The Weblate Manual

Выпуск 4.5.2

Michal Čihař

мар. 26, 2021

Руководство пользователя

1 Руководство пользователя **1**
1.1 Основы Weblate . . . 1
1.2 Регистрация и профиль пользователя . . . 1
1.3 Перевод с помощью Weblate . . . 10
1.4 Скачивание и загрузка переводов . . . 20
1.5 Глоссарий . . . 24
1.6 Проверки и исправления . . . 26
1.7 Поиск . . . 43
1.8 Рабочие процессы перевода . . . 49
1.9 Часто задаваемые вопросы . . . 53
1.10 Поддерживаемые форматы файлов . . . 61
1.11 Интеграция с системой контроля версий . . . 80
1.12 REST API Weblate'a . . . 87
1.13 Клиент Weblate . . . 132
1.14 Python API Weblate'a . . . 137

2 Руководство администратора **140**
2.1 Инструкции по настройке . . . 140
2.2 Развёртывание Weblate . . . 200
2.3 Обновление Weblate . . . 201
2.4 Резервное копирование и перенос Weblate . . . 208
2.5 Аутентификация . . . 214
2.6 Управление доступом . . . 224
2.7 Проекты перевода . . . 234
2.8 Определение языков . . . 251
2.9 Непрерывная локализация . . . 254
2.10 Лицензирование переводов . . . 264
2.11 Процесс перевода . . . 266
2.12 Проверки и исправления . . . 272
2.13 Машинный перевод . . . 281
2.14 Надстройки . . . 288
2.15 Память переводов . . . 299
2.16 Конфигурация . . . 300
2.17 Пример файла настроек . . . 330
2.18 Команды управления . . . 346
2.19 Объявления . . . 357
2.20 Списки компонентов . . . 360
2.21 Дополнительные модули Weblate . . . 361
2.22 Настройка Weblate . . . 366
2.23 Интерфейс управления . . . 368
2.24 Получение поддержки по Weblate . . . 376

2.25 Правовые документы . . . 379

3 Руководство участника **381**
3.1 Как внести свой вклад в Weblate . . . 381
3.2 Начало участия в разработке Weblate . . . 383
3.3 Исходный код Weblate . . . 387
3.4 Отладка Weblate . . . 388
3.5 Внутреннее устройство Weblate . . . 390
3.6 Разработка надстроек . . . 391
3.7 Фронтэнд Weblate . . . 393
3.8 Сообщение о проблемах в Weblate'е . . . 394
3.9 Система автоматического тестирования и непрерывной интеграции Weblate . . . 395
3.10 Схемы данных . . . 396
3.11 Выпуск новых версий Weblate . . . 401
3.12 Безопасность и конфиденциальность . . . 402
3.13 О Weblate . . . 402
3.14 Лицензия . . . 403

4 История изменений **404**
4.1 Weblate 4.5.2 . . . 404
4.2 Weblate 4.5.1 . . . 404
4.3 Weblate 4.5 . . . 405
4.4 Weblate 4.4.2 . . . 406
4.5 Weblate 4.4.1 . . . 406
4.6 Weblate 4.4 . . . 406
4.7 Weblate 4.3.2 . . . 407
4.8 Weblate 4.3.1 . . . 407
4.9 Weblate 4.3 . . . 408
4.10 Weblate 4.2.2 . . . 409
4.11 Weblate 4.2.1 . . . 409
4.12 Weblate 4.2 . . . 409
4.13 Weblate 4.1.1 . . . 410
4.14 Weblate 4.1 . . . 410
4.15 Weblate 4.0.4 . . . 411
4.16 Weblate 4.0.3 . . . 412
4.17 Weblate 4.0.2 . . . 412
4.18 Weblate 4.0.1 . . . 413
4.19 Weblate 4.0 . . . 413
4.20 Серия Weblate 3.x . . . 414
4.21 Серия Weblate 2.x . . . 425
4.22 Серия Weblate 1.x . . . 437
4.23 Серия Weblate 0.x . . . 441

Содержание модулей Python **445**

HTTP Routing Table **446**

Алфавитный указатель **449**

ГЛАВА 1

Руководство пользователя

1.1 Основы Weblate

1.1.1 Структура проектов и компонентов

Переводы в Weblate организованы по проектам и компонентам. Каждый проект может содержать несколько компонентов, а те — переводы на отдельные языки. Компоненту соответствует один переводимый файл (например, *GNU gettext* или *Строковые ресурсы Android*). Проекты служат для организации компонентов в логические наборы (например, для группировки всех переводов, используемых в одном приложении).

Внутри каждого проекта переводы общих строк по умолчанию распространяются на другие его компоненты. Это облегчает бремя повторного и много-версионного перевода. Если в вашем проекте перевод в разных компонентов должен отличаться, то такое «распространение» можно отключить в *настройках компонента* с помощью параметра «*Разрешить распространение перевода*».

См.также:

../devel/integration

1.2 Регистрация и профиль пользователя

1.2.1 Регистрация

По умолчанию любой может просматривать проекты, смотреть или предлагать переводы. Право сохранять изменения имеют только зарегистрированные пользователи и они упоминаются как авторы каждого сделанного ими перевода.

Вы можете зарегистрироваться, выполнив несколько простых шагов:

1. Заполните свои учётные данные в форму регистрации.
2. Активируйте регистрацию, пройдя по ссылке из письма, которое придёт вам на электронную почту.
3. По желанию настройте свой профиль, выбрав, какие языки вы знаете.

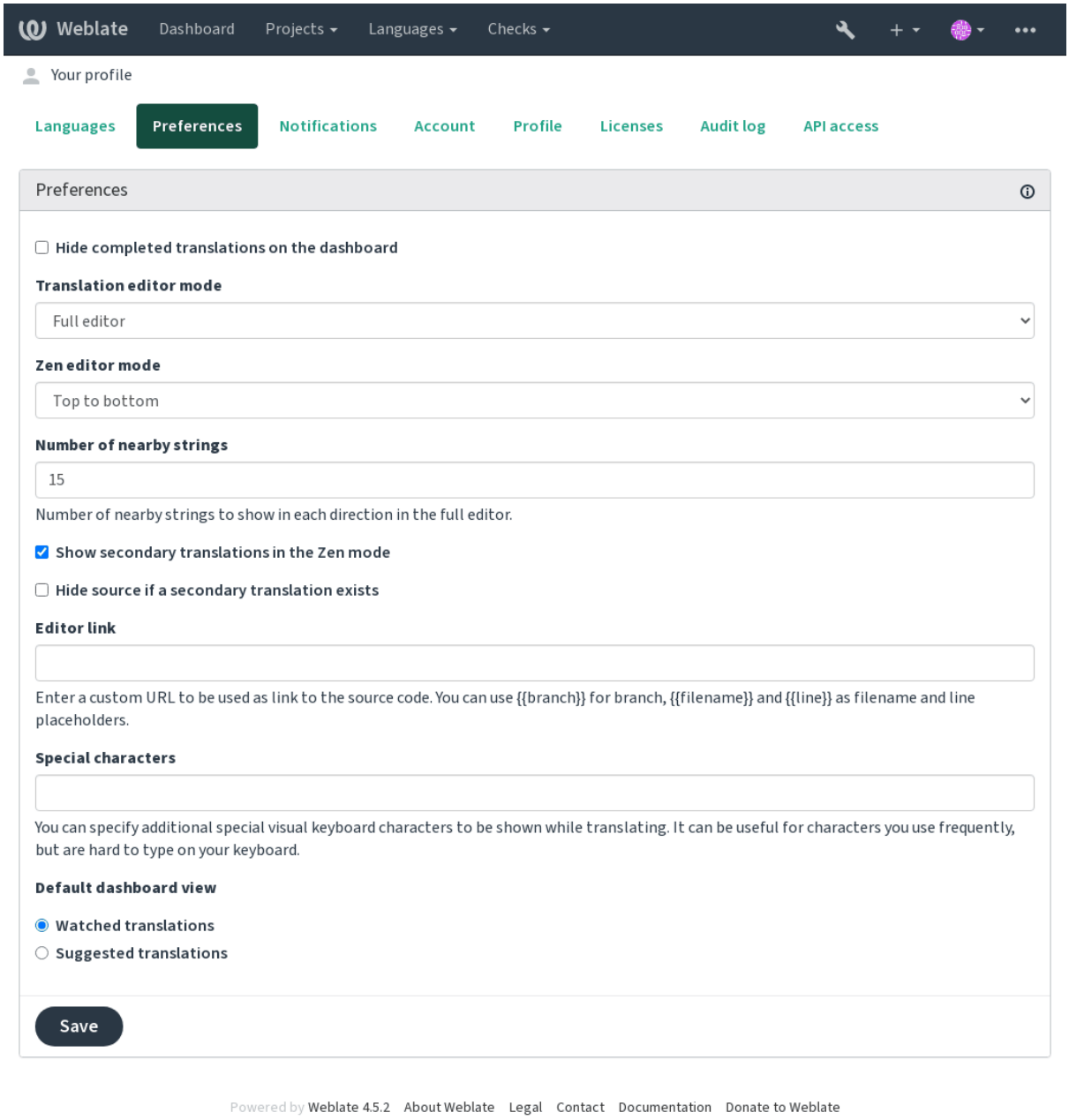
1.2.2 Панель управления

При входе в систему вы увидите сводку с проектами и их компонентами, а также информацию о прогрессе перевода по каждому из них.

Добавлено в версии 2.5.

По умолчанию показываются компоненты отслеживаемых вами проектов со ссылками на предпочитаемые вами языки.

Подсказка: При помощи вкладок навигации вы можете переключиться на другие представления.



Меню содержит следующие параметры:

- Пункт *Проекты* > *Просмотр всех проектов* в главном меню позволит перейти на страницу, где можно посмотреть статус перевода всех проектов, которые есть на данном экземпляре Weblate.

