita*	
		przykłady	24	
		omyślnie	0	
	• **przesłane**	*Liczba przesłanych zrzutów ekranu*		
		Typ	*liczba całkowita*	
		przykłady	1	
		omyślnie	0	
	• **hide_completed**	*Ukryj na kokpicie ukończone tłumaczenia*		
		Typ	*logiczne*	
		przykłady	Fałsz	
		omyślnie	rawda	
	• **secondary_in_zen**	*Pokaż dodatkowe tłumaczenia w trybie zen*		
		Typ	*logiczne*	
		przykłady	rawda	
		omyślnie	rawda	
	• **hide_source_secondary**	*Ukryj źródło, jeśli istnieje dodatkowe tłumaczenie*		
		Typ	*logiczne*	
		przykłady	Fałsz	
		omyślnie	rawda	
	• **odnośnik_edytora**	*Odnośnik edytora*		
		Typ	*ciąg*	
		przykłady		
		wzór	^.*$	
		omyślnie		
	• **tryb_tłumaczenia**	*Tryb edytora tłumaczeń*		
		Typ	*liczba całkowita*	
		przykłady	0	
		omyślnie	0	
	• **zen_mode**	*Tryb edytora Zen*		
		Typ	*liczba całkowita*	
		przykłady	0	
		omyślnie	0	
	• **znaki_specjalne**	*Znaki specjalne*		
		Typ	*ciąg*	
		przykłady		
		wzór	^.*$	
		omyślnie		
	• **domyślny_widok kokpitu**	*Domyślny widok kokpitu*		
		Typ	*liczba całkowita*	
		przykłady	1	
		omyślnie	0	
	• **dashboard_component_list**	*Domyślna lista komponentów*		
		omyślnie	null	
		którykolwiek z	Typ	*nieważne*
			Typ	*liczba całkowita*

ciąg dalszy na następnej stronie

Tabela 2 – kontynuacja poprzedniej strony

	• **ęzyki**	*Tłumaczone języki*		
		Typ	*tablica*	
		omyślnie		
		menty	*Kod języka*	
			Typ	*ciąg*
			przykłady	cs
			wzór	^.*$
			omyślnie	
	• **języ-ki_dodatkowe**	*Języki dodatkowe*		
		Typ	*tablica*	
		omyślnie		
		menty	*Kod języka*	
			Typ	*ciąg*
			przykłady	sk
			wzór	^.*$
			omyślnie	
	• **watched**	*Projekty obserwowane*		
		Typ	*tablica*	
		omyślnie		
		menty	*Końcówka adresu URL projektu*	
			Typ	*ciąg*
			przykłady	weblate
			wzór	^.*$
			omyślnie	
• **dziennik ak-tywności**	*Dziennik aktywności*			
	Typ	*tablica*		
	omyślnie			
	menty	*przedmioty*		
		Typ	*obiekt*	
		właściwości		
		• **adres**	*Adres IP*	
			Typ	*ciąg*
			przykłady	127.0.0.1
			wzór	^.*$
			omyślnie	
		• **agent_użytkownika**	*Klient użytkownika*	
			Typ	*ciąg*
			przykłady	PC / Linux / Firefox 70.0
			wzór	^.*$
			omyślnie	
		• **znacznik czasu**	**Znacznik czasu**	
			Typ	*ciąg*
			przykłady	2019-11-18T18:58:30.845Z
			wzór	^.*$
			omyślnie	
		• **aktywność**	*Aktywność*	
			Typ	*ciąg*
			przykłady	zaloguj się
			wzór	^.*$
			omyślnie	
definicje				

Zobacz także:

Profil użytkownika, `dumpuserdata`

3.11 Wydanie Weblate

3.11.1 Harmonogram wydawania

Weblate has two month release cycle for releases (x.y). These are usually followed by a bunch of bugfix releases to fix issues which slip into them (x.y.z).

The change in the major version indicates that the upgrade process can not skip this version - you always have to upgrade to x.0 before upgrading to higher x.y releases.

Zobacz także:

Aktualizacja Weblate

3.11.2 Planowanie wydania

The features for upcoming releases are collected using GitHub milestones, you can see our roadmap at <https://github.com/WeblateOrg/weblate/milestones>.

3.11.3 Proces uwalniania

Things to check prior to release:

1. Check newly translated languages by **`./scripts/list-translated-languages`**.
2. Set final version by **`./scripts/prepare-release`**.
3. Make sure screenshots are up to date **`make -j 12 -C docs update-screenshots`**.
4. Merge any possibly pending translations **`wlc push; git remote update; git merge origin/weblate`**

Wykonaj wydanie:

5. Create a release **`./scripts/create-release --tag`** (see below for requirements).

Ręczne kroki po wydaniu:

6. Aktualizowanie obrazu platformy Docker.
7. Zamknij kamień milowy GitHub.
8. Once the Docker image is tested, add a tag and push it.
9. Update Helm chart to new version.
10. Include new version in `.github/workflows/migrations.yml` to cover it in migration testing.
11. Increase version in the website download links.
12. Increase version in the repository by **`./scripts/set-version`**.

To create tags using the **`./scripts/create-release`** script you will need following:

- GnuPG with private key used to sign the release
- 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)

3.12 Bezpieczeństwo i prywatność

Wskazówka: W Weblate bezpieczeństwo utrzymuje środowisko, które ceni prywatność naszych użytkowników.

Rozwój Weblate przestrzega najlepszych praktyk Core Infrastructure Initiative od Linux Foundation.

Zobacz także:

Problemy bezpieczeństwa

3.12.1 Tracking dependencies for vulnerabilities

Problemy bezpieczeństwa w naszych zależnościach są monitorowane przy użyciu Dependabot. Sprawdzane są biblioteki Pythona i JavaScript, a najnowsze stabilne wydanie ma zaktualizowane zależności, aby uniknąć luk.

Podpowiedź: There might be vulnerabilities in third-party libraries which do not affect Weblate, so those are not addressed by releasing bugfix versions of Weblate.

3.12.2 Bezpieczeństwo kontenera Dockera

The Docker containers are regularly scanned using Anchore and Trivy security scanners.

Dzięki temu możemy wcześnie wykryć luki w zabezpieczeniach i szybko wprowadzić ulepszenia.

Wyniki skanowania można przeglądać na GitHubie - są one przechowywane jako artefakty na naszym CI w formacie SARIF (Static Analysis Results Interchange Format).

Zobacz także:

Ciągła integracja

3.13 Współtworzenie modułów Weblate

Besides the main repository, Weblate consists of several Python modules. All these follow same structure and this documentation covers them all.

Na przykład obejmuje to:

- wlc, Python client library, see *Klient Weblate*
- translation-finder, used to discover translatable files in the repository
- language-data, language definitions for Weblate, see *Definicje języków*

3.13.1 Wytyczne dotyczące kodowania

Any code for Weblate should be written with Security by Design Principles in mind.

Any code should come with documentation explaining the behavior. Don't forget documenting methods, complex code blocks, or user visible features.

Any new code should utilize **PEP 484** type hints. We're not checking this in our CI yet as existing code does not yet include them.

3.13.2 Przeprowadzanie testów

The tests are executed using **py.test**. First you need to install test requirements:

```
pip install -r requirements-test.txt
```

You can then execute the testsuite in the repository checkout:

```
py.test
```

Zobacz także:

The CI integration is very similar to *Pakiet testowy Weblate i ciągła integracja*.

3.13.3 Coding standard and linting the code

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 its configuration is placed in `setup.cfg`.

The easiest approach to enforce all this is to install pre-commit. The 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.

You can also trigger check manually, to check all files run:

```
pre-commit run --all
```

Zobacz także:

Kod źródłowy Weblate

3.14 O Weblate

3.14.1 Cele projektu

Web-based continuous localization tool with tight *Integracja kontroli wersji* supporting a wide range of *file formats*, making it easy for translators to contribute.

3.14.2 Nazwa projektu

„Weblate” is a portmanteau of the words „web” and „translate”.

3.14.3 Strona internetowa projektu

Strona docelowa to https://weblate.org, a usługa hostowana w chmurze znajduje się pod adresem https://hosted.weblate.org. Dokumentację można przeczytać na https://docs.weblate.org.

3.14.4 Logo projektu

The project logos and other graphics are available in https://github.com/WeblateOrg/graphics.

3.14.5 Kierownictwo

This project is maintained by Michal Čihař, who can be reached at michal@cihar.com.

3.14.6 Autorzy

Weblate was started by Michal Čihař. Since its inception in 2012, thousands of people have contributed.

3.15 Licencja

Copyright © 2012–2022 Michal Čihař michal@cihar.com

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/>.

ROZDZIAŁ 4

Historia zmian

4.1 Weblate 4.13

Released on June 15th 2022.

- Zmieniono zachowanie aktualizacji nazw języków.
- Dodano paginację do listy projektów.
- API for creating new units now returns information about newly created unit.
- Odnajdywanie komponentów obsługuje teraz konfigurowanie języka pośredniego.
- Added fixed encoding variants to CSV formats.
- Changed handling of context and location for some formats to better fit underlying implementation.
- Dodano obsługę formatu ResourceDictionary.
- Improved progress bar colors for color blind.
- Naprawiono czyszczenie wariantów podczas usuwania ciągów.
- Kompatybilność z Django 4.1.
- Dodano obsługę przechowywania zakodowanych elementów XML w XLIFF.
- Ulepszone formatowanie błędów sprawdzania symboli zastępczych.
- Redirect /.well-known/change-password to /accounts/password/.
- Usługi tłumaczenia maszynowego są teraz konfigurowalne dla każdego projektu.
- Added separate permission for resolving comments and grant it to the *Review strings* role.
- Added support for storing alternative translations in the CSV file.
- The placeholders check can now be case-insensitive as well.

Wszystkie zmiany w szczegółach.

4.2 Weblate 4.12.2

Wydano 11 maja 2022 r.

- Naprawiono przebudowę pamięci tłumaczeniowej projektu dla niektórych komponentów.
- Naprawiono sortowanie komponentów według nieprzetłumaczonych ciągów.
- Naprawiono możliwą utratę tłumaczeń podczas dodawania nowego języka.
- Ensure Weblate SSH key is generated during migrations.

Wszystkie zmiany w szczegółach.

4.3 Weblate 4.12.1

Wydano 29 kwietnia 2022 r.

- Naprawiono tytuł wiadomości żądania wciągnięcia.
- Ulepszona obsługa błędów składniowych w formacie Fluent.
- Naprawiono wyświetlanie awatara w wiadomościach e-mail z powiadomieniami.
- Dodane obsługę Web Monetization.
- Naprawiono usuwanie nieaktualnych ciągów źródłowych podczas usuwania tłumaczeń.

Wszystkie zmiany w szczegółach.

4.4 Weblate 4.12

Wydano 20 kwietnia 2022.

- Dodano obsługę języka amharskiego w *Niedopasowana kropka*.
- Dodano obsługę języka birmańskiego w *Niedopasowany znak zapytania*.
- Rozszerzono opcje dodatku *Generowanie pseudolokalizacji*.
- Added `ignore-all-checks` flag to ignore all quality checks on a string.
- Uniknięto wyzwalania nieudanych testów przez dodatek *Generowanie pseudolokalizacji*.
- Dodano wsparcie dla *Gitea pull requests*.
- Dodano kod języka w stylu Linuksa do *Styl kodu języka*.
- Dodano obsługę przebudowy pamięci tłumaczeniowej projektu.
- Ulepszone API do tworzenia komponentów z pliku.
- Dodano przyciski kopiowania i klonowania do innych tłumaczeń.
- Make merge request message configurable at component level.
- Improved maximal length restriction behavior with XML tags.
- Naprawiono ładowanie plików Fluent z dodatkowymi komentarzami.

Wszystkie zmiany w szczegółach.

4.5 Weblate 4.11.2

Wydano 4 marca 2022 r.

- Fixed corrupted MO files in the binary release.

Wszystkie zmiany w szczegółach.

4.6 Weblate 4.11.1

Wydano 4 marca 2022 r.