- Выбор какого-либо из ваших основных языков в главном меню *Языки* выдаст страницу, где отображается статус перевода всех проектов, на этот конкретный язык.
- На вкладке *Отслеживаемые переводы* панели управления показывается статус перевода тех проектов, которые вы отслеживаете, и только на те языки, которые вы выбрали своими основными.

Кроме того, в выпадающем списке может отображаться любое количество *списков компонентов* — наборов компонентов различных проектов, которые заранее настроил администратор Weblate; смотрите подробности в разделе *Списки компонентов*.

Вы можете изменить внешний вид панели управления под себя на вкладке *Настройки* в параметрах своего профиля.

Примечание: Если Weblate настроен для перевода одного проекта (установлен параметр `SINGLE_PROJECT` в файле `settings.py`, смотрите раздел *Конфигурация*), то вместо того, чтобы отображать панель управления, Weblate будет сразу перенаправлять пользователь на страницу этого проекта (или его компонента).

1.2.3 Профиль пользователя

Чтобы перейти к настройкам своего профиля, щелкните по иконке пользователя в верхней правой части экрана (в верхнем меню) и выберите пункт *Параметры*.

В профиле пользователя содержатся ваши предпочтения. Имя и адрес электронной почты будут использоваться в коммитах системы контроля версий, поэтому укажите в них верную информацию.

Примечание: Во всех пунктах настроек, связанных с языками, можно выбрать только те языки, которые в данный момент уже используются в переводах.

Подсказка: Чтобы сделать доступными и другие языки, на которые вы хотите переводить, запросите их добавление или добавьте их, нажав на соответствующую кнопку.

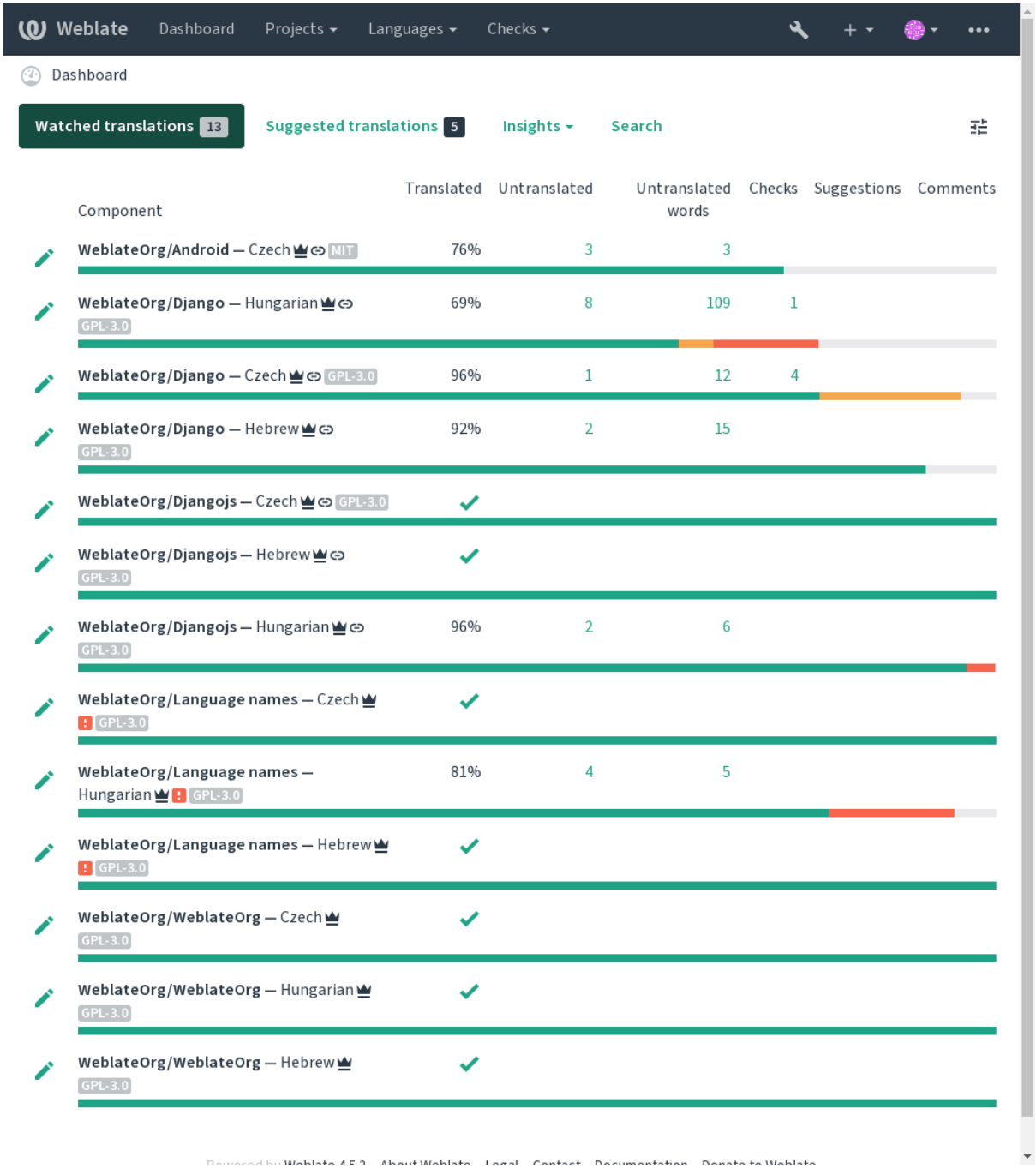
Языки

1.2.4 Interface language

Choose the language you want to display the UI in.

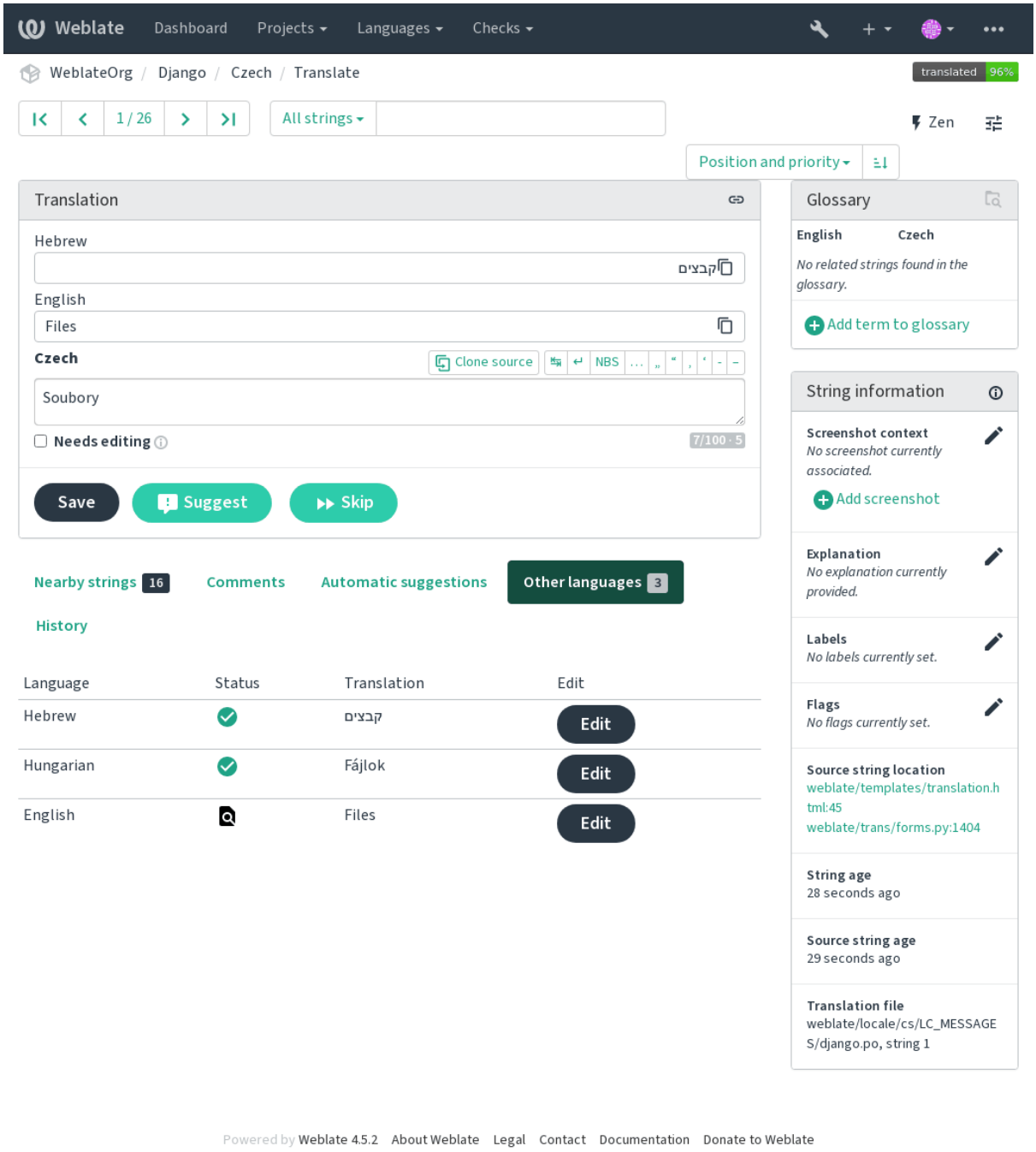
Языки перевода

Выберите, на какие языки вы предпочитаете переводить, и они будут показаны на главной странице отслеживаемых проектов, так что доступ ко всем переводам на каждый из этих языков для вас упростится.



Вспомогательные языки

Вы можете определить вспомогательные языки, которые будут вам показываться при переводе для ориентира. Пример показан на следующем рисунке, на котором вспомогательным языком выбран иврит:



1.2.5 Настройки

Вид панели управления по умолчанию

На вкладке *Настройки* можно выбрать, какой из доступных видов панели управления будет показан по умолчанию. При выборе пункта *Список компонентов* необходимо будет выбрать из раскрывающегося списка *Список компонентов по умолчанию*, какой именно список компонентов будет отображаться.