- Fixed missing sanitizing of arguments to Git and Mercurial - CVE-2022-23915, see GHSA-3872-f48p-pxqj for more details.
- Naprawiono ładowanie rozmytych ciągów z plików CSV.
- Dodano obsługę tworzenia zespołów za pomocą interfejsu API.
- Naprawiono wyświetlanie sugestii wspominania użytkowników.
- The project tokens access can now be customized.

Wszystkie zmiany w szczegółach.

4.7 Weblate 4.11

Wydano 25 lutego 2022 r.

- Fixes stored XSS - CVE-2022-24710, see GHSA-6jp6-9rf9-gc66 for more details.
- Naprawiono instalację dodatku przy użyciu API.
- Renamed *Strings needing action* to *Unfinished strings*.
- Naprawiono fałszywe alarmy z *Składnia ICU MessageFormat*.
- Indicate lock and contributor agreement on other occurrences listing.
- Fixed updating PO files with obsolete strings or missing plurals.
- Improved squash add-on compatibility with Gerrit.
- Automatically initialize user languages based on the Accept-Language header.
- Ulepszona obsługa błędów przy usuwaniu ciągów.
- Weblate now requires Python 3.7 or newer.
- Naprawiono niektóre operacje zapisu z uwierzytelnieniem tokenem projektu.
- Naprawiono śledzenie stanu ciągów, gdy ciągi zmieniają się w repozytorium.
- Śledzenie zmian ciągów z repozytorium.
- Sticky header on translations listing to improve navigation.
- Naprawiono nietłumaczalne ciągi w *Właściwości Java*.
- Fixed Git operation with non-ascii branch names.
- Nowy dodatek *Wypełnij wstępnie tłumaczenie źródłem*.
- Dodano :guilabel:` Scal bez przewijania do przodu` *Styl scalania*.
- Naprawiono *Tłumaczenie automatyczne* wyzwalacz dodatku dla nowo dodanych ciągów.
- Ulepszone sprawdzanie interpunkcji dla języka birmańskiego.

- Added support for defining custom teams at project level to grant users access, see *Zarządzanie kontrolą dostępu dla poszczególnych projektów*.
- Dodano do dokumentacji łącza do alertów.
- Docker container automatically enables TLS/SSL for outgoing e-mail when needed.
- Dodano obsługę wyszukiwania rozwiązanych komentarzy.
- Dodano wsparcie dla borgbackup 1.2.
- Fixed applying of *Automatically translated* label.

Wszystkie zmiany w szczegółach.

4.8 Weblate 4.10.1

Wydano 22 grudnia 2021 r.

- Documented changes introduced by upgrading to Django 4.0.
- Fixed displaying of *Automatically translated* label.
- Fixed API display of branch in components with a shared repository.
- Ulepszona analiza nieudanego alertu push.
- Fixed manually editing page when browsing changes.
- Poprawiona dokładność *Użyto litery Kashida*.
- The Weblate Docker container now uses Python 3.10.

Wszystkie zmiany w szczegółach.

4.9 Weblate 4.10

Wydano 16 grudnia 2021 r.

- Dodano obsługę wyboru formalności tłumaczenia i symboli zastępczych w DeepL.
- Bulk edit and search and replace are now available on project and language level.
- Dodano filtrowanie wyszukiwania i zastępowania.
- Fixed: „Perform automatic translation” privilege is no longer part of the *Languages* group.
- „Perform automatic translation” is in the *Administration* and the new *Automatic translation* group.
- Naprawiono generowanie plików XLSX ze znakami specjalnymi.
- Added ability to the GitHub authentication backend to check if the user belongs to a specific GitHub organization or team.
- Improved feedback on invalid parameters passed to API.
- Dodano obsługę tokenów dostępu do API w zakresie projektu.
- Naprawiono usuwanie ciągów w niektórych przypadkach.
- Naprawiono tłumaczenie nowo dodanych ciągów.
- Oznacz automatycznie przetłumaczone ciągi, aby ułatwić ich filtrowanie.

Wszystkie zmiany w szczegółach.

4.10 Weblate 4.9.1

Wydano 19 listopada 2021 r.

- Fixed upload of monolingual files after changing template.
- Ulepszona obsługa białych znaków we flagach.
- Dodaj obsługę filtrowania w API pobierania.
- Naprawiono wyświetlanie statystyk podczas dodawania nowych tłumaczeń.
- Mitigate issues with GitHub SSH key change.

Wszystkie zmiany w szczegółach.

4.11 Weblate 4.9

Wydano 10 listopada 2021 r.

- Provide more details for events in history.
- Ulepszone renderowanie historii.
- Poprawiono wydajność stron tłumaczeniowych.
- Dodano obsługę ograniczania pobierania plików z tłumaczeniami.
- The `safe-html` can now understand Markdown when used with `md-text`.
- The `max-length` tag now ignores XML markup when used with `xml-text`.
- Fixed dimensions of rendered texts in *Maksymalny rozmiar tłumaczenia*.
- Lowered app store title length to 30 to assist with upcoming Google policy changes.
- Dodano obsługę dostosowywania wywołania SSH przez `SSH_EXTRA_ARGS`.
- Dodano kontrole dla ICU MessageFormat.
- Ulepszona obsługa warunków błędów w backendach tłumaczeń maszynowych.
- Highlight unusual whitespace characters in the strings.
- Added option to stay on translated string while editing.
- Dodano obsługę dostosowywania wywołania Borg przez `BORG_EXTRA_ARGS`.
- Naprawiono generowanie plików MO dla tłumaczeń jednojęzycznych.
- Added API endpoint to download all component translations as a ZIP file.
- Dodano wsparcie dla Pythona 3.10.
- Dodano obsługę ponownego wysyłania zaproszeń e-mail z interfejsu zarządzania.

Wszystkie zmiany w szczegółach.

4.12 Weblate 4.8.1

Wydano 10 września 2021 r.

- Naprawiono usuwanie użytkowników w interfejsie administratora Django.
- Bardziej szczegółowe dokumentowanie parametrów dodatków.
- Naprawiono błąd JavaScript w słowniku.
- Dodano limit liczby dopasowań w kontroli spójności.
- Poprawiona obsługa symboli zastępczych w tłumaczeniach maszynowych.
- Naprawiono tworzenie dodatków za pomocą API.
- Added `PRIVACY_URL` setting to add privacy policy link to the footer.
- Hide member e-mail addresses from project admins.
- Improved gettext PO merging in case of conflicts.
- Ulepszone podświetlanie słownika.
- Improved `safe-html` flag behavior with XML checks.
- Naprawiono opisy commitów dla połączonych komponentów.

Wszystkie zmiany w szczegółach.

4.13 Weblate 4.8

Wydano 21 sierpnia 2021 r.

- Dodano wsparcie dla formatu Apple stringsdict.
- The exact search operator is now case-sensitive with PostgreSQL.
- Fixed saving glossary explanations in some cases.
- Poprawa dokumentacji.
- Poprawa wydajności.
- Improved squash add-on compatibility with Gerrit.
- Naprawiono dodawanie ciągów do komponentów słownika jednojęzycznego.
- Poprawiona wydajność obsługi wariantów.
- Fixed squash add-on sometimes skipping parsing upstream changes.
- Zachowaj rozszerzenia pobieranych plików.
- Dodano obsługę formatu Fluent.
- Dodano obsługę używania tabulatorów do wcięć w formacie JSON.

Wszystkie zmiany w szczegółach.

4.14 Weblate 4.7.2

Wydano 15 lipca 2021 r.

- Obsługa większej liczby aliasów językowych do skonfigurowania w projekcie.
- Naprawiono walidację wyszukiwanych ciągów w API.
- Naprawiono adresy URL eksporterów Git po zmianie domeny.
- Naprawiono dodatek do czyszczenia plików Windows RC.
- Naprawiono możliwą awarię podczas aktualizacji XLIFF.

Wszystkie zmiany w szczegółach.

4.15 Weblate 4.7.1

Wydano 30 czerwca 2021 r.

- Ulepszone wyskakujące okienko do dodawania terminów do słownika.
- Dodano obsługę usługi tłumaczenia maszynowego LibreTranslate.
- Dodano ograniczenie szybkości przy tworzeniu nowych projektów.
- Poprawiono wydajność aktualizacji plików.

Wszystkie zmiany w szczegółach.

4.16 Weblate 4.7

Wydano 17 czerwca 2021 r.

- Ulepszona kontrola kondycji konfiguracji.
- Dodano wsparcie dla `object-pascal-format` używanego w gettext PO, zobacz check-object-pascal-format`.
- Renamed *Nearby keys* to *Similar keys* to better describe the purpose.
- Dodano obsługę *Pliki lang dla mi18n*.
- Ulepszona integracja uwierzytelniania SAML.
- Fixed *Gerrit* integration to better handle corner cases.
- Weblate now requires Django 3.2.
- Naprawiono zapraszanie użytkowników, gdy uwierzytelnianie przez e-mail jest wyłączone.
- Ulepszone definicje języków.
- Dodano obsługę blokowania użytkowników przed współtworzeniem projektu.
- Naprawiono automatyczne tworzenie słowników języków.
- Rozszerzona dokumentacja dotycząca dodatków.
- Poprawa wydajności komponentów z powiązanymi repozytoriami.
- Dodano obsługę darmowego interfejsu API DeepL.
- Zarządzanie użytkownikami nie wymaga już interfejsu administratora Django.

Wszystkie zmiany w szczegółach.

4.17 Weblate 4.6.2

Wydano 8 maja 2021 r.

- Naprawiono awarię po przeniesieniu współdzielonego komponentu między projektami.
- Naprawiono dodawanie nowych ciągów do pustych plików właściwości.
- Naprawiono wyrównanie ikony kopiowania w językach RTL.
- Rozszerzone statystyki ciągów na karcie Informacje.
- Fixed handling of translation files ignored in Git.
- Poprawiona wydajność metryk.
- Naprawiono możliwy błąd w zapisywaniu słowników.
- Fixed consistency check behavior on languages with different plural rules.

Wszystkie zmiany w szczegółach.

4.18 Weblate 4.6.1

Wydano 2 maja 2021 r.

- Usunięto przestarzały kod ochrony przed spamem.
- Popraw dokładność kontroli liczby mnogiej źródła.
- Zaktualizowano listę języków interfejsu użytkownika w Dockerze.
- Ulepszono komunikaty o błędach podczas tworzenia pull requestów.
- Naprawiono tworzenie pull requestów na Pagure.
- Naprawiono automatyczne uruchamianie instalowanych dodatków.
- Naprawiono możliwe problemy z pamięcią podręczną podczas aktualizacji.
- Naprawiono dodawanie nowych jednostek do tłumaczeń jednojęzycznych za pomocą przesyłania.

Wszystkie zmiany w szczegółach.

4.19 Weblate 4.6

Wydano 19 kwietnia 2021 r.

- The auto_translate management command has now a parameter for specifying translation mode.
- Dodano wsparcie dla *Pliki tekstowe*.
- Dodano trendy i metryki dla wszystkich obiektów.
- Dodano obsługę bezpośredniego kopiowania tekstu z języków dodatkowych.
- Added date filtering when browsing changes.
- Ulepszone wykresy aktywności.
- Sender for contact form e-mails can now be configured.
- Ulepszona walidacja parametrów w API tworzenia komponentów.
- The rate limiting no longer applies to superusers.
- Ulepszona wydajność i niezawodność dodatku do automatycznego tłumaczenia.

- The rate limiting now can be customized in the Docker container.
- API for creating components now automatically uses *Wewnętrzne adresy URL Weblate*.
- Uproszczone wskazywanie stanu podczas przeglądania ciągów.
- Password hashing now uses Argon2 by default.
- Uproszczone paski postępu wskazujące stan tłumaczenia.
- Renamed *Dodaj brakujące języki* to clarify the purpose.
- Poprawiono zapisywanie stanu łańcuchów do XLIFF.
- Dodano wyszukiwanie w całym języku.
- Initial support for *Scaling horizontally* the Docker deployment.

Wszystkie zmiany w szczegółach.

4.20 Weblate 4.5.3

Wydano 1 kwietnia 2021 r.