См.также:

Списки компонентов

Ссылка на редактор

Ссылка на исходный код показывается в web-браузере, настроенном по умолчанию в разделе *Настройки компонента*.

Подсказка: Установив *Ссылку на редактор*, вы будете использовать свой локальный редактор для открытия исходного кода файла из системы контроля версий с переведёнными строками. Вы можете использовать *шаблонную разметку*.

Обычно хорошим выбором является что-то вроде `editor://open/?file={{filename}}&line={{line}}`.

См.также:

Более подробную информацию о регистрации пользовательских URL-протоколов для редактора вы можете найти в документации к Nette (на английском языке).

1.2.6 Уведомления

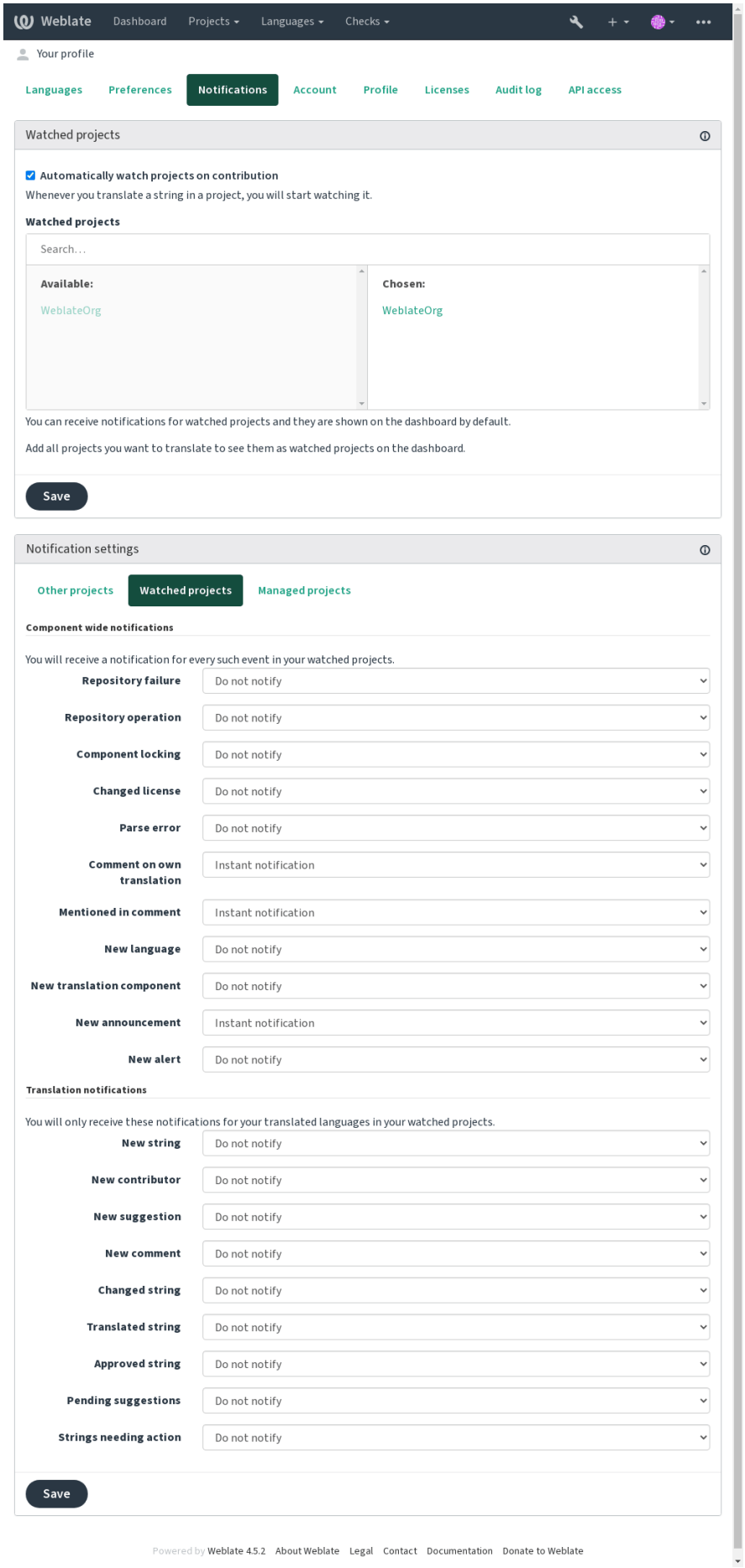
На вкладке *Уведомления* можно подписаться на различные уведомления. Уведомления по выбранным событиям для отслеживаемых или управляемых проектов будут высылаться вам по электронной почте.

Некоторые уведомления отправляются только для событий, связанных с вашим языком (например, уведомление о новых строках для перевода), в то время как другие срабатывают на уровне компонента (например, уведомление об ошибках слияния). Эти две группы уведомлений в настройках визуально разделены.

Вы можете включать или отключать уведомления для отслеживаемых и управляемых проектов, а в дальнейшем их можно настроить (или полностью отключить) для каждого отдельного проекта или компонента. Посетите страницу компонента и выберите соответствующий пункт в меню *Отслеживание*.

Если параметр *Автоматически добавлять проект в отслеживаемые при правке* включён, то при редактировании какого-либо проекта он будет автоматически добавлен в ваш список отслеживаемых. Значение по умолчанию зависит от параметра `DEFAULT_AUTO_WATCH`.

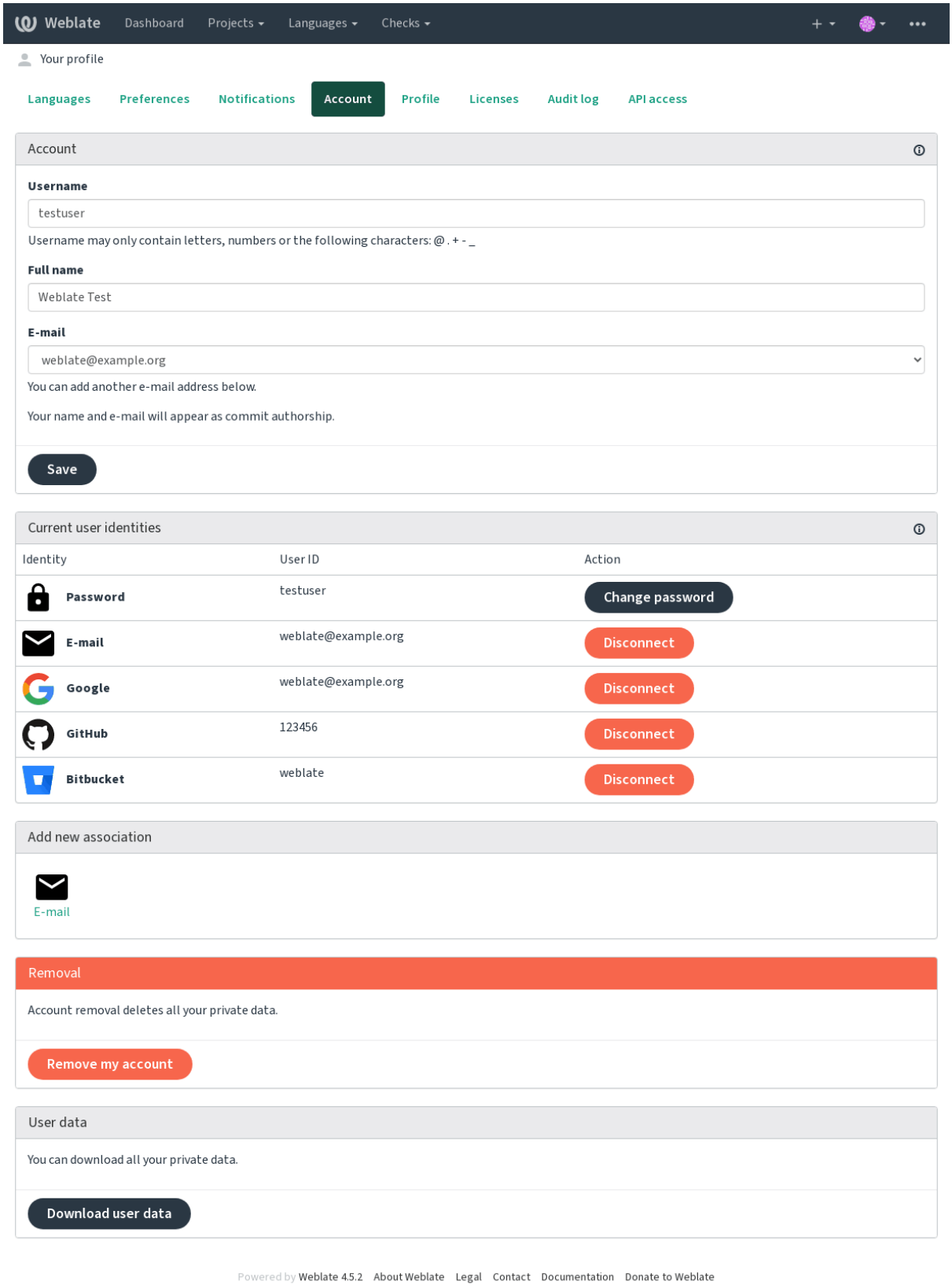
Примечание: Вы не будете получать уведомления о собственных действиях.


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
Repository operation
Component locking
Changed license
Parse error
Comment on own translation
Mentioned in comment
New language
New translation component
New announcement
New alert
Do not notify
Instant notification
Translation notifications
You will only receive these notifications for your translated languages in your watched projects.
New string
New contributor
New suggestion
New comment
Changed string
Translated string
Approved string
Pending suggestions
Strings needing action
Save
Powered by Weblate 4.5.2
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.2.7 Учётная запись

Вкладка *Учётная запись* позволяет вам настроить основные данные учётной записи, подключить различные сервисы, которые вы можете использовать для входа в Weblate, полностью удалить свою учётную запись или скачать свои данные пользователя (смотрите раздел *Файл экспорта данных пользователя Weblate'а*).

Примечание: Список сервисов зависит от ваших настроек Weblate и может включать такие популярные сайты, как GitLab, GitHub, Google, Facebook или Bitbucket, а также других провайдеров OAuth 2.0.


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 4.5.2
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.2.8 Профиль

Все поля на этой странице являются необязательными и в любое время могут быть удалены, и заполнив их, вы даёте нам согласие на показ этих данных везде, где появляется ваш профиль пользователя.

Аватар может показываться для каждого пользователя (это зависит от параметра `ENABLE_AVATARS`). Эти изображения получаются от сервиса https://gravatar.com/.

1.2.9 Лицензии

1.2.10 Доступ к API

Вы можете получить или пересоздать свой токен доступа здесь.

1.2.11 Журнал аудита

Журнал аудита отслеживает действия, произведённые с вашей учётной записи. Для каждого важного действия с вашей учётной записью в нём регистрируются IP-адрес и браузер. Критические действия вдобавок инициируют уведомление на основной адрес электронной почты.

См.также:

Работа за обратным прокси

1.3 Перевод с помощью Weblate

Благодарим вас за проявленный интерес к переводу с помощью Weblate. Проекты могут быть настроены либо на прямой перевод, либо на принятие предложений, внесённых пользователями без учётных записей.

В целом, существует два режима перевода:

- Проект принимает прямые переводы
- Проект принимает только предложения, которые автоматически подтверждаются по достижении определённого числа голосов

Для получения более подробной информации о процессе перевода, пожалуйста, ознакомьтесь с разделом *Рабочие процессы перевода*.

Варианты видимости проекта перевода:

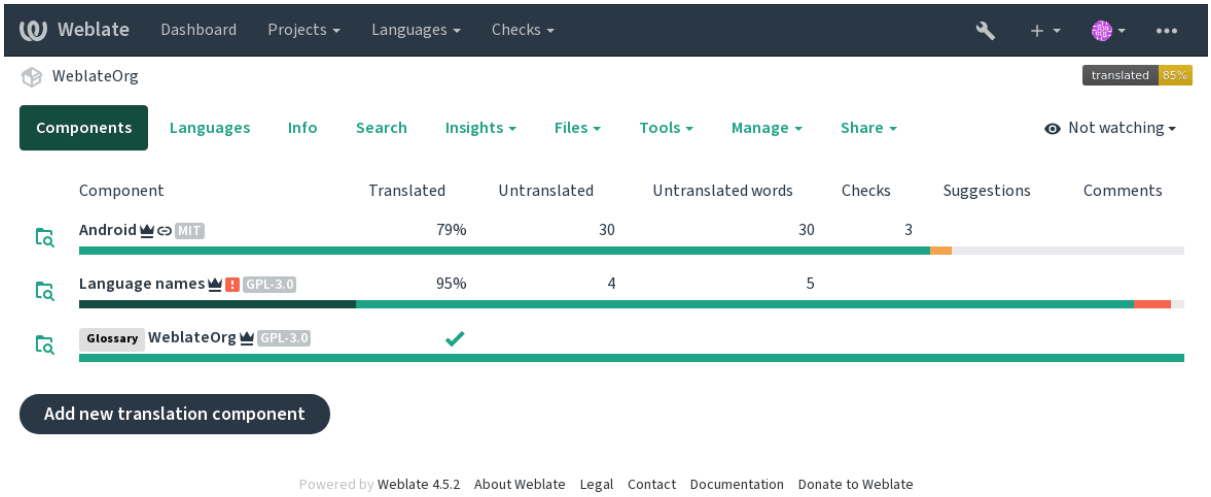
- Видимый публично, в который внести свой вклад может любой желающий
- Видимый только для определённой группы переводчиков

См.также:

Управление доступом, *Рабочие процессы перевода*

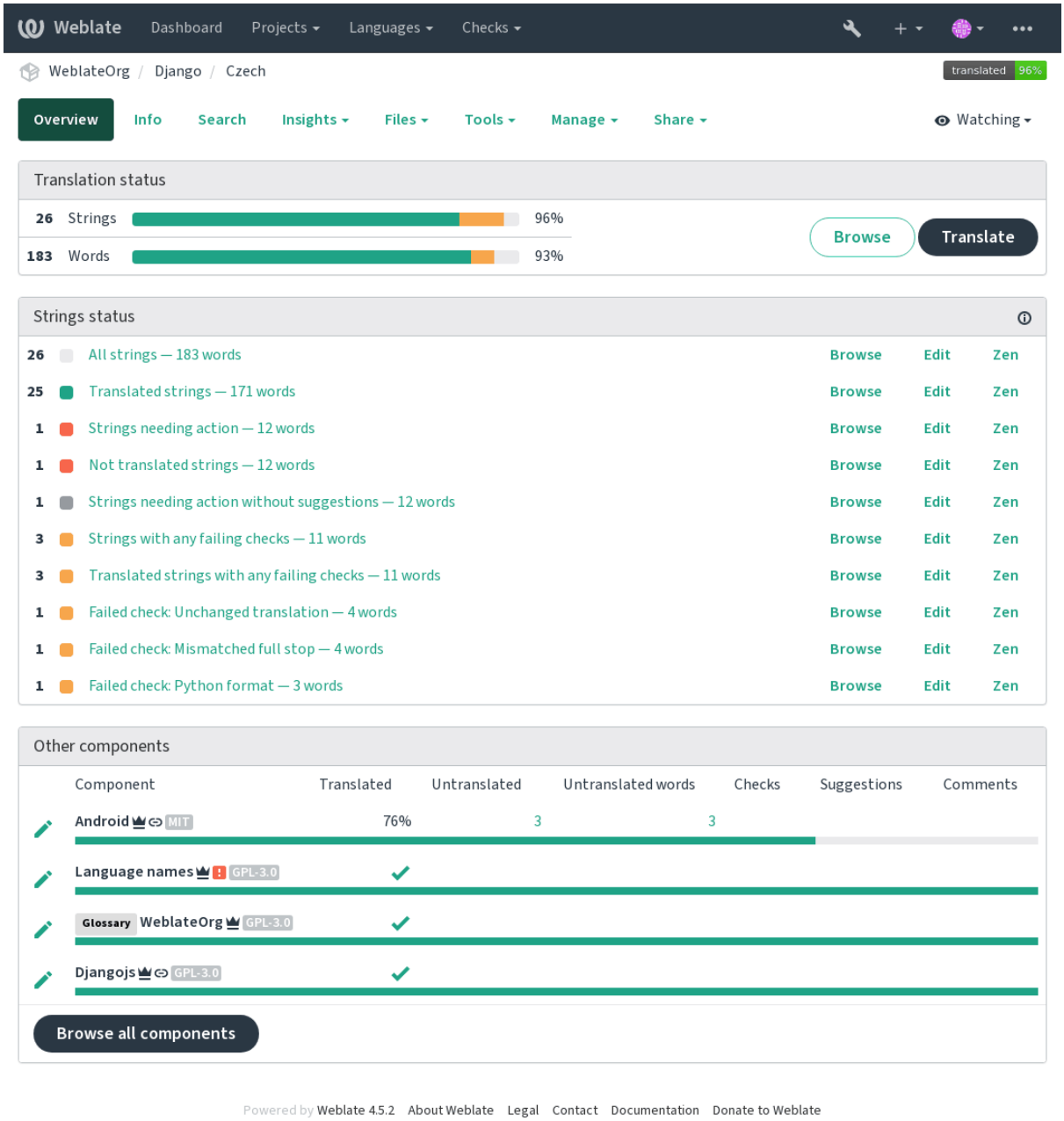
1.3.1 Проекты перевода

В проекте перевода содержатся относящиеся к нему компоненты — ресурсы связанные с одним и тем же программным обеспечением, книгой или проектом.



1.3.2 Ссылки на перевод

Когда вы зайдёте в компонент, там будет список со ссылками, которые ведут к собственно переводу. Далее перевод подразделяется на отдельные проверки, такие как *Непереведённые строки* или *Строки, требующие действий*. Если весь проект переведён и в нём не обнаружено ошибок, то в нём всё равно будет доступна ссылка *Все строки*. Также вы можете воспользоваться полем поиска для того чтобы найти определённую строку или термин.



1.3.3 Предложения

Примечание: Фактические разрешения могут отличаться в зависимости от ваших настроек Weblate.

Анонимные пользователи могут только вносить предложения (по умолчанию). Тоже самое могут делать и зарегистрированные пользователи, если они неуверенны в собственном переводе. Эти предложения в дальнейшем сможет просмотреть и отрецензировать другой переводчик.

Предложения ежедневно сканируются с целью удаления дубликатов и предложений, совпадающих с текущим переводом.

1.3.4 Комментарии

Three types of comments can be posted: for translations, source strings, or to report source string bugs when this functionality is turned on using *Включить рецензии исходных строк*. 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.

Во всех типах комментариев вы можете использовать синтаксис Markdown и упоминать других пользователей, используя синтаксис `@упоминаний`.

См.также:

report-source, *Рецензирование исходных строк*, *Включить рецензии исходных строк*

1.3.5 Варианты

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.

См.также:

variants, *Варианты*

1.3.6 Метки

Метки используются для категоризации строк в проекте для дальнейшего изменения рабочего процесса локализации (например, для определения категорий строк).

См.также:

labels

1.3.7 Перевод

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.

Все специальные пробельные символы подчёркиваются красным цветом и обозначаются серыми видимыми символами-заменителями. Более одного последовательного пробела также подчёркиваются красным цветом, чтобы предупредить переводчика о потенциальной проблеме форматирования.

На этой странице также могут быть показаны различные кусочки дополнительной информации, большая часть которой берётся из исходного кода проекта (например, контекст, комментарии разработчика или место, где используется сообщение). Если переводчик в параметрах своего профиля выбрал дополнительные языки, то над исходной строкой будет также показан перевод на каждый из этих языков (смотрите раздел *Вспомогательные языки*).