- Naprawiono zbieranie metryk.
- Naprawiono możliwą awarię podczas dodawania ciągów.
- Ulepszone przykłady zapytań wyszukiwania.
- Fixed possible loss of newly added strings on replace upload.

4.21 Weblate 4.5.2

Wydano 26 marca 2021 r.

- Konfigurowalny harmonogram automatycznego tłumaczenia.
- Dodano kontrolę formatu Lua.
- Ignore format strings in the *Kolejne zduplikowane wyrazy* check.
- Allow uploading screenshot from a translate page.
- Added forced file synchronization to the repository maintenance.
- Fixed automatic suggestions for languages with a longer code.
- Poprawiono wydajność podczas dodawania nowych ciągów.
- Kilka poprawek błędów w kontrolach jakości.
- Kilka ulepszeń wydajności.
- Dodano integrację z *Odkryj Weblate*.
- Naprawiono zachowanie sprawdzania ciągów tylko do odczytu.

Wszystkie zmiany w szczegółach.

4.22 Weblate 4.5.1

Wydano 5 marca 2021 r.

- Fixed editing of glossary flags in some corner cases.
- Extend metrics usage to improve performance of several pages.
- Przechowuj poprawny język źródłowy w plikach TMX.
- Lepsza obsługa przesyłania jednojęzycznych plików PO za pomocą API.
- Ulepszone zachowanie alertów w komponentach słownika.
- Poprawiono kontrole łączy Markdown.
- Indicate glossary and source language in breadcrumbs.
- Podzielony na strony wykaz komponentów dużych projektów.
- Poprawiona wydajność tłumaczenia, usuwania komponentów lub projektów.
- Poprawiono wydajność edycji zbiorczej.
- Fixed preserving „Needs editing" and „Approved" states for ODF files.
- Ulepszony interfejs do dostosowywania pobierania plików tłumaczeń

Wszystkie zmiany w szczegółach.

4.23 Weblate 4.5

Wydano 19 lutego 2021 r.

- Dodano wsparcie dla `lua-format` używanego w gettext PO.
- Dodano obsługę udostępniania komponentu między projektami.
- Fixed multiple unnamed variables check behavior with multiple format flags.
- Dropped mailing list field on the project in favor of generic instructions for translators.
- Dodano dodatek generowania pseudolokalizacji.
- Dodano wsparcie dla plików TermBase eXchange.
- Dodano obsługę ręcznego definiowania wariantów ciągu za pomocą flagi.
- Poprawiona wydajność kontroli spójności.
- Poprawiona wydajność pamięci tłumaczeniowej dla długich ciągów.
- Dodano obsługę wyszukiwania w objaśnieniach.
- Strings can now be added and removed in bilingual formats as well.
- Extend list of supported languages in Amazon Translate machine translation.
- Automatycznie włączaj sprawdzanie formatu wiadomości Java dla właściwości Java.
- Dodano nową metodę przesyłania dla dodawania nowych ciągów do tłumaczenia.
- Dodano prosty interfejs do przeglądania tłumaczeń.
- Glossaries are now stored as regular components.
- Dropped specific API for glossaries as component API is used now.
- Added simplified interface to toggle some of the flags.
- Dodano obsługę nieprzetłumaczalnych lub zabronionych terminów w słowniku.

- Dodano obsługę definiowania terminologii w słowniku.
- Moved text direction toggle to get more space for the visual keyboard.
- Dodano opcję automatycznego obserwowania projektów, do których przyczynił się użytkownik.
- Added check whether translation matches the glossary.
- Dodano obsługę dostosowywania koloru tekstu nawigacji.

Wszystkie zmiany w szczegółach.

4.24 Weblate 4.4.2

Wydano 14 stycznia 2021 r.

- Fixed corruption of one distributed MO file.

4.25 Weblate 4.4.1

Wydano 13 stycznia 2021 r.

- Fixed reverting plural changes.
- Fixed displaying help for project settings.
- Ulepszona administracja użytkownikami.
- Ulepszono obsługa kontekstu w jednojęzycznych plikach PO.
- Fixed cleanup add-on behavior with HTML, ODF, IDML and Windows RC formats.
- Poprawione parsowanie „położenia” w plikach CSV.
- Use content compression for file downloads.
- Improved user experience on importing from ZIP file.
- Ulepszone wykrywanie formatu plików do przesyłania.
- Avoid duplicate pull requests on Pagure.
- Poprawiona wydajność podczas wyświetlania tłumaczenia-ducha.
- Reimplemented translation editor to use native browser textarea.
- Naprawiono dodatek do czyszczenia psujący dodawanie nowych ciągów.
- Dodano API dla dodatków.

Wszystkie zmiany w szczegółach.

4.26 Weblate 4.4

Wydano 15 grudnia 2020 r.

- Ulepszona walidacja podczas tworzenia komponentu.
- Weblate now requires Django 3.1.
- Dodano obsługę dostosowywania wyglądu w interfejsie zarządzania.
- Fixed read-only state handling in bulk edit.
- Ulepszona integracja z CodeMirror.

- Dodano dodatek do usuwania pustych ciągów z plików tłumaczeń.
- Edytor CodeMirror jest teraz używany do tłumaczeń.
- Syntax highlighting in translation editor for XML, HTML, Markdown and reStructuredText.
- Podświetl obiekty do umieszczania w edytorze tłumaczeń.
- Ulepszona obsługa niestandardowych kodów językowych.
- Dodano alert podczas używania niejednoznacznych kodów językowych.
- The user is now presented with a filtered list of languages when adding a new translation.
- Extended search capabilities for changes in history.
- Ulepszono strony szczegółów płatności i przepływ pracy z darmowym hostingiem.
- Rozszerzony interfejs API statystyk tłumaczenia.
- Ulepszona karta „inne tłumaczenia” podczas tłumaczenia.
- Dodano API zadań.
- Poprawiona wydajność przesyłania plików.
- Improved display of user defined special characters.
- Poprawiona wydajność automatycznego tłumaczenia.
- Several minor improvements in the user interface.
- Improved naming of ZIP downloads.
- Dodano opcję otrzymywania powiadomień o nieobserwowanych projektach.

  Wszystkie zmiany w szczegółach.

4.27 Weblate 4.3.2

Released on November 4th 2020.

- Naprawiono awarię niektórych masek plików komponentów.
- Poprawiono dokładność kontroli kolejnych zduplikowanych wyrazów.
- Dodano obsługę pull requestów na Pagure.
- Ulepszone komunikaty o błędach dla nieudanych rejestracji.
- Reverted rendering developer comments as Markdown.
- Simplified setup of Git repositories with different default branch than „master”.
- Newly created internal repositories now use main as the default branch.
- Reduced false positives rate of unchanged translation while translating reStructuredText.
- Fixed CodeMirror display issues in some situations.
- Renamed Template group to „Sources” to clarify its meaning.
- Fixed GitLab pull requests on repositories with longer paths.

Wszystkie zmiany w szczegółach.

4.28 Weblate 4.3.1

Released on October 21st 2020.

- Poprawiona wydajność automatycznego tłumaczenia.
- Fixed session expiry for authenticated users.
- Dodano obsługę ukrywania informacji o wersji.
- Improve hooks compatibility with Bitbucket Server.
- Poprawiona wydajność aktualizacji pamięci tłumaczeń.
- Zmniejszone zużycie pamięci.
- Poprawiona wydajność widoku matrycy.
- Dodano potwierdzenie przed usunięciem użytkownika z projektu.

  Wszystkie zmiany w szczegółach.

4.29 Weblate 4.3

Released on October 15th 2020.

- Include user stats in the API.
- Fixed component ordering on paginated pages.
- Define source language for a glossary.
- Rewritten support for GitHub and GitLab pull requests.
- Naprawiono liczenie statystyk po usunięciu sugestii.
- Rozszerzony publiczny profil użytkownika.
- Fixed configuration of enforced checks.
- Improve documentation about built-in backups.
- Moved source language attribute from project to a component.
- Dodano kontrolę formatowania Vue I18n.
- Generic placeholders check now supports regular expressions.
- Poprawiony wygląd trybu matrycy.
- Maszyny są teraz nazywane automatycznymi sugestiami.
- Added support for interacting with multiple GitLab or GitHub instances.
- Extended API to cover project updates, unit updates and removals and glossaries.
- Unit API now properly handles plural strings.
- Component creation can now handle ZIP file or document upload.
- Consolidated API response status codes.
- Obsługa Markdown w umowie z autorem.
- Ulepszone śledzenie ciągów źródłowych.
- Improved JSON, YAML and CSV formats compatibility.
- Dodano obsługę usuwania ciągów.
- Improved performance of file downloads.

- Improved repository management view.
- Automatically enable java-format for Android.
- Dodano obsługę przetłumaczonych zrzutów ekranu.
- Dodano wsparcie dla Pythona 3.9.
- Fixed translating HTML files under certain conditions.

Wszystkie zmiany w szczegółach.

4.30 Weblate 4.2.2

Released on September 2nd 2020.

- Poprawiono dopasowywanie ciągów źródłowych dla formatów JSON.
- Fixed login redirect for some authentication configurations.
- Fixed LDAP authentication with group sync.
- Naprawiono awarię podczas raportowania postępu tłumaczenia automatycznego.
- Fixed Git commit squashing with trailers enabled.
- Fixed creating local VCS components using API.

4.31 Weblate 4.2.1

Released on August 21st 2020.

- Fixed saving plurals for some locales in Android resources.
- Naprawiono awarię dodatku do czyszczenia dla niektórych plików XLIFF.
- Allow setting up localization CDN in Docker image.

4.32 Weblate 4.2

Released on August 18th 2020.

- Ulepszone strony użytkowników i dodano listę użytkowników.
- Zrezygnowano z wsparcia dla migracji z wersji 3.x, migracji za pośrednictwem wersji 4.1 lub 4.0.
- Dodano eksport do kilku jednojęzycznych formatów.
- Ulepszone wykresy aktywności.
- Number of displayed nearby strings can be configured.
- Dodano obsługę blokowania komponentów, w których występują błędy repozytorium.
- Simplified main navigation (replaced buttons with icons).
- Improved language code handling in Google Translate integration.
- The Git squash add-on can generate `Co-authored-by:` trailers.
- Ulepszony analizator wyszukiwania zapytań.
- Improved user feedback from format strings checks.
- Improved performance of bulk state changes.

- Added compatibility redirects after project or component renaming.
- Added notifications for strings approval, component locking and license change.
- Dodano wsparcie dla ModernMT.
- Allow to avoid overwriting approved translations on file upload.
- Dropped support for some compatibility URL redirects.
- Dodano sprawdzanie literałów szablonów ECMAScript.
- Dodano opcję obserwowania komponentu.
- Removed leading dot from JSON unit keys.
- Usunięto oddzielną kolejkę Celery dla pamięci tłumaczeniowej.
- Zezwalanie na jednoczesne tłumaczenie wszystkich komponentów na jeden język.
- Allow to configure `Content-Security-Policy` HTTP headers.
- Added support for aliasing languages at project level.
- New add-on to help with HTML or JavaScript localization, see *Lokalizacja JavaScript CDN*.
- The Weblate domain is now configured in the settings, see `SITE_DOMAIN`.
- Dodaj obsługę wyszukiwania według komponentu i projektu.

4.33 Weblate 4.1.1

Wydano 19 czerwca 2020 r.

- Fixed changing autofix or add-ons configuration in Docker.
- Fixed possible crash in „About” page.
- Improved installation of byte-compiled locale files.
- Naprawiono dodawanie słów do słownika.
- Naprawiono skróty klawiaturowe dla maszyn.
- Removed debugging output causing discarding log events in some setups.
- Fixed lock indication on project listing.
- Fixed listing GPG keys in some setups.
- Added option for which DeepL API version to use.
- Added support for acting as SAML Service Provider, see *Uwierzytelnianie SAML*.

4.34 Weblate 4.1

Wydano 15 czerwca 2020 roku.

- 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.
- Ulepszone dopasowanie pamięci tłumaczeniowej.
- Grupuj te same wyniki maszynowe.
- Add direct link to edit screenshot from translate page.
- Ulepszone okno dialogowe potwierdzenia usunięcia.
- 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: *Ciągi znaków Laravel PHP*, *Pliki HTML*, *Format OpenDocument*, *Format IDML*, *Pliki RC systemu Windows*, *Tłumaczenia INI*, *Tłumaczenia Inno Setup INI*, *Właściwości GWT*, *go-i18n pliki JSON*, *Plik ARB*.
- Consistently use dismissed as state of dismissed checks.
- Dodano obsługę konfigurowania domyślnych dodatków do włączenia.
- Fixed editor keyboard shortcut to dismiss checks.
- Improved machine translation of strings with placeholders.
- Pokazywanie tłumaczenia-ducha dla języków użytkownika, aby ułatwić rozpoczęcie pracy.
- Ulepszone analizowanie kodu języka.
- Show translations in user language first in the list.
- Renamed shapings to more generic name variants.
- Added new quality checks: *Wiele nienazwanych zmiennych*, *Długo nieprzetłumaczone*, *Kolejne zduplikowane wyrazy*.
- 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 *Dostosowywanie zachowania za pomocą flag*.
- Always show rendered text check if enabled.
- API now supports filtering of changes.
- Added support for sharing glossaries between projects.

4.35 Weblate 4.0.4

Wydano 7 maja 2020 r.

- 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.
- Naprawiono listę zrzutów ekranu.
- Fixed monthly digest notifications.
- Fixed intermediate translation behavior with units non existing in translation.

4.36 Weblate 4.0.3

Wydano 2 maja 2020 r.

- Naprawiono możliwą awarię w raportach.
- 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.

4.37 Weblate 4.0.2

Wydano 27 kwietnia 2020.

- Poprawiono wydajność statystyk tłumaczenia.
- Poprawiono wydajność zmiany etykiet.
- Poprawiono wydajność edycji zbiorczej.
- Poprawiono wydajność pamięci tłumaczeniowej.
- 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.
- Naprawiono eksport XLIFF z symbolami zastępczymi.
- Fixed false positive with zero width check.
- Ulepszono raportowanie błędów konfiguracji.
- Naprawiono dwujęzyczne przesyłanie źródeł.

- Automatically detect supported languages for DeepL machine translation.
- Fixed progress bar display in some corner cases.
- Fixed some checks triggering on non translated strings.

4.38 Weblate 4.0.1

Wydano 16 kwietnia 2020.

- Naprawiono instalację pakietu z PyPI.

4.39 Weblate 4.0

Wydano 16 kwietnia 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.
- Wyświetlanie sugestii nieudanych kontroli.
- Record source string changes in history.
- Upgraded Microsoft Translator to version 3 API.
- Reimplemented translation memory backend.
- Added support for several `is:` lookups in *Wyszukiwanie*.
- Allow to make *Niezmienione tłumaczenie* avoid internal blacklist.
- Improved comments extraction from monolingual po files.
- Zmieniono nazwy wiadomości kokpitu na ogłoszenia.
- Fixed occasional problems with registration mails.
- Improved LINGUAS update add-on to handle more syntax variants.
- Fixed editing monolingual XLIFF source file.
- Added support for exact matching in *Wyszukiwanie*.
- 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.

4.40 Seria Weblate 3.x

4.40.1 Weblate 3.11.3

Wydano 11 marca 2020.

- Fixed searching for fields with certain priority.
- Fixed predefined query for recently added strings.
- Naprawiono wyszukiwanie zwracające zduplikowane dopasowania.
- Naprawiono renderowanie powiadomień w Gmailu.
- 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.
- Naprawiono linki dokumentacji dla dodatków.
- 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.

4.40.2 Weblate 3.11.2

Wydano 22 lutego 2020 roku.

- Poprawiono renderowanie sugestii.
- Fixed some strings wrongly reported as having no words.

4.40.3 Weblate 3.11.1

Wydano 20 lutego 2020 roku.

- Udokumentowane zmiany konfiguracji Celery.
- 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.
- Ulepszone dopasowywanie słownika.

4.40.4 Weblate 3.11

Wydano 17 lutego 2020.

- Allow using VCS push URL during component creation via API.
- Rendered width check now shows image with the render.
- Poprawione linki w e-mailach z powiadomieniami.
- Poprawiono wygląd wiadomości e-mail w postaci zwykłego tekstu.
- Display ignored checks and allow to make them active again.
- Display nearby keys on monolingual translations.
- Dodano wsparcie dla grupowania kształtów ciągów.
- 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 file mask twice.
- Poprawiono wyodrębnianie obiektów do umieszczenia XML.
- The `SINGLE_PROJECT` can now enforce redirection to chosen project.
- Dodano opcję rozwiązywania komentarzy.
- Dodano zbiorczą edycję flag.
- Dodano wsparcie dla labels.
- Dodano dodatek do edycji zbiorczej.
- Dodano opcję dla *Wymuszanie kontroli*.
- Increased default validity of confirmation links.
- Ulepszona integracja Matomo.
- Fixed *Zostało przetłumaczone* to correctly handle source string change.
- Extended automatic updates configuration by `AUTO_UPDATE`.
- Dodatki LINGUAS wykonują teraz pełną synchronizację tłumaczeń w Weblate.

4.40.5 Weblate 3.10.3

Wydano 18 stycznia 2020.

- Obsługa translate-toolkit 2.5.0.

4.40.6 Weblate 3.10.2

Wydano 18 stycznia 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.
- Ulepszone przetwarzanie powiadomień w tle.
- Fixed broken form submission when navigating back in web browser.
- New add-on to configure YAML formatting.
- Fixed same plurals check to not fire on single plural form languages.
- Fixed regex search on some fields.

4.40.7 Weblate 3.10.1

Released on January 9th 2020.

- Extended API with translation creation.
- Fixed several corner cases in data migrations.
- Zgodność z Django 3.0.
- Poprawiona wydajność czyszczenia danych.
- 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.

4.40.8 Weblate 3.10

Released on December 20th 2019.

- Ulepszono interfejs użytkownika aplikacji.
- Dodano kontrolę podwójnej spacji.
- Naprawiono tworzenie nowych języków.
- Avoid sending auditlog notifications to deleted e-mails.
- Dodano obsługę ciągów tylko do odczytu.
- Added support for Markdown in comments.
- Allow placing translation instruction text in project info.
- Add copy to clipboard for secondary languages.
- Ulepszone wsparcie dla 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.
- Ulepszono lista tłumaczeń.
- Rozszerzone możliwości wyszukiwania.
- 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.

4.40.9 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.

4.40.10 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.
- Dodano dodatek automatycznego tłumaczenia do rozpoczynania tłumaczeń.
- Extend automatic translation to add suggestions.
- Display add-on 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 add-on for RESX to avoid unwanted file updates.
- Dodano zaawansowane możliwości wyszukiwania.
- 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.

4.40.11 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.
- Poprawiona wydajność dodatków związanych z repozytorium.
- Improved look of notification e-mails.
- Naprawiono zachowanie resetowania hasła.
- Improved performance on most of translation pages.
- Fixed listing of languages not known to Weblate.
- Dodano obsługę klonowania dodatków do wykrytych komponentów.
- 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.
- Dodano widoki zarządzania językami.
- 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.

4.40.12 Weblate 3.7.1

Released on June 28th 2019.

- Aktualizacje dokumentacji.
- Poprawiono pewne ograniczenia wymagań.
- Zaktualizowano bazę danych języka.
- Aktualizacje lokalizacji.
- Różne poprawki interfejsu użytkownika.
- Improved handling of unsupported but discovered translation files.
- More verbosely report missing file format requirements.

4.40.13 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.
- Dodano obsługę konfigurowania dodatku msgmerge.
- Delay opening SMTP connection when sending notifications.
- Ulepszono rejestrowanie błędów.
- Pozwolono na niestandardowe lokalizacje w dodatku do generowania MO.
- Dodano dodatki do czyszczenia starych sugestii lub komentarzy.
- 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.
- Dodano narzędzie do zarządzania czcionkami.
- 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.

4.40.14 Weblate 3.6.1

Released on April 26th 2019.

- Improved handling of monolingual XLIFF files.
- Fixed digest notifications in some corner cases.
- Fixed add-on 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.

4.40.15 Weblate 3.6

Released on April 20th 2019.

- Add support for downloading user data.
- Dodatki są teraz automatycznie uruchamiane podczas instalacji.
- Improved instructions for resolving merge conflicts.
- Cleanup add-on 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 add-ons 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.
- Dodano powiadomienia podsumowujące.
- 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.

4.40.16 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 add-on 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 Zeep releases.
- Aktualizacje lokalizacji.

4.40.17 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.
- Add-on 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.
- Zgodność z Tesseract 4.0.
- Billing add-on now removes projects for unpaid billings after 45 days.

4.40.18 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.
- Nowy dodatek do squashowania commitów przed wysłaniem.
- 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.

4.40.19 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 resname as context when available.
- Added support for XLIFF states.
- Added check for non writable files in DATA_DIR.
- Improved CSV export for changes.

4.40.20 Weblate 3.2.2

Released on October 20th 2018.

- Remove no longer needed Babel dependency.
- Uaktualnione definicje językowe.
- Poprawiona dokumentacja dla dodatków, LDAP i Celery.
- Fixed enabling new dos-eol and auto-java-messageformat flags.
- Fixed running setup.py test from PyPI package.
- Ulepszono obsługa liczby mnogiej.
- Fixed translation upload API failure in some corner cases.
- Fixed updating Git configuration in case it was changed manually.

4.40.21 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.

4.40.22 Weblate 3.2

Released on October 6th 2018.

- Add install_addon management command for automated add-on 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 add-ons.
- 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.

- Konfigurowalny tryb edycji (zen/pełny).
- Added support for error reporting to Sentry.
- Added support for automated daily update of repositories.
- Added support for creating projects and components by users.
- Wbudowana pamięć tłumaczeniowa teraz automatycznie przechowuje wykonane tłumaczenia.
- 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.

4.40.23 Weblate 3.1.1

Released on July 27th 2018.

- Fix testsuite failure on some setups.

4.40.24 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.
- Zgodność z Django 2.1.
- Django system checks are now used to diagnose problems with installation.
- Removed support for soon shutdown libravatar service.
- Nowy dodatek do oznaczania niezmienionych tłumaczeń jako wymagających edycji.
- 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.

4.40.25 Weblate 3.0.1

Released on June 10th 2018.

- Fixed possible migration issue from 2.20.
- Aktualizacje lokalizacji.
- Usunięto przestarzałe przykłady hooków.
- Ulepszono dokumentacje buforowania.
- Fixed displaying of admin documentation.
- Improved handling of long language names.

4.40.26 Weblate 3.0

Released on June 1st 2018.

- Przepisano kontrola dostępu.
- Several code cleanups that lead to moved and renamed modules.
- Nowy dodatek do automatycznego wykrywania komponentów.
- The import_project management command has now slightly different parameters.
- Added basic support for Windows RC files.
- New add-on to store contributor names in PO file headers.
- The per component hook scripts are removed, use add-ons instead.
- Add support for collecting contributor agreements.
- Access control changes are now tracked in history.
- Nowy dodatek zapewniający, że wszystkie komponenty w projekcie mają takie same tłumaczenia.
- Support for more variables in commit message templates.
- Add support for providing additional textual context.

4.41 Seria Weblate 2.x

4.41.1 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 add-on to configure gettext output wrapping.
- New add-on 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.
- Dodano wbudowaną pamięć tłumaczeń.
- 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.
- Dodano wsparcie do zmiany kolejności scommitowanych zmian.

4.41.2 Weblate 2.19.1

Released on February 20th 2018.

- Fixed migration issue on upgrade from 2.18.
- Improved file upload API validation.

4.41.3 Weblate 2.19

Released on February 15th 2018.

- 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.
- Zgodność z 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.
- Ulepszona wbudowana usługa tłumaczenia maszynowego.
- Added support for SAP Translation Hub service.
- Added support for Microsoft Terminology service.
- Removed support for advertisement in notification e-mails.
- Improved translation progress reporting at language level.
- Improved support for different plural formulas.
- Added support for Subversion repositories not using stdlayout.
- Dodano dodatki do dostosowywania przepływów pracy w tłumaczeniu.