Под переводом будут показаны предложения, внесённые другими пользователями, которые вы, в свою очередь, можете принять(✓), принять с изменениями(?) или удалить(?).

Формы множественного числа

Слова, изменяющие свою форму в зависимости от того, какое число находится рядом с ними, называются в Weblate «формами множественного числа». Каждый язык имеет своё определение множественного числа. В английском языке, например, существует всего одна такая форма. Если слово стоит в единичном числе, к примеру, «автомобиль» — «car» — неявно подразумевается именно одна машина, а во множественном — «автомобили» — «cars» — подразумеваются уже две или более машины, либо же просто понятие «автомобили» как существительное. Такие языки, как, например, чешский, арабский или русский, имеют больше форм множественного числа, а правила употребления этих форм отличаются от правил английского.

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.

См.также:

Формула множественного числа

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 Clone source NBS
%(count)s slovo
15/140 · 14

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

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

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

Needs editing

Save 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
15 seconds ago

Source string age
16 seconds ago

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

Powered by Weblate 4.5.2 About Weblate Legal Contact Documentation Donate to Weblate

Горячие клавиши

Изменено в версии 2.18: Горячие клавиши в версии 2.18 были исправлены, чтобы они с меньшей вероятностью конфликтовали с клавишами по умолчанию браузера или системы.

При переводе можно использовать следующие горячие клавиши:

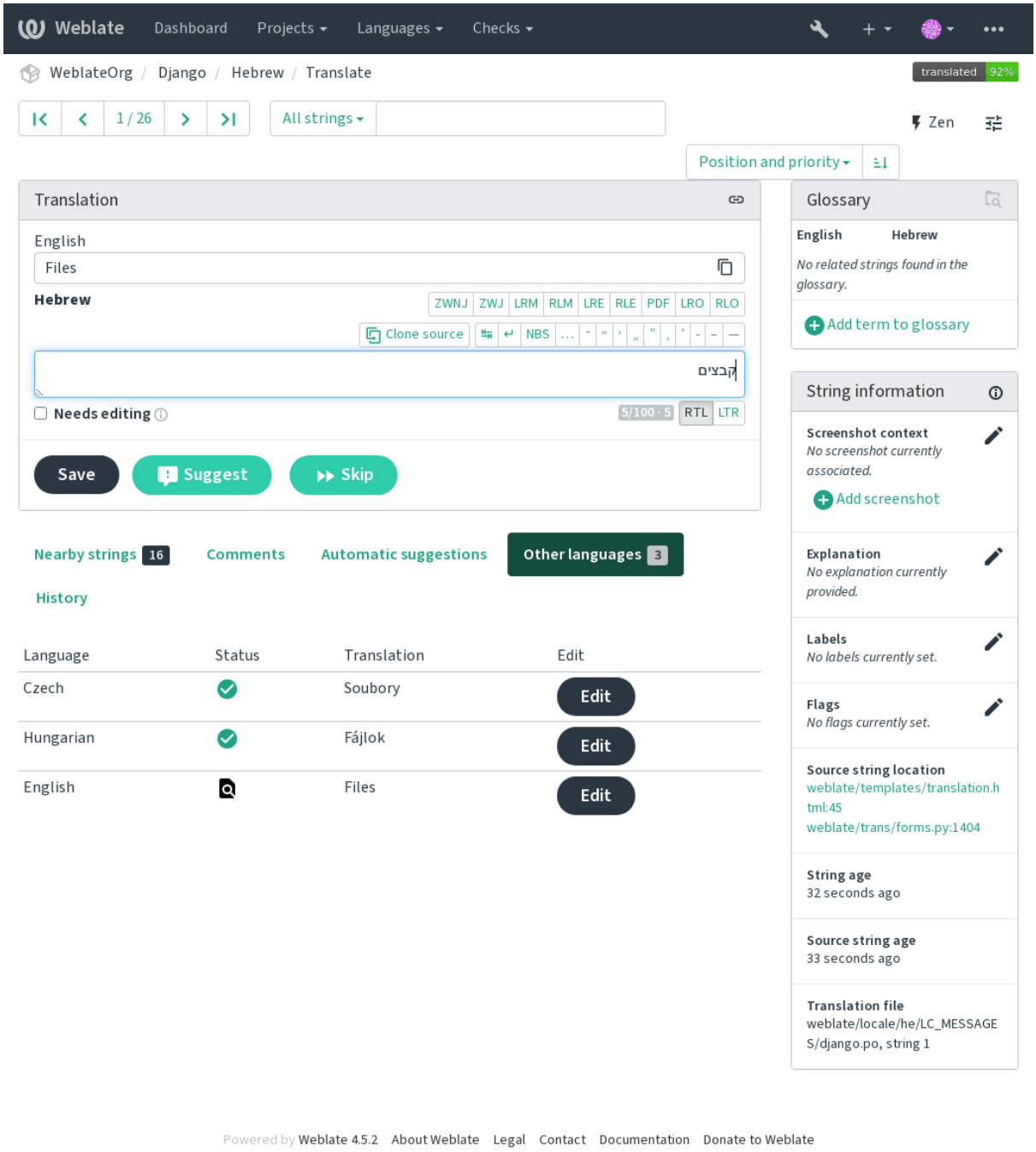
Горячие клавиши	Описание
`Alt+Home`	Переход к первому переводу в текущем поиске.
`Alt+End`	Переход к последнему переводу в текущем поиске.
`Alt+PageUp` или `Ctrl` ↑ или `Alt` ↑ или `Cmd` ↑	Переход к предыдущему переводу в текущем поиске.
`Alt+PageDown` или `Ctrl+`↓ или `Alt+`↓ или `Cmd+`↓	Переход к последующему переводу в текущем поиске.
`Alt+Enter` или `Ctrl+Enter` или `Cmd+Enter`	Сохранение текущего перевода.
`Ctrl+Shift+Enter` или `Cmd+Shift+Enter`	Unmark translation as needing edit and submit it.
`Ctrl+E` или `Cmd+E`	Фокусировка на редакторе перевода.
`Ctrl+U` или `Cmd+U`	Фокусировка на редакторе комментариев.
`Ctrl+M` или `Cmd+M`	Показ вкладки *Автоматические предложения*, см. *Автоматические предложения*.
`Ctrl+1` — `Ctrl+9` или `Cmd+1` — `Cmd+9`	Копирование из исходной строки кусочка под соответствующим номером.
`Ctrl+M+1` to `9` or `Cmd+M+1` to `9`	Копирование в текущий перевод машинного перевода с заданным номером.
`Ctrl+I+1` to `9` or `Cmd+I+1` to `9`	Игнорирование одного элемента в списке неудачных проверок.
`Ctrl+J` или `Cmd+J`	Показ вкладки *Соседние строки*.
`Ctrl+S` или `Cmd+S`	Focus search field.
`Ctrl+O` или `Cmd+O`	Copy source string.
`Ctrl+Y` или `Cmd+Y`	Toggle the *Needs editing* flag.

Визуальная клавиатура

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.

Показываемые символы разделены на три категории:

- Пользовательские символы, определённые в его *профиле*
- Предоставляемые Weblate символы для конкретного языка (например, кавычки или символы, специфичные для текста справа-налево)
- Символы, настроенные через параметр `SPECIAL_CHARS`



Контекст перевода

This contextual description provides related info about the current string.

Атрибуты строки Такие вещи, как идентификатор сообщения, его контекст (`msgctxt`) или местоположение в исходном коде.

Снимки экрана В Weblate могут быть загружены снимки экрана, чтобы лучше информировать переводчиков о том, где и как используется строка, смотрите раздел *Визуальный контекст для строк*.

Соседние строки Отображает соседние сообщения из файла перевода. Обычно они также используются в аналогичном контексте и оказываются полезными для поддержания согласованности перевода.

Другие вхождения В случае, если сообщение появляется в нескольких местах (например, в нескольких компонентах), эта вкладка показывает их все, если в них найдены противоречия (смотрите раздел *Про-*

тиворечия). Вы можете выбрать, какую из строк использовать.

Память переводов Показывает похожие строки, переведённые в прошлом, смотрите раздел *Память переводов*.

Глоссарий Отображает термины из глоссария проекта, используемые в текущем сообщении.

Недавние изменения Список людей, которые недавно изменили это сообщение, используя Weblate.

Проект Информация о проекте, например, инструкции для переводчиков, или каталог, или ссылка на строку в репозитории системе контроля версий, используемой проектом.

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

История переводов

Каждое изменение по умолчанию (если это не отключено в настройках компонента) сохраняется в базе данных и может быть отменено. Также при необходимости через нижележащую систему управления версиями можно отменить всё, что угодно.

Длина переведённой строки

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 *Двуязычные и одноязычные форматы* for more info.
- Максимальная длина в символах, определяемая в файле перевода или флагом, смотрите раздел *Максимальная длина перевода*.
- Максимальный отрисовываемый размер текста в пикселях, определяемый флагами, смотрите раздел *Максимальный размер перевода*.