4.41.4 Weblate 2.18

Released on December 15th 2017.

- Rozszerzone statystyki współautorów.
- 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.

4.41.5 Weblate 2.17.1

Released on October 13th 2017.

- Fixed running testsuite in some specific situations.
- Aktualizacje ustawień regionalnych.

4.41.6 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.
- Rozszerzone możliwości wyszukiwania.
- 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.

4.41.7 Weblate 2.16

Released on August 11th 2017.

- Różne ulepszenia wydajności.
- Added support for nested JSON format.
- Added support for WebExtension JSON format.
- Poprawiono uwierzytelnianie eksportera git.
- 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.
- Różne porządki w interfejsie użytkownika.
- Fixed component/project/site wide search for translations.

4.41.8 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.

4.41.9 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.

4.41.10 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.

4.41.11 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.

4.41.12 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.
- Różne drobne ulepszenia interfejsu użytkownika.

4.41.13 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.

4.41.14 Weblate 2.11

Released on Jan 31st 2017.

- Include language detailed information on language page.
- Ulepszenia backendu Mercurial.
- 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.
- Ulepszono dokumentację deweloperską.
- Better compatibility with various Git versions in Git exporter add-on.
- 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.
- Różne ulepszenia wydajności.

4.41.15 Weblate 2.10.1

Released on Jan 20th 2017.

- Do not leak account existence on password reset form (CVE-2017-5537).

4.41.16 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.
- Dodano widok matrycy dla szybkiego przeglądu wszystkich tłumaczeń.
- Added basic API for changes and strings.
- Added support for Apertium APy server for machine translations.

4.41.17 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.

4.41.18 Weblate 2.8

Released on Aug 31st 2016.

- Poprawa dokumentacji.
- Tłumaczenia.
- Zaktualizowano dołączone biblioteki JavaScript.
- Dodano polecenie zarządzania list_translators.
- Django 1.8 is no longer supported.
- Fixed compatibility with Django 1.10.
- Dodano obsługę Subversion.
- 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.

4.41.19 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.
- Zgodność z Mercurial 3.8.
- Dodano polecenie zarządzania import_json.
- 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.

4.41.20 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.
- Dodano REST API.
- Dodano odność z Django 1.10.
- Added categories to whiteboard messages.

4.41.21 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.
- Poprawa dokumentacji.
- 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.
- Poprawiono wyszukiwanie na całej stronie pod kątem nieudanych kontroli.

- 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.
- Zoptymalizowany indeks wyszukiwania pełnotekstowego.
- Added management command for auto translation.
- Dodano podświetlenie „obiektów do umieszczenia”.
- Added keyboard shortcuts for placeables, checks and machine translations.
- Ulepszono blokowanie tłumaczeń.
- Added quality check for AngularJS interpolation.
- Added extensive group based ACLs.
- Clarified terminology on strings needing edit (formerly fuzzy).
- Clarified terminology on strings needing action and untranslated strings.
- Wsparcie dla Pythona 3.
- Dropped support for Django 1.7.
- Dropped dependency on msginit for creating new gettext PO files.
- Dodano konfigurowalne widoki kokpitu.
- 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.
- Poprawa dokumentacji.
- Added support for configurable automatic group assignments.
- Improved adding of new translations.

4.41.22 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.
- Ulepszony format pliku - autodetekcja.
- Rozszerzono skróty klawiaturowe.
- 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.

4.41.23 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.

4.41.24 Weblate 2.2

Released on Feb 19th 2015.

- Poprawa wydajności.
- Fulltext search on location and comments fields.
- Nowe wykresy aktywności oparte na SVG/JavaScript.
- Wsparcie dla Django 1.8.
- Wsparcie dla usuwania komentarzy.
- 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.

4.41.25 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.
- Poprawa dokumentacji.
- Różne poprawki błędów.
- Automatic hiding of columns in translation listing for small screens.
- Changed configuration of filesystem paths.
- Improved SSH keys handling and storage.
- Ulepszono blokowanie repozytorium.
- Customizable quality checks per source string.
- Allow to hide completed translations from dashboard.

4.41.26 Weblate 2.0

Released on Nov 6th 2014.

- New responsive UI using Bootstrap.
- Przepisany backend VCS.
- Poprawa dokumentacji.
- Added whiteboard for site wide messages.
- Konfigurowalny priorytet ciągów.
- Added support for JSON file format.

- Fixed generating mo files in certain cases.
- Added support for GitLab notifications.
- Added support for disabling translation suggestions.
- Obsługa Django 1.7.
- ACL projects now have user management.
- Rozszerzone możliwości wyszukiwania.
- Give more hints to translators about plurals.
- Naprawiono blokowanie repozytorium Git.
- Compatibility with older Git versions.
- Ulepszono obsługę listy ACL.
- Added buttons for per language quotes and other special characters.
- Support for exporting stats as JSONP.

4.42 Seria Weblate 1.x

4.42.1 Weblate 1.9

Released on May 6th 2014.

- Zgodność z Django 1.6.
- 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.
- Awatary można wyłączyć.
- Merged first and last name attributes.
- Avatars are now fetched and cached server side.
- Added support for shields.io badge.

4.42.2 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.
- Poprawa dokumentacji.
- Ulepszony przegląd ciągów źródłowych.
- Wyszukiwanie we wszystkich ciągach.

- Better tracking of source strings.
- Ochrona Captcha do rejestracji.

4.42.3 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.
- Szczegółowe statystyki tłumaczeń.
- 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.
- Dodaj widżet odznaki statusu.
- Dropped invalid text direction check.
- Changes in dictionary are now logged in history.
- Ulepszenia wydajności dla widoku tłumaczenia.

4.42.4 Weblate 1.6

Released on July 25th 2013.

- Nicer error handling on registration.
- Przeglądanie zmian.
- 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.
- Różne ulepszenia sprawdzania poprawności formularza.
- Różne ulepszenia kontroli jakości.
- Ulepszono wydajności importu.
- Added support for voting on suggestions.
- Oczyszczanie interfejsu administratora.

4.42.5 Weblate 1.5

Wydano 16 kwietnia 2013 r.

- Please check manual for upgrade instructions.
- Dodano publiczne strony użytkowników.
- Lepsze nazewnictwo form liczby mnogiej.
- Dodano wsparcie dla eksportu słownika TBX.
- Dodano wsparcie dla powiadomień Bitbucket.
- Wykresy aktywności są teraz dostępne dla każdego tłumaczenia, języka lub użytkownika.
- Rozszerzone opcje polecenia administratora import_project.
- Kompatybilność z Django 1.5.
- Avatars are now shown using libravatar.
- Added possibility to pretty print JSON export.
- Różne ulepszenia wydajności.
- 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.

4.42.6 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.
- Ulepszona obsługa blokowania tłumaczeń.
- 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.
- Różne ulepszenia kontroli jakości.
- Support for per project ACL.
- Basic code coverage by unit tests.

4.42.7 Weblate 1.3

Released on November 16th 2012.

- Compatibility with PostgreSQL database backend.
- Removes languages removed in upstream git repository.
- Ulepszone przetwarzanie kontroli jakości.
- Added new checks (BBCode, 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.
- Bardziej widoczne eksporty danych.
- New buttons to enter some special characters.
- Obsługa eksportowania słownika.
- 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.
- Lepsze śledzenie dziennika zmian.
- Changes can now be monitored using RSS.
- Improved support for RTL languages.

4.42.8 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.

- Prosta ochrona antyspamowa.
- 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.
- Buforowanie liczby ciągów z nieudanymi kontrolami.
- Checking of requirements during setup.
- Poprawa dokumentacji.

4.42.9 Weblate 1.1

Released on July 4th 2012.

- Poprawiono kilka tłumaczeń.
- 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.

4.42.10 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.
- Ulepszono instrukcje instalacji.
- Ulepszona nawigacja w słowniku.

4.43 Seria Weblate 0.x

4.43.1 Weblate 0.9

Released on April 18th 2012.

- Fixed import of unknown languages.
- Improved listing of nearby messages.
- Poprawiono kilka kontroli.
- Aktualizacje dokumentacji.

- Added definition for several more languages.
- Różne porządki w kodzie.
- Poprawa dokumentacji.
- Zmieniono układ pliku.
- Update helper scripts to Django 1.4.
- Ulepszona nawigacja podczas tłumaczenia.
- 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 untranslated strings ignores format string only messages.

4.43.2 Weblate 0.8

Released on April 3rd 2012.

- Replaced own full text search with Whoosh.
- Various fixes and improvements to checks.
- Nowe polecenie updatechecks.
- Wiele aktualizacji tłumaczeń.
- 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.
- Zgodność z Django 1.4.
- Ability to push changes to remote repo from web interface.
- Added review of translations done by others.

4.43.3 Weblate 0.7

Released on February 16th 2012.

- Direct support for GitHub notifications.
- Added support for cleaning up orphaned checks and translations.
- Displays nearby strings while translating.
- Displays similar strings while translating.
- Ulepszone wyszukiwanie ciągów.

4.43.4 Weblate 0.6

Released on February 14th 2012.

- Added various checks for translated messages.
- Regulowana kontrola dostępu.
- Improved handling of translations with new lines.
- Added client side sorting of tables.
- Please check upgrading instructions in case you are upgrading.

4.43.5 Weblate 0.5

Released on February 12th 2012.

- **Support for machine translation using following online services:**
  - Apertium
  - Microsoft Translator
  - MyMemory
- Kilka nowych tłumaczeń.
- 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).

4.43.6 Weblate 0.4

Released on February 8th 2012.

- Added usage guide to documentation.
- Fixed API hooks not to require CSRF protection.

4.43.7 Weblate 0.3

Released on February 8th 2012.

- 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.
- Nowe statystyki dla poszczególnych języków.

4.43.8 Weblate 0.2

Released on February 7th 2012.

- Improved validation of several forms.
- Warn users on profile upgrade.
- Zapamiętaj adres URL logowania.
- Naming of text areas while entering plural forms.
- Automatic expanding of translation area.

4.43.9 Weblate 0.1

Released on February 6th 2012.

- Wydanie początkowe.

Indeks modułów Pythona

w

`wlc`, 156
`wlc.config`, 157
`wlc.main`, 157

HTTP Routing Table

/

ANY /, 103

/api

GET /api/, 106

/api/addons

GET /api/addons/, 143
GET /api/addons/(int:id)/, 143
PUT /api/addons/(int:id)/, 144
DELETE /api/addons/(int:id)/, 144
PATCH /api/addons/(int:id)/, 144

/api/changes

GET /api/changes/, 140
GET /api/changes/(int:id)/, 141

/api/component-lists

GET /api/component-lists/, 144
GET /api/component-lists/(str:slug)/, 144
POST /api/component-lists/(str:slug)/components/, 145
PUT /api/component-lists/(str:slug)/, 145
DELETE /api/component-lists/(str:slug)/, 145
DELETE /api/component-lists/(str:slug)/components/(str:component_slug), 145
PATCH /api/component-lists/(str:slug)/, 145

/api/components

GET /api/components/, 122
GET /api/components/(string:project)/(string:comp 123
GET /api/components/(string:project)/(string:comp 127
GET /api/components/(string:project)/(string:comp 127
GET /api/components/(string:project)/(string:compo 132
GET /api/components/(string:project)/(string:compo 128
GET /api/components/(string:project)/(string:compo 130
GET /api/components/(string:project)/(string:compo 130
GET /api/components/(string:project)/(string:compo 129
GET /api/components/(string:project)/(string:compo 127
GET /api/components/(string:project)/(string:compo 132
GET /api/components/(string:project)/(string:compo 130
POST /api/components/(string:project)/(string:comp 144
POST /api/components/(string:project)/(string:comp 132
POST /api/components/(string:project)/(string:comp 128
POST /api/components/(string:project)/(string:comp 129
POST /api/components/(string:project)/(string:comp 130
PUT /api/components/(string:project)/(string:compo 126
DELETE /api/components/(string:project)/(string:co 127
DELETE /api/components/(string:project)/(string:co 132
PATCH /api/components/(string:project)/(string:com 125

/api/groups

GET /api/groups/, 110
GET /api/groups/(int:id)/, 110
POST /api/groups/, 110
POST /api/groups/(int:id)/componentlists/, 113
POST /api/groups/(int:id)/components/, 112

POST /api/groups/(int:id)/languages/, 112
POST /api/groups/(int:id)/projects/, 112
POST /api/groups/(int:id)/roles/, 112
PUT /api/groups/(int:id)/, 111
DELETE /api/groups/(int:id)/, 111
DELETE /api/groups/(int:id)/componentlists/(int:slug)/, 113
DELETE /api/groups/(int:id)/components/(int:id), 112
DELETE /api/groups/(int:id)/languages/(string:language_code), 112
DELETE /api/groups/(int:id)/projects/(int:project_id), 112
PATCH /api/groups/(int:id)/, 111

/api/languages

GET /api/languages/, 114
GET /api/languages/(string:language)/, 114
GET /api/languages/(string:language)/statistics/, 116
POST /api/languages/, 114
PUT /api/languages/(string:language)/, 115
DELETE /api/languages/(string:language)/, 116
PATCH /api/languages/(string:language)/, 115

/api/metrics

GET /api/metrics/, 146

/api/projects

GET /api/projects/, 116
GET /api/projects/(string:project)/, 116
GET /api/projects/(string:project)/changes/, 118
GET /api/projects/(string:project)/components/, 119
GET /api/projects/(string:project)/languages/, 122
GET /api/projects/(string:project)/repository/, 118
GET /api/projects/(string:project)/statistics/, 122
POST /api/projects/, 116
POST /api/projects/(string:project)/components/, 119
POST /api/projects/(string:project)/repository/, 118
PUT /api/projects/(string:project)/, 117
DELETE /api/projects/(string:project)/, 117
PATCH /api/projects/(string:project)/, 117

/api/roles

GET /api/roles/, 113
GET /api/roles/(int:id)/, 113
POST /api/roles/, 113
PUT /api/roles/(int:id)/, 114
DELETE /api/roles/(int:id)/, 114
PATCH /api/roles/(int:id)/, 114

/api/screenshots

GET /api/screenshots/, 141
GET /api/screenshots/(int:id)/, 141
GET /api/screenshots/(int:id)/file/, 141
POST /api/screenshots/, 142
POST /api/screenshots/(int:id)/file/, 141
POST /api/screenshots/(int:id)/units/, 142
PUT /api/screenshots/(int:id)/, 143
DELETE /api/screenshots/(int:id)/, 143
DELETE /api/screenshots/(int:id)/units/(int:unit_
PATCH /api/screenshots/(int:id)/, 143

/api/tasks

GET /api/tasks/, 146
GET /api/tasks/(str:uuid)/, 146

/api/translations

GET /api/translations/, 133
GET /api/translations/(string:project)/(string:co
133
GET /api/translations/(string:project)/(string:co
135
GET /api/translations/(string:project)/(string:co
136
GET /api/translations/(string:project)/(string:co
137
GET /api/translations/(string:project)/(string:co
138
GET /api/translations/(string:project)/(string:co
135
POST /api/translations/(string:project)/(string:c
136
POST /api/translations/(string:project)/(string:c
137
POST /api/translations/(string:project)/(string:c
137
POST /api/translations/(string:project)/(string:c
136
DELETE /api/translations/(string:project)/(string
135

/api/units

GET /api/units/, 138

GET /api/units/(int:id)/, 138
PUT /api/units/(int:id)/, 140
DELETE /api/units/(int:id)/, 140
PATCH /api/units/(int:id)/, 139

/api/users

GET /api/users/, 107
GET /api/users/(str:username)/, 107
GET /api/users/(str:username)/notifications/, 109
GET /api/users/(str:username)/notifications/(int:subscription_id)/, 109
GET /api/users/(str:username)/statistics/, 108
POST /api/users/, 107
POST /api/users/(str:username)/groups/, 108
POST /api/users/(str:username)/notifications/, 109
PUT /api/users/(str:username)/, 108
PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 109
DELETE /api/users/(str:username)/, 108
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 110
PATCH /api/users/(str:username)/, 108
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 109

/exports

GET /exports/rss/, 150
GET /exports/rss/(string:project)/, 150
GET /exports/rss/(string:project)/(string:component)/, 150
GET /exports/rss/(string:project)/(string:component)/(string:language)/, 150
GET /exports/rss/language/(string:language)/, 150
GET /exports/stats/(string:project)/(string:component)/, 149

/hooks

GET /hooks/update/(string:project)/, 147
GET /hooks/update/(string:project)/(string:component)/, 147
POST /hooks/azure/, 148
POST /hooks/bitbucket/, 147
POST /hooks/gitea/, 148
POST /hooks/gitee/, 148
POST /hooks/github/, 147
POST /hooks/gitlab/, 147
POST /hooks/pagure/, 148

Indeks

Symbole

.XML resource file
    file format, 88
--add
    auto_translate command line option, 379
--addon
    install_addon command line option, 386
--age
    commit_pending command line option, 380
--author
    add_suggestions command line option, 379
--author-email
    wlc command line option, 154
--author-name
    wlc command line option, 154
--base-file-template
    import_project command line option, 384
--check
    importusers command line option, 385
--config
    wlc command line option, 152
--config-section
    wlc command line option, 152
--configuration
    install_addon command line option, 386
--convert
    wlc command line option, 153
--email
    createadmin command line option, 381
--file-format
    import_project command line option, 384
--force
    loadpo command line option, 387
--force-commit
    pushgit command line option, 388
--format
    wlc command line option, 152
--fuzzy
    wlc command line option, 154
--ignore
    import_json command line option, 382
--inconsistent
    auto_translate command line option, 379
--input
    wlc command line option, 153
--key
    wlc command line option, 152
--lang
    loadpo command line option, 387
--language-code
    list_translators command line option, 386
--language-map
    import_memory command line option, 383
--language-regex
    import_project command line option, 384
--license
    import_project command line option, 384
--license-url
    import_project command line option, 384
--main-component
    import_json command line option, 382
    import_project command line option, 384
--method
    wlc command line option, 154
--mode
    auto_translate command line option, 379
--mt
    auto_translate command line option, 379
--name
    createadmin command line option, 381
--name-template

import_project command line option, 384
--new-base-template
import_project command line option, 384
--no-password
createadmin command line option, 381
--no-privs-update
setupgroups command line option, 388
--no-projects-update
setupgroups command line option, 388
--no-update
setuplang command line option, 389
--output
wlc command line option, 153
--overwrite
auto_translate command line option, 379
wlc command line option, 153
--password
createadmin command line option, 381
--project
import_json command line option, 382
--source
auto_translate command line option, 379
--threshold
auto_translate command line option, 379
--update
createadmin command line option, 381
import_json command line option, 382
install_addon command line option, 386
--url
wlc command line option, 152
--user
auto_translate command line option, 379
--username
createadmin command line option, 381
--vcs
import_project command line option, 384

A

add_suggestions
weblate admin command, 379
add_suggestions command line option
--author, 379
ADMINS
setting, 205
AKISMET_API_KEY
setting, 337
ALLOWED_HOSTS
setting, 205
Android
file format, 83
ANONYMOUS_USER_NAME
setting, 337
API, 103, 150, 155
Apple strings
file format, 84
ARB
file format, 87
AUDITLOG_EXPIRY
setting, 338
AUTH_LOCK_ATTEMPTS
setting, 338
AUTH_TOKEN_VALID
setting, 339
auto_translate
weblate admin command, 379
auto_translate command line option
--add, 379
--inconsistent, 379
--mode, 379
--mt, 379
--overwrite, 379
--source, 379
--threshold, 379
--user, 379
AUTO_UPDATE
setting, 338
AUTOFIX_LIST
setting, 339
AVATAR_URL_PREFIX
setting, 338

B

BACKGROUND_TASKS
setting, 340
BASE_DIR
setting, 340
BaseAddon (*klasa w module weblate.addons.base*), 424
BASIC_LANGUAGES
setting, 340
bilingual
translation, 74
BORG_EXTRA_ARGS
setting, 340

C

can_install() (*weblate.addons.base.BaseAddon metoda klasy*), 424
CELERY_BACKUP_OPTIONS, 163, 179
CELERY_BEAT_OPTIONS, 163, 179
CELERY_MAIN_OPTIONS, 163, 179
CELERY_MEMORY_OPTIONS, 163, 179
CELERY_NOTIFY_OPTIONS, 163, 179
celery_queues
weblate admin command, 380
CELERY_TRANSLATE_OPTIONS, 163, 179
changes
wlc command line option, 153

CHECK_LIST
    setting, 341
checkgit
    weblate admin command, 380
cleanup
    wlc command line option, 153
cleanup_ssh_keys
    weblate admin command, 381
cleanuptrans
    weblate admin command, 381
Comma separated values
    file format, 89
Command (*klasa w module wlc.main*), 157
COMMENT_CLEANUP_DAYS
    setting, 342
commit
    wlc command line option, 152
commit_pending
    weblate admin command, 380
commit_pending command line option
    --age, 380
COMMIT_PENDING_HOURS
    setting, 342
commitgit
    weblate admin command, 380
configure() (*weblate.addons.base.BaseAddon metoda*), 424
CONTACT_FORM
    setting, 342
createadmin
    weblate admin command, 381
createadmin command line option
    --email, 381
    --name, 381
    --no-password, 381
    --password, 381
    --update, 381
    --username, 381
CSP_CONNECT_SRC
    setting, 341
CSP_FONT_SRC
    setting, 341
CSP_IMG_SRC
    setting, 341
CSP_SCRIPT_SRC
    setting, 341
CSP_STYLE_SRC
    setting, 341
CSV
    file format, 89

D

daily() (*weblate.addons.base.BaseAddon metoda*), 424
DATA_DIR
    setting, 342
DATABASE_BACKUP
    setting, 343
DATABASES
    setting, 206
DEBUG
    setting, 206
DEFAULT_ACCESS_CONTROL
    setting, 343
DEFAULT_ADD_MESSAGE
    setting, 344
DEFAULT_ADDON_MESSAGE
    setting, 344
DEFAULT_ADDONS
    setting, 344
DEFAULT_AUTO_WATCH
    setting, 343
DEFAULT_COMMIT_MESSAGE
    setting, 344
DEFAULT_COMMITER_EMAIL
    setting, 344
DEFAULT_COMMITER_NAME
    setting, 345
DEFAULT_DELETE_MESSAGE
    setting, 344
DEFAULT_FROM_EMAIL
    setting, 206
DEFAULT_LANGUAGE
    setting, 345
DEFAULT_MERGE_MESSAGE
    setting, 344
DEFAULT_MERGE_STYLE
    setting, 345
DEFAULT_PAGE_LIMIT
    setting, 353
DEFAULT_PULL_MESSAGE
    setting, 346
DEFAULT_RESTRICTED_COMPONENT
    setting, 344
DEFAULT_SHARED_TM
    setting, 345
DEFAULT_TRANSLATION_PROPAGATION
    setting, 345
download
    wlc command line option, 153
DTD
    file format, 91
dump_memory
    weblate admin command, 382
dumpuserdata
    weblate admin command, 382

E

ENABLE_AVATARS
    setting, 346
ENABLE_HOOKS
    setting, 346
ENABLE_HTTPS
    setting, 346
ENABLE_SHARING
    setting, 346

F

file format
    .XML resource file, 88
    Android, 83
    Apple strings, 84
    ARB, 87
    Comma separated values, 89
    CSV, 89
    DTD, 91
    gettext, 77
    go-i18n, 86
    GWT properties, 81
    i18next, 86
    INI translations, 81, 82
    Java properties, 80
    Joomla translations, 82
    JSON, 85
    mi18n lang, 81
    PHP strings, 84
    PO, 77
    Qt, 82
    RC, 91
    ResourceDictionary, 88
    RESX, 88
    Ruby YAML, 90
    Ruby YAML Ain't Markup Language, 90
    string resources, 83
    TS, 82
    WPF, 88
    XLIFF, 78
    XML, 91
    YAML, 90
    YAML Ain't Markup Language, 90