1.3.8 Автоматические предложения

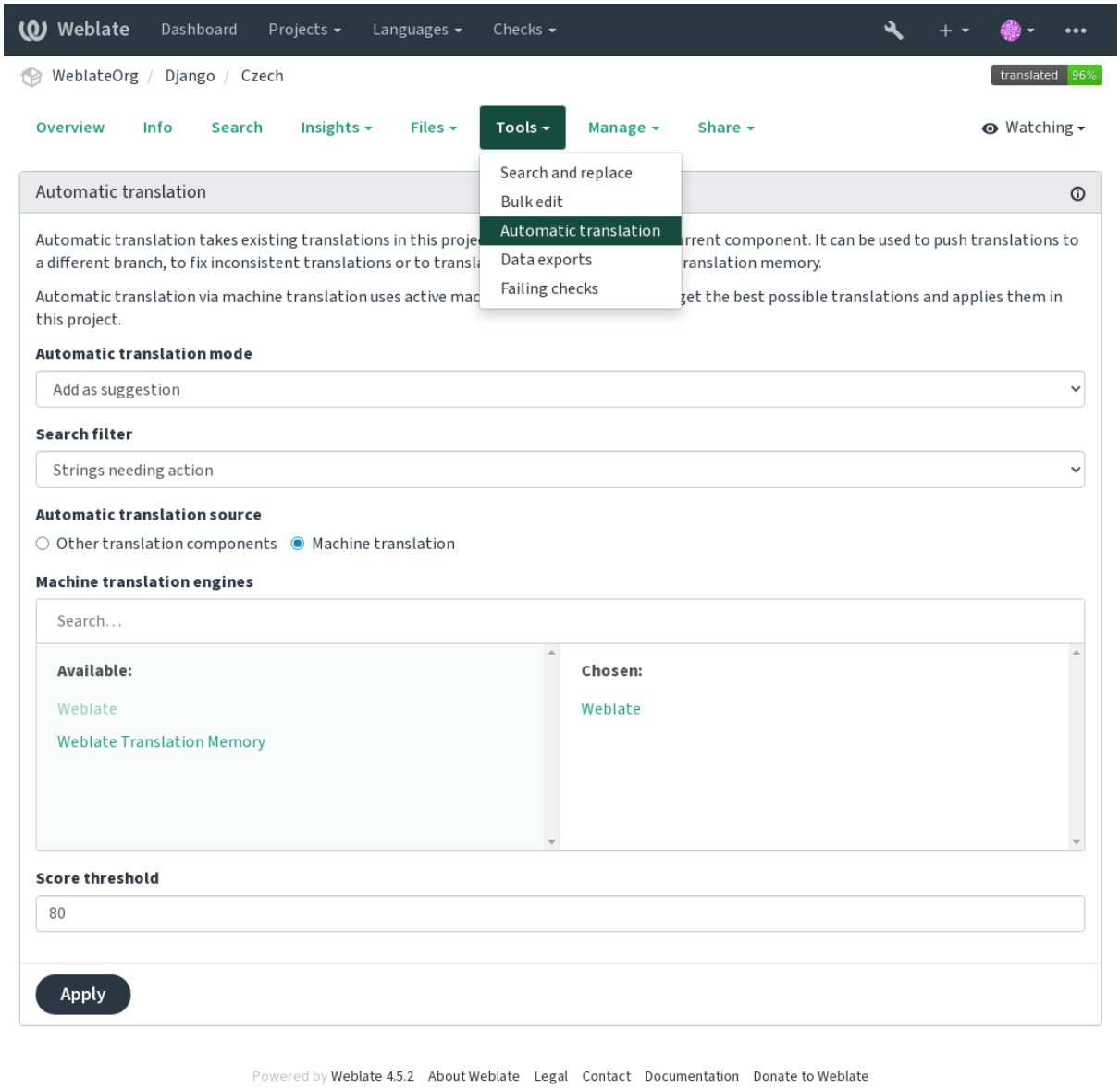
Based on configuration and your translated language, Weblate provides suggestions from several machine translation tools and *Память переводов*. All machine translations are available in a single tab of each translation page.

См.также:

Список поддерживаемых инструментов можно найти в разделе *Машинный перевод*.

1.3.9 Автоматический перевод

Для автоматического начального перевода вашего проекта на основе внешних источников вы можете использовать *Автоматический перевод*, который доступен в меню *Инструменты* для определённой пары компонента и языка:



Возможны два режима работы этого инструмента:

- Использование в качестве источника для переводов других компонентов Weblate.
- Использование выбранных сервисов машинного перевода с переводами выше определённого порога качества.

Вы также можете выбрать, какие строки должны быть переведены автоматически.

> **Предупреждение:** Имейте в виду, что если использовать широкие фильтры, вроде фильтра *Все строки*, это приведёт к перезаписи существующих переводов.

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).

См.также:

Поддержание единого перевода в разных компонентах

1.3.10 Ограничение частоты запросов

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 *Ограничение частоты запросов*.

1.3.11 Поиск и замена

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

Подсказка: Don't worry about messing up the strings. This is a two-step process showing a preview of edited strings before the actual change is confirmed.

1.3.12 Массовая правка

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).
- Корректировка флагов перевода (смотрите раздел *Настройка поведения с помощью флагов*)
- Корректировка меток строк (смотрите раздел labels)

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

См.также:

Надстройка «Массовая правка»

1.4 Скачивание и загрузка переводов

Вы можете экспортировать файлы из перевода, внести в них изменения и снова их импортировать. Это позволяет поработать над переводом в автономном режиме, а затем влить изменения обратно в существующий перевод. Это работает, даже если существующий перевод за это время был изменён.

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

1.4.1 Скачивание переводов

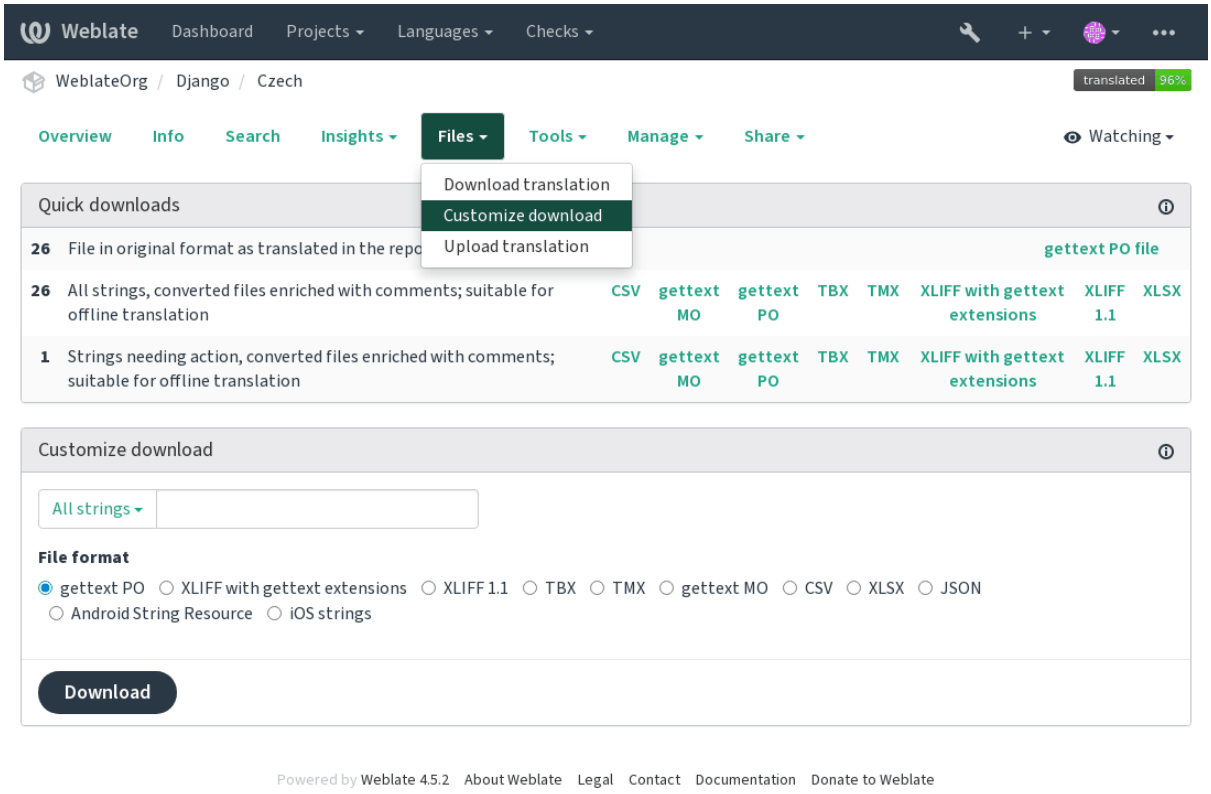
From the project or component dashboard, translatable files can be downloaded in the *Files* menu.

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 с расширениями gettext

- XLIFF 1.1
- TermBase eXchange
- Translation Memory eXchange
- gettext MO
- CSV
- Excel Open XML
- JSON
- Строковый ресурс Android
- Строки iOS