G

get() (*wlc.Weblate metoda*), 156
get_add_form() (*weblate.addons.base.BaseAddon metoda klasy*), 424
GET_HELP_URL
    setting, 347
get_settings_form() (*weblate.addons.base.BaseAddon metoda*), 424
gettext
    file format, 77
GITEA_CREDENTIALS
    setting, 347
GITEA_TOKEN
    setting, 347
GITEA_USERNAME
    setting, 347
GITHUB_CREDENTIALS
    setting, 348
GITHUB_TOKEN
    setting, 349
GITHUB_USERNAME
    setting, 349
GITLAB_CREDENTIALS
    setting, 347
GITLAB_TOKEN
    setting, 348
GITLAB_USERNAME
    setting, 348
go-i18n
    file format, 86
GOOGLE_ANALYTICS_ID
    setting, 349
GWT properties
    file format, 81

H

HIDE_REPO_CREDENTIALS
    setting, 349
HIDE_VERSION
    setting, 349

I

i18next
    file format, 86
import_demo
    weblate admin command, 382
import_json
    weblate admin command, 382
import_json command line option
    --ignore, 382
    --main-component, 382
    --project, 382
    --update, 382
import_memory
    weblate admin command, 383
import_memory command line option
    --language-map, 383
import_project
    weblate admin command, 383
import_project command line option
    --base-file-template, 384
    --file-format, 384
    --language-regex, 384
    --license, 384
    --license-url, 384
    --main-component, 384
    --name-template, 384
    --new-base-template, 384
    --vcs, 384
importuserdata
    weblate admin command, 385
importusers
    weblate admin command, 385
importusers command line option
    --check, 385
INI translations
    file format, 81, 82
install_addon
    weblate admin command, 386
install_addon command line option
    --addon, 386
    --configuration, 386

--update, 386
INTERLEDGER_PAYMENT_POINTERS
setting, 349
iOS
translation, 84
IP_BEHIND_REVERSE_PROXY
setting, 350
IP_PROXY_HEADER
setting, 350
IP_PROXY_OFFSET
setting, 350

J

Java properties
file format, 80
Joomla translations
file format, 82
JSON
file format, 85

L

LEGAL_URL
setting, 351
LICENSE_EXTRA
setting, 351
LICENSE_FILTER
setting, 351
LICENSE_REQUIRED
setting, 352
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
setting, 352
list_languages
weblate admin command, 386
list_translators
weblate admin command, 386
list_translators command line option
--language-code, 386
list_versions
weblate admin command, 387
list-components
wlc command line option, 152
list-languages
wlc command line option, 152
list-projects
wlc command line option, 152
list-translations
wlc command line option, 152
load() (*wlc.config.WeblateConfig metoda*), 157
loadpo
weblate admin command, 387
loadpo command line option
--force, 387
--lang, 387
LOCALIZE_CDN_PATH
setting, 352
LOCALIZE_CDN_URL
setting, 352
lock
wlc command line option, 153
lock_translation
weblate admin command, 387
lock-status
wlc command line option, 153
LOGIN_REQUIRED_URLS
setting, 352
LOGIN_REQUIRED_URLS_EXCEPTIONS
setting, 352
ls
wlc command line option, 152

M

main() (*w module wlc.main*), 157
MATOMO_SITE_ID
setting, 353
MATOMO_URL
setting, 353
mi18n lang
file format, 81
moduł
wlc, 156
wlc.config, 157
wlc.main, 157
monolingual
translation, 74
move_language
weblate admin command, 387

N

NEARBY_MESSAGES
setting, 353

P

PAGURE_CREDENTIALS
setting, 354
PAGURE_TOKEN
setting, 354
PAGURE_USERNAME
setting, 354
PHP strings
file format, 84
PIWIK_SITE_ID
setting, 353
PIWIK_URL
setting, 353
PO
file format, 77
post() (*wlc.Weblate metoda*), 156
post_add() (*weblate.addons.base.BaseAddon metoda*), 424
post_commit() (*weblate.addons.base.BaseAddon metoda*), 424
post_push() (*weblate.addons.base.BaseAddon metoda*), 424
post_update() (*weblate.addons.base.BaseAddon metoda*), 424

pre_commit() (*weblate.addons.base.BaseAddon metoda*), 425
pre_push() (*weblate.addons.base.BaseAddon metoda*), 425
pre_update() (*weblate.addons.base.BaseAddon metoda*), 425
PRIVACY_URL
    setting, 354
pull
    wlc command line option, 153
push
    wlc command line option, 153
pushgit
    weblate admin command, 388
pushgit command line option
    --force-commit, 388
Python, 155
Python Enhancement Proposals
    PEP 484, 421, 434

Q

Qt
    file format, 82

R

RATELIMIT_ATTEMPTS
    setting, 355
RATELIMIT_LOCKOUT
    setting, 355
RATELIMIT_WINDOW
    setting, 355
RC
    file format, 91
REDIS_PASSWORD, 176
register_command() (*w module wlc.main*), 157
REGISTRATION_ALLOW_BACKENDS
    setting, 355
REGISTRATION_CAPTCHA
    setting, 356
REGISTRATION_EMAIL_MATCH
    setting, 356
REGISTRATION_OPEN
    setting, 356
repo
    wlc command line option, 153
REPOSITORY_ALERT_THRESHOLD
    setting, 357
REQUIRE_LOGIN
    setting, 357
reset
    wlc command line option, 153
ResourceDictionary
    file format, 88
REST, 103
RESX
    file format, 88
RFC
    RFC 5646, 74
Ruby YAML
    file format, 90
Ruby YAML Ain't Markup Language
    file format, 90

S

save_state() (*weblate.addons.base.BaseAddon metoda*), 425
SECRET_KEY
    setting, 206
SENTRY_DSN
    setting, 357
SERVER_EMAIL
    setting, 206
SESSION_COOKIE_AGE_AUTHENTICATED
    setting, 357
SESSION_ENGINE
    setting, 206
setting
    ADMINS, 205
    AKISMET_API_KEY, 337
    ALLOWED_HOSTS, 205
    ANONYMOUS_USER_NAME, 337
    AUDITLOG_EXPIRY, 338
    AUTH_LOCK_ATTEMPTS, 338
    AUTH_TOKEN_VALID, 339
    AUTO_UPDATE, 338
    AUTOFIX_LIST, 339
    AVATAR_URL_PREFIX, 338
    BACKGROUND_TASKS, 340
    BASE_DIR, 340
    BASIC_LANGUAGES, 340
    BORG_EXTRA_ARGS, 340
    CHECK_LIST, 341
    COMMENT_CLEANUP_DAYS, 342
    COMMIT_PENDING_HOURS, 342
    CONTACT_FORM, 342
    CSP_CONNECT_SRC, 341
    CSP_FONT_SRC, 341
    CSP_IMG_SRC, 341
    CSP_SCRIPT_SRC, 341
    CSP_STYLE_SRC, 341
    DATA_DIR, 342
    DATABASE_BACKUP, 343
    DATABASES, 206
    DEBUG, 206
    DEFAULT_ACCESS_CONTROL, 343
    DEFAULT_ADD_MESSAGE, 344
    DEFAULT_ADDON_MESSAGE, 344
    DEFAULT_ADDONS, 344
    DEFAULT_AUTO_WATCH, 343
    DEFAULT_COMMIT_MESSAGE, 344
    DEFAULT_COMMITER_EMAIL, 344
    DEFAULT_COMMITER_NAME, 345
    DEFAULT_DELETE_MESSAGE, 344
    DEFAULT_FROM_EMAIL, 206
    DEFAULT_LANGUAGE, 345
    DEFAULT_MERGE_MESSAGE, 344

DEFAULT_MERGE_STYLE, 345
DEFAULT_PAGE_LIMIT, 353
DEFAULT_PULL_MESSAGE, 346
DEFAULT_RESTRICTED_COMPONENT, 344
DEFAULT_SHARED_TM, 345
DEFAULT_TRANSLATION_PROPAGATION, 345
ENABLE_AVATARS, 346
ENABLE_HOOKS, 346
ENABLE_HTTPS, 346
ENABLE_SHARING, 346
GET_HELP_URL, 347
GITEA_CREDENTIALS, 347
GITEA_TOKEN, 347
GITEA_USERNAME, 347
GITHUB_CREDENTIALS, 348
GITHUB_TOKEN, 349
GITHUB_USERNAME, 349
GITLAB_CREDENTIALS, 347
GITLAB_TOKEN, 348
GITLAB_USERNAME, 348
GOOGLE_ANALYTICS_ID, 349
HIDE_REPO_CREDENTIALS, 349
HIDE_VERSION, 349
INTERLEDGER_PAYMENT_POINTERS, 349
IP_BEHIND_REVERSE_PROXY, 350
IP_PROXY_HEADER, 350
IP_PROXY_OFFSET, 350
LEGAL_URL, 351
LICENSE_EXTRA, 351
LICENSE_FILTER, 351
LICENSE_REQUIRED, 352
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 352
LOCALIZE_CDN_PATH, 352
LOCALIZE_CDN_URL, 352
LOGIN_REQUIRED_URLS, 352
LOGIN_REQUIRED_URLS_EXCEPTIONS, 352
MATOMO_SITE_ID, 353
MATOMO_URL, 353
NEARBY_MESSAGES, 353
PAGURE_CREDENTIALS, 354
PAGURE_TOKEN, 354
PAGURE_USERNAME, 354
PIWIK_SITE_ID, 353
PIWIK_URL, 353
PRIVACY_URL, 354
RATELIMIT_ATTEMPTS, 355
RATELIMIT_LOCKOUT, 355
RATELIMIT_WINDOW, 355
REGISTRATION_ALLOW_BACKENDS, 355
REGISTRATION_CAPTCHA, 356
REGISTRATION_EMAIL_MATCH, 356
REGISTRATION_OPEN, 356
REPOSITORY_ALERT_THRESHOLD, 357
REQUIRE_LOGIN, 357
SECRET_KEY, 206
SENTRY_DSN, 357
SERVER_EMAIL, 206
SESSION_COOKIE_AGE_AUTHENTICATED, 357
SESSION_ENGINE, 206
SIMPLIFY_LANGUAGES, 357
SINGLE_PROJECT, 358
SITE_DOMAIN, 357
SITE_TITLE, 358
SPECIAL_CHARS, 358
SSH_EXTRA_ARGS, 358
STATUS_URL, 359
SUGGESTION_CLEANUP_DAYS, 359
UPDATE_LANGUAGES, 359
URL_PREFIX, 359
VCS_BACKENDS, 360
VCS_CLONE_DEPTH, 360
WEBLATE_ADDONS, 360
WEBLATE_EXPORTERS, 361
WEBLATE_FORMATS, 361
WEBLATE_GPG_IDENTITY, 362
WEBLATE_MACHINERY, 361
WEBSITE_REQUIRED, 362
setupgroups
weblate admin command, 388
setupgroups command line option
--no-privs-update, 388
--no-projects-update, 388
setuplang
weblate admin command, 389
setuplang command line option
--no-update, 389
show
wlc command line option, 152
SIMPLIFY_LANGUAGES
setting, 357
SINGLE_PROJECT
setting, 358
SITE_DOMAIN
setting, 357
SITE_TITLE
setting, 358
SPECIAL_CHARS
setting, 358
SSH_EXTRA_ARGS
setting, 358
statistics
wlc command line option, 153
STATUS_URL
setting, 359
store_post_load() (*weblate.addons.base.BaseAddon metoda*), 425
string resources
file format, 83
SUGGESTION_CLEANUP_DAYS
setting, 359

T

translation

bilingual, 74
iOS, 84
monolingual, 74
TS
file format, 82

U

unit_pre_create() (*weblate.addons.base.BaseAddon metoda*), 425
unlock
wlc command line option, 153
unlock_translation
weblate admin command, 388
UPDATE_LANGUAGES
setting, 359
updatechecks
weblate admin command, 389
updategit
weblate admin command, 389
upload
wlc command line option, 153
URL_PREFIX
setting, 359