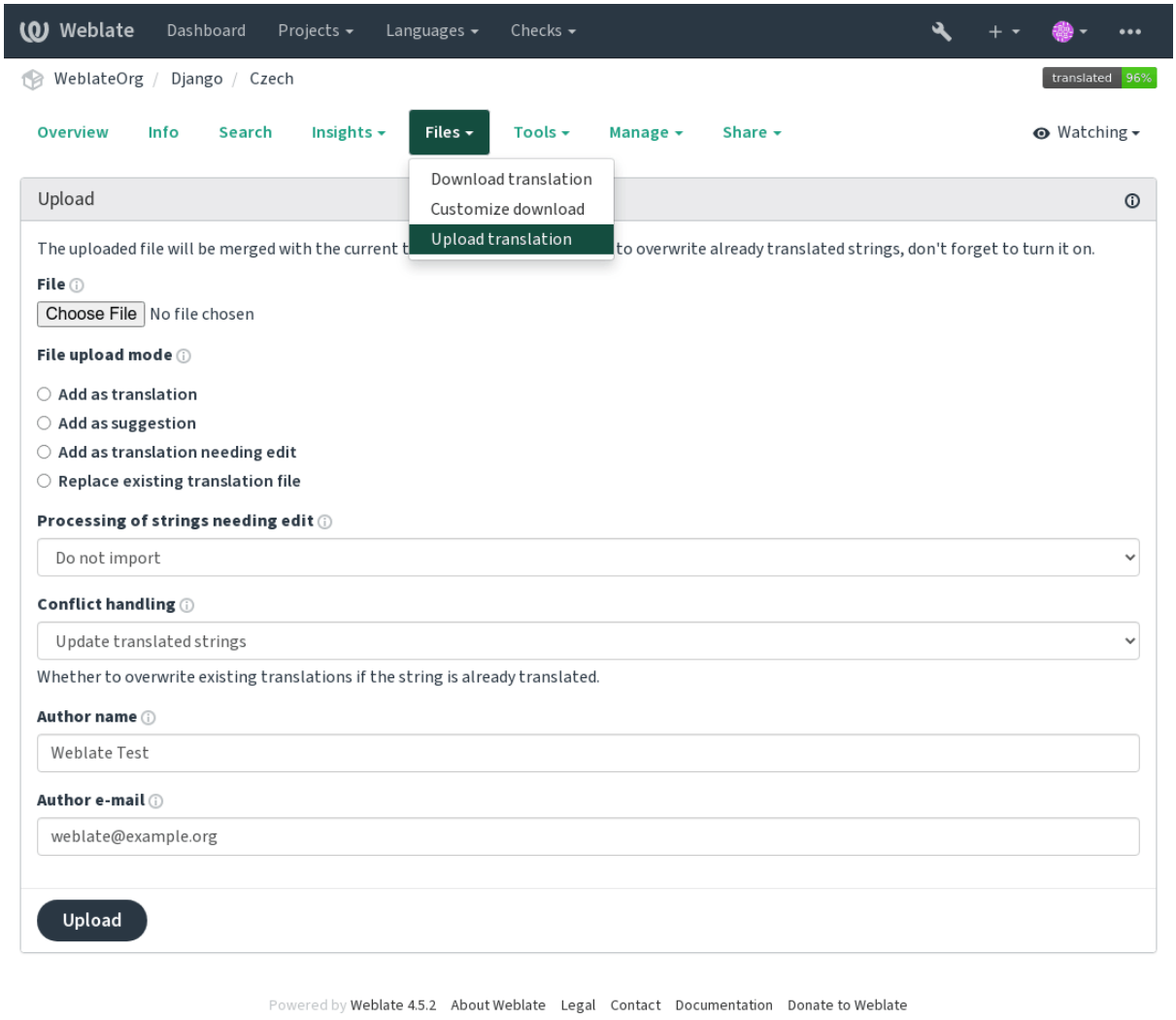


См.также:

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

1.4.2 Загрузка переводов

После того, как вы внесли в перевод свои изменения, воспользуйтесь пунктом *Загрузить перевод* меню *Файлы*.



Поддерживаемые форматы файлов

Можно загрузить любой файл в поддерживаемом формате, но всё же рекомендуется использовать тот же самый формат файла, который используется для перевода, в противном случае некоторые возможности могут быть перенесены неправильно.

См.также:

Поддерживаемые форматы файлов

Загруженный файл объединяется с существующим для обновления перевода, по умолчанию перезаписывая существующие записи (это можно отключить или включить в диалоге загрузки).

Способы импорта

При загрузке файлов перевода вы можете выбирать из следующих вариантов:

Добавить как перевод (`translate`) Импортированные переводы добавляются как переводы. Это наиболее распространённое использование и поведение по умолчанию.

Добавить как предложение (`suggest`) Импортированные переводы добавляются как предложения, делайте это тогда, когда хотите, чтобы загруженные строки были отрецензированы.

Добавить перевод как «На правку» (`fuzzy`) Импортированные переводы добавляются как переводы, требующие правки. Это может быть полезно, когда вы хотите, чтобы переводы не только использовались, но также были и отрецензированы.

Заменить существующий файл перевода (`replace`) Существующий файл заменяется новым содержимым. Это может привести к потере существующих переводов, используйте этот вариант с осторожностью.

Обновить исходные строки (`source`) Обновляет исходные строки в двуязычном файле перевода. Это поведение похоже на то, что делает надстройка *Обновлять PO-файлы до соответствия POT (msgmerge)*.

Этот вариант поддерживается только для некоторых форматов файлов.

Добавить новые строки (`add`) Добавляет новые строки для перевода, пропуская те, что уже существуют.

Если вы хотите и добавить новые строки, и обновить существующие переводы, то загрузите файл второй раз в режиме *Добавить как перевод*.

Этот вариант доступен только если включено *управление строками*.

См.также:

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

Разрешение конфликтов

Определяет, что делать с загружаемыми строками, которые уже переведены.

Строки, отмеченные как «На правку»

Также существует параметр, контролирующий то, что делать со строками в импортируемом файле, нуждающимися в редактировании . Такие строки можно обработать одним из трёх следующих способов: «Не импортировать», «Импортировать строки, как отмеченные „На правку“» или «Импортировать как переведённое».

Переопределение авторства

С разрешениями администратора вы также можете указать авторство загружаемого файла. Это может быть полезно в случае, если вы получили файл другим способом и хотите объединить переводы из него с существующими с правильным указанием фактического автора.

1.5 Глоссарий

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.

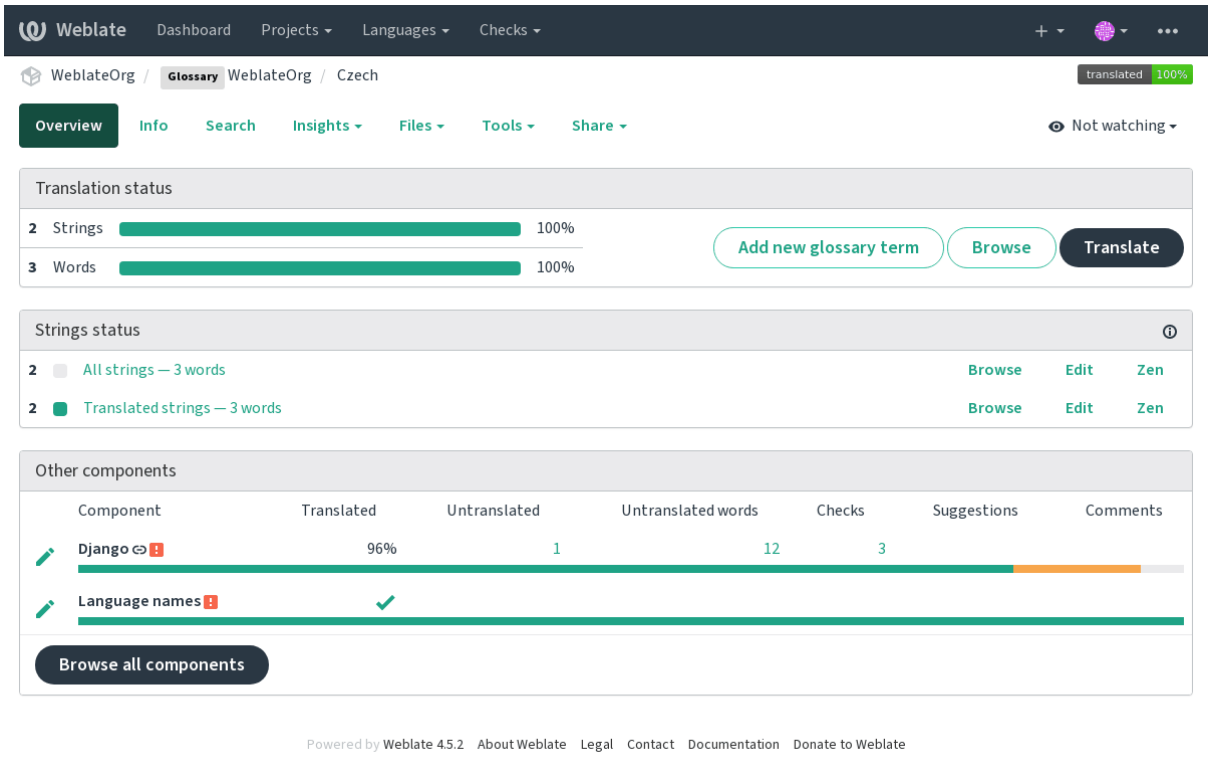
1.5.1 Управление глоссариями

Изменено в версии 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.

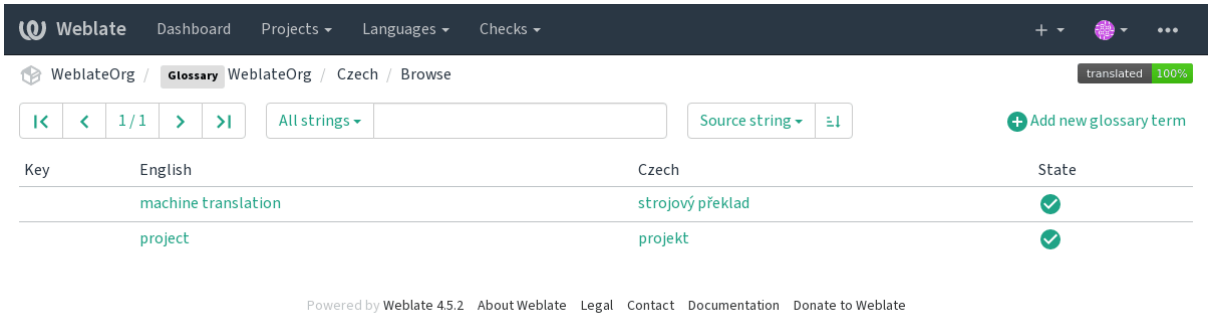
Use any component as a glossary by turning on *Использовать в качестве глоссария*. You can create multiple glossaries for one project.

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 *Разделяемые компоненты* from the respective glossary component.