V

VCS_BACKENDS
setting, 360
VCS_CLONE_DEPTH
setting, 360
version
wlc command line option, 152

W

WEB_WORKERS, 163, 179
Weblate (*klasa w module wlc*), 156
weblate admin command
add_suggestions, 379
auto_translate, 379
celery_queues, 380
checkgit, 380
cleanup_ssh_keys, 381
cleanuptrans, 381
commit_pending, 380
commitgit, 380
createadmin, 381
dump_memory, 382
dumpuserdata, 382
import_demo, 382
import_json, 382
import_memory, 383
import_project, 383
importuserdata, 385
importusers, 385
install_addon, 386
list_languages, 386
list_translators, 386
list_versions, 387
loadpo, 387
lock_translation, 387
move_language, 387
pushgit, 388
setupgroups, 388
setuplang, 389
unlock_translation, 388
updatechecks, 389
updategit, 389
WEBLATE_ADDONS
setting, 360
WEBLATE_ADMIN_EMAIL, 163--165
WEBLATE_ADMIN_NAME, 163--165
WEBLATE_ADMIN_PASSWORD, 159, 163--165
WEBLATE_ADMIN_PASSWORD_FILE, 165
WEBLATE_AKISMET_API_KEY, 396
WEBLATE_ALLOWED_HOSTS, 205, 210, 358
WEBLATE_API_RATELIMIT_ANON, 106
WEBLATE_API_RATELIMIT_USER, 106
WEBLATE_AUTH_LDAP_BIND_PASSWORD, 170
WEBLATE_EMAIL_HOST_PASSWORD, 177
WEBLATE_EMAIL_PORT, 176, 177
WEBLATE_EMAIL_USE_SSL, 176, 177
WEBLATE_EMAIL_USE_TLS, 176, 177
WEBLATE_ENABLE_HTTPS, 238
WEBLATE_EXPORTERS
setting, 361
WEBLATE_FORMATS
setting, 361
WEBLATE_GPG_IDENTITY
setting, 362
WEBLATE_LOCALIZE_CDN_PATH, 178
WEBLATE_MACHINERY
setting, 361
WEBLATE_RATELIMIT_ATTEMPTS, 398
WEBLATE_REQUIRE_LOGIN, 357
WEBLATE_SECURE_PROXY_SSL_HEADER, 166
WEBLATE_SERVICE, 163
WEBLATE_SILENCED_SYSTEM_CHECKS, 235
WEBLATE_SITE_DOMAIN, 207, 226, 358
WEBLATE_WORKERS, 163, 179, 180
WeblateConfig (*klasa w module wlc.config*), 157
WeblateException, 156
WEBSITE_REQUIRED
setting, 362
wlc, 150
moduł, 156
wlc command line option
--author-email, 154
--author-name, 154
--config, 152
--config-section, 152
--convert, 153
--format, 152
--fuzzy, 154
--input, 153
--key, 152
--method, 154
--output, 153

--overwrite, 153
--url, 152
changes, 153
cleanup, 153
commit, 152
download, 153
list-components, 152
list-languages, 152
list-projects, 152
list-translations, 152
lock, 153
lock-status, 153
ls, 152
pull, 153
push, 153
repo, 153
reset, 153
show, 152
statistics, 153
unlock, 153
upload, 153
version, 152
wlc.config
moduł, 157
wlc.main
moduł, 157
WPF
file format, 88

X

XLIFF
file format, 78
XML
file format, 91

Y

YAML
file format, 90
YAML Ain't Markup Language
file format, 90

Z

zmienna środowiskowa
CELERY_BACKUP_OPTIONS, 163, 179
CELERY_BEAT_OPTIONS, 163, 179
CELERY_MAIN_OPTIONS, 163, 179
CELERY_MEMORY_OPTIONS, 163, 179
CELERY_NOTIFY_OPTIONS, 163, 179
CELERY_TRANSLATE_OPTIONS, 163, 179
POSTGRES_ALTER_ROLE, 174
POSTGRES_CONN_MAX_AGE, 175
POSTGRES_DATABASE, 174
POSTGRES_DISABLE_SERVER_SIDE_CURSORS, 175
POSTGRES_HOST, 174
POSTGRES_PASSWORD, 174
POSTGRES_PASSWORD_FILE, 174
POSTGRES_PORT, 174
POSTGRES_SSL_MODE, 174
POSTGRES_USER, 174
REDIS_DB, 175
REDIS_HOST, 175
REDIS_PASSWORD, 175, 176
REDIS_PASSWORD_FILE, 176
REDIS_PORT, 175
REDIS_TLS, 176
REDIS_VERIFY_SSL, 176
ROLLBAR_ENVIRONMENT, 178
ROLLBAR_KEY, 178
SENTRY_DSN, 178
SENTRY_ENVIRONMENT, 178
SOCIAL_AUTH_SLACK_SECRET, 173
WEB_WORKERS, 163, 179, 180
WEBLATE_ADD_ADDONS, 179
WEBLATE_ADD_APPS, 179
WEBLATE_ADD_AUTOFIX, 179
WEBLATE_ADD_CHECK, 179
WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 167
WEBLATE_ADMIN_EMAIL, 163--165
WEBLATE_ADMIN_NAME, 163--165
WEBLATE_ADMIN_PASSWORD, 159, 163--165
WEBLATE_ADMIN_PASSWORD_FILE, 165
WEBLATE_AKISMET_API_KEY, 168, 396
WEBLATE_ALLOWED_HOSTS, 165, 205, 210, 358
WEBLATE_API_RATELIMIT_ANON, 106, 169
WEBLATE_API_RATELIMIT_USER, 106, 169
WEBLATE_AUTH_LDAP_BIND_DN, 170
WEBLATE_AUTH_LDAP_BIND_PASSWORD, 170
WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE, 170
WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS, 170
WEBLATE_AUTH_LDAP_SERVER_URI, 170
WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 170
WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 170
WEBLATE_AUTH_LDAP_USER_SEARCH, 170
WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 170
WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 170
WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 170
WEBLATE_AUTO_UPDATE, 177
WEBLATE_BASIC_LANGUAGES, 169
WEBLATE_BORG_EXTRA_ARGS, 170
WEBLATE_CONTACT_FORM, 165
WEBLATE_CSP_CONNECT_SRC, 168
WEBLATE_CSP_FONT_SRC, 169
WEBLATE_CSP_IMG_SRC, 168
WEBLATE_CSP_SCRIPT_SRC, 168
WEBLATE_CSP_STYLE_SRC, 168

WEBLATE_DATABASE_BACKUP, 175
WEBLATE_DEBUG, 164
WEBLATE_DEFAULT_ACCESS_CONTROL, 168
WEBLATE_DEFAULT_AUTO_WATCH, 169
WEBLATE_DEFAULT_COMMITER_EMAIL, 168
WEBLATE_DEFAULT_COMMITER_NAME, 168
WEBLATE_DEFAULT_FROM_EMAIL, 165
WEBLATE_DEFAULT_PULL_MESSAGE, 168
WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 168
WEBLATE_DEFAULT_SHARED_TM, 168
WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 168
WEBLATE_EMAIL_BACKEND, 177
WEBLATE_EMAIL_HOST, 176
WEBLATE_EMAIL_HOST_PASSWORD, 176, 177
WEBLATE_EMAIL_HOST_PASSWORD_FILE, 177
WEBLATE_EMAIL_HOST_USER, 176
WEBLATE_EMAIL_PORT, 176, 177
WEBLATE_EMAIL_USE_SSL, 176, 177
WEBLATE_EMAIL_USE_TLS, 176, 177
WEBLATE_ENABLE_AVATARS, 169
WEBLATE_ENABLE_HOOKS, 169
WEBLATE_ENABLE_HTTPS, 166, 238
WEBLATE_GET_HELP_URL, 178
WEBLATE_GITHUB_TOKEN, 167
WEBLATE_GITHUB_USERNAME, 167
WEBLATE_GITLAB_TOKEN, 167
WEBLATE_GITLAB_USERNAME, 167
WEBLATE_GOOGLE_ANALYTICS_ID, 167
WEBLATE_GPG_IDENTITY, 168
WEBLATE_HIDE_VERSION, 169
WEBLATE_INTERLEDGER_PAYMENT_POINTERS, 166
WEBLATE_IP_PROXY_HEADER, 166
WEBLATE_LEGAL_URL, 178
WEBLATE_LICENSE_FILTER, 169
WEBLATE_LICENSE_REQUIRED, 169
WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 169
WEBLATE_LOCALIZE_CDN_PATH, 178
WEBLATE_LOCALIZE_CDN_URL, 178
WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 167
WEBLATE_LOGLEVEL, 164
WEBLATE_LOGLEVEL_DATABASE, 164
WEBLATE_NO_EMAIL_AUTH, 174
WEBLATE_PAGURE_TOKEN, 168
WEBLATE_PAGURE_USERNAME, 167
WEBLATE_PRIVACY_URL, 178
WEBLATE_RATELIMIT_ATTEMPTS, 169, 398
WEBLATE_RATELIMIT_LOCKOUT, 169
WEBLATE_RATELIMIT_WINDOW, 169
WEBLATE_REGISTRATION_ALLOW_BACKENDS, 165
WEBLATE_REGISTRATION_OPEN, 165
WEBLATE_REMOVE_ADDONS, 179
WEBLATE_REMOVE_APPS, 179
WEBLATE_REMOVE_AUTOFIX, 179
WEBLATE_REMOVE_CHECK, 179
WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 167
WEBLATE_REQUIRE_LOGIN, 167, 357
WEBLATE_SAML_IDP_ENTITY_ID, 174
WEBLATE_SAML_IDP_IMAGE, 174
WEBLATE_SAML_IDP_TITLE, 174
WEBLATE_SAML_IDP_URL, 174
WEBLATE_SAML_IDP_X509CERT, 174
WEBLATE_SECURE_PROXY_SSL_HEADER, 166
WEBLATE_SERVER_EMAIL, 165
WEBLATE_SERVICE, 163, 180
WEBLATE_SILENCED_SYSTEM_CHECKS, 168, 235
WEBLATE_SIMPLIFY_LANGUAGES, 168
WEBLATE_SITE_DOMAIN, 164, 207, 226, 358
WEBLATE_SITE_TITLE, 164
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 172
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 172
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 173
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECR 173
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENA 173
WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 172
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY, 172
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET, 172
WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, 172
WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 172
WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 172
WEBLATE_SOCIAL_AUTH_FEDORA, 173
WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 171
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY, 171
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME, 171
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET, 171
WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 171
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID, 171
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY, 171
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET, 171

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, 172
WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 172
WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, 172
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, 172
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, 172
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS, 172
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, 172
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, 173
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM, 173
WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL, 173
WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE, 173
WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY, 173
WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY, 173
WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET, 173
WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE, 173
WEBLATE_SOCIAL_AUTH_OIDC_KEY, 173
WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT, 173
WEBLATE_SOCIAL_AUTH_OIDC_SECRET, 173
WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY, 173
WEBLATE_SOCIAL_AUTH_OPENSUSE, 173
WEBLATE_SOCIAL_AUTH_SLACK_KEY, 173
WEBLATE_SOCIAL_AUTH_UBUNTU, 173
WEBLATE_SSH_EXTRA_ARGS, 170
WEBLATE_STATUS_URL, 178
WEBLATE_TIME_ZONE, 165
WEBLATE_URL_PREFIX, 168
WEBLATE_WEBSITE_REQUIRED, 169
WEBLATE_WORKERS, 163, 179, 180
WL_BRANCH, 334
WL_COMPONENT_NAME, 334
WL_COMPONENT_SLUG, 334
WL_COMPONENT_URL, 334
WL_ENGAGE_URL, 334
WL_FILE_FORMAT, 334
WL_FILEMASK, 334
WL_LANGUAGE, 334
WL_NEW_BASE, 334
WL_PATH, 334
WL_PREVIOUS_HEAD, 334
WL_PROJECT_NAME, 334
WL_PROJECT_SLUG, 334
WL_REPO, 334
WL_TEMPLATE, 334
WL_VCS, 334