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



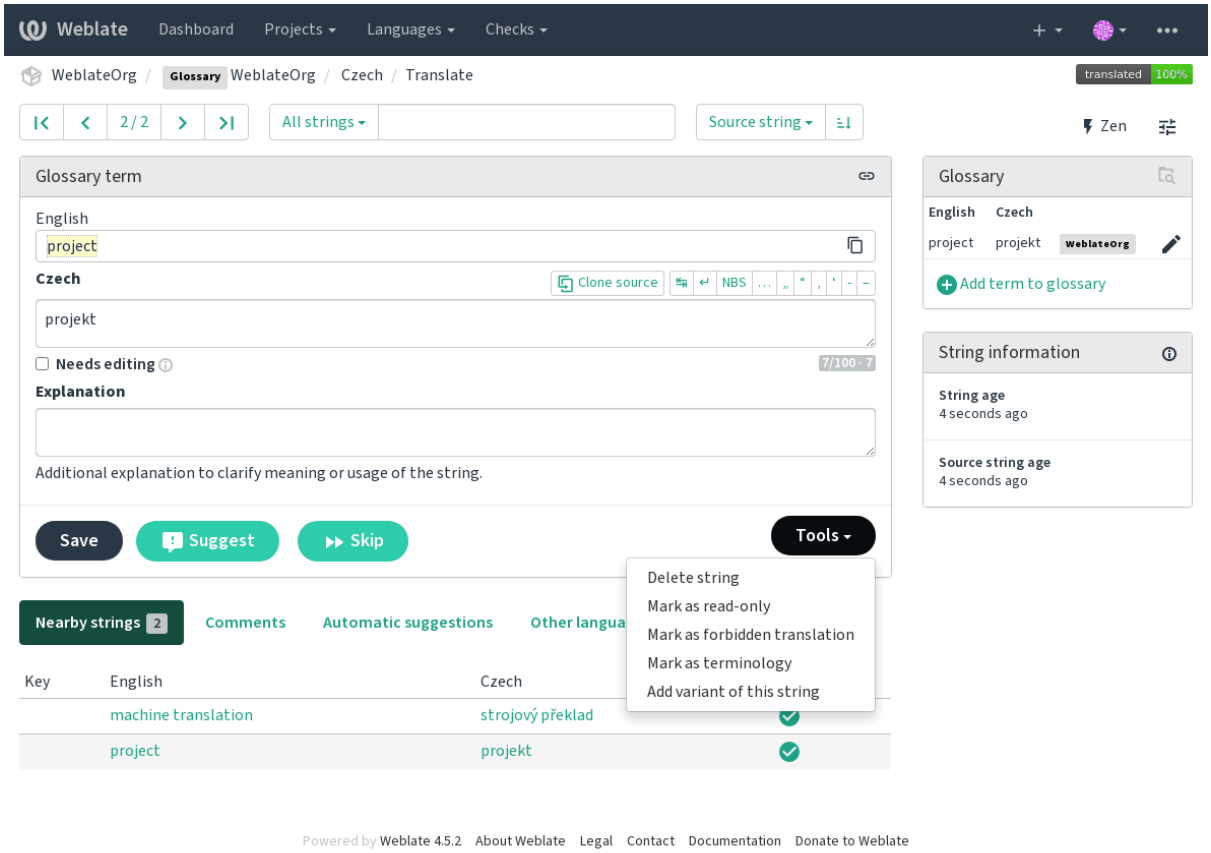
Также вы можете воспользоваться навигацией по всем терминам глоссария:



or edit them as any translations.

1.5.2 Термины глоссария

Процесс перевода терминов, добавленных в глоссарий, точно такой же, как и у обычных строк. Такжы вы можете производить некоторые дополнителльные действия с отдельными терминами с помощью кнопки с меню *Инструменты*.



Непереводимые термины

Добавлено в версии 4.5.

Flagging certain glossary term translations `read-only` by bulk-editing, typing in the flag, or by using *Tools* ↓:guilabel:*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.

См.также:

Настройка поведения с помощью флагов

Нежелательные переводы

Добавлено в версии 4.5.

Flagging certain glossary term translations as `forbidden`, by bulk-editing, typing in the flag, or by using *Tools* ↓:guilabel:*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.

См.также:

Настройка поведения с помощью флагов

Терминология

Добавлено в версии 4.5.

Flagging certain glossary terms as `terminology` by bulk-editing, typing in the flag, or or by using *Tools* ↓:guilabel:*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.

См.также:

Настройка поведения с помощью флагов

Варианты

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

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

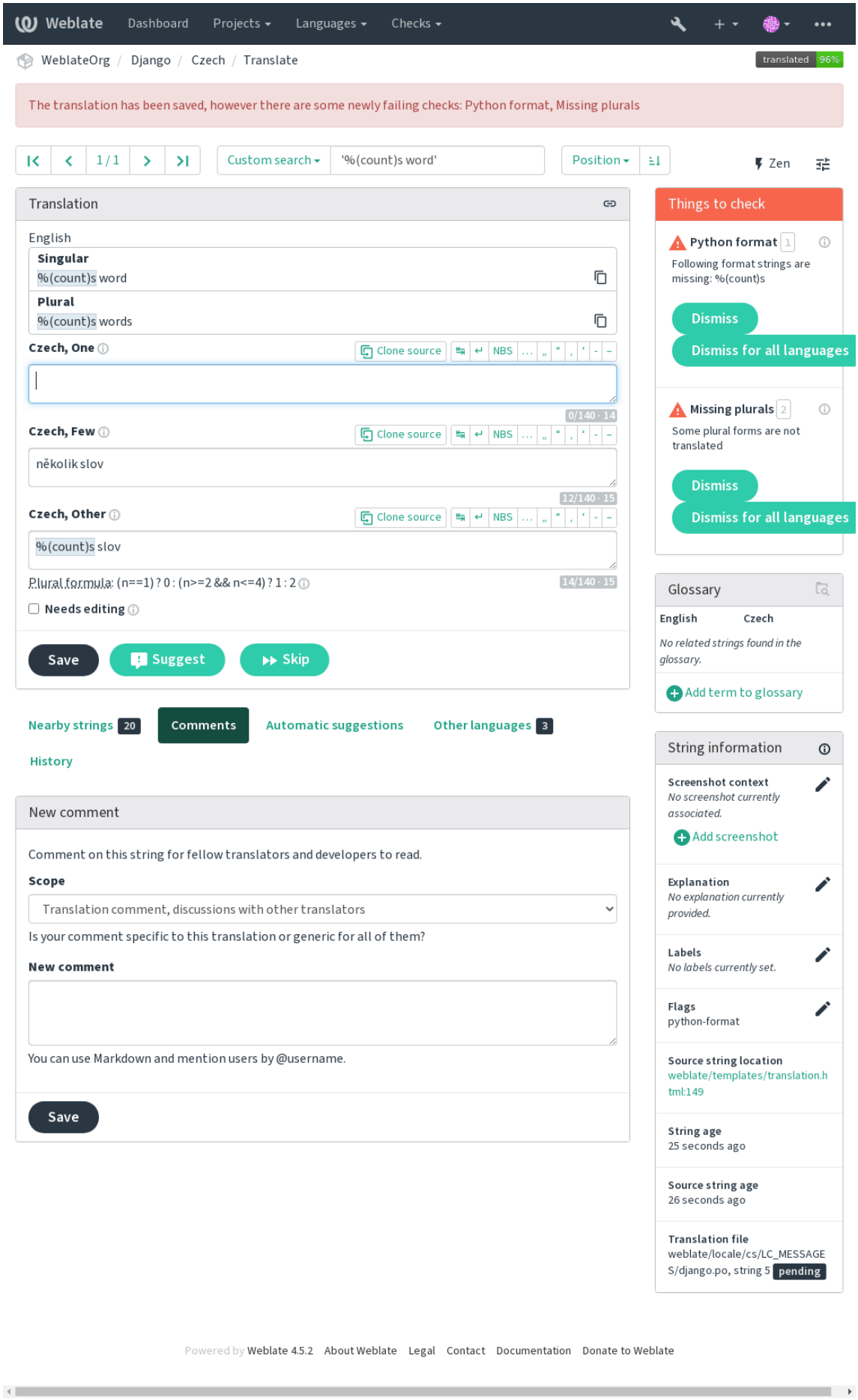
См.также:

variants

1.6 Проверки и исправления

Проверки качества помогают выявить распространённые ошибки переводчиков, что гарантирует хорошее состояние перевода. В случае ложно-положительных срабатываний проверки могут быть проигнорированы.

После отправки перевода с неудачной проверкой, она сразу же показывается пользователю:


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
Clone source
NBS
0/140 · 14
Czech, Few
několik slov
12/140 · 15
Czech, Other
%(count)s slov
14/140 · 15
Plural formula: (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2
Needs editing
Save
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
Dismiss for all languages
Missing plurals 2
Some plural forms are not translated
Dismiss
Dismiss for all languages
Glossary
English
Czech
No related strings found in the glossary.
Add term to glossary
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
25 seconds ago
Source string age
26 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5
pending
Powered by Weblate 4.5.2
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.6.1 Автоматические исправления

В дополнение к *проверкам качества*, Weblate может автоматически исправлять некоторые распространённые ошибки в переведённых строках. Используйте эту возможность с осторожностью, чтобы не допустить добавления новых ошибок.

См.также:

`AUTOFIX_LIST`

1.6.2 Проверки качества

Weblate использует широкий спектр проверок качества строк. В следующем разделе они все описываются более подробно. Также существуют проверки, специфичные для конкретных языков. Если какая-то проверка срабатывает ошибочно, пожалуйста, отправьте сообщение об ошибке.

См.также:

`CHECK_LIST`, *Настройка поведения с помощью флагов*

1.6.3 Проверки перевода

Выполняется при каждом изменении перевода, помогая переводчикам поддерживать высокое качество перевода.

BBcode-разметка

BBcode в переводе не соответствует источнику

BBCode представляет собой простую разметку, полезную для, например, выделения важных частей сообщения жирным шрифтом или курсивом.

Эта проверка гарантирует, что они также будут присутствовать в переводе.

Примечание: На данный момент метод обнаружения тегов BBcode достаточно прост, поэтому данная проверка может привести к ложным срабатываниям.

Повторяющиеся слова

Одно и то же слово повторяется в тексте дважды подряд:

Добавлено в версии 4.1.

Проверяет, что в переводе не встречаются последовательные дубликаты слова. Обычно это указывает на ошибку в переводе.

Подсказка: Эта проверка включает в себя правила, специфичные для конкретных языков, чтобы избежать ложных срабатываний. В случае ложного срабатывания этой проверки в вашем случае, сообщите нам об этом. Смотрите раздел *Сообщение о проблемах в Weblate'е*.

Не соответствует глоссарию

Добавлено в версии 4.5.

Этот перевод не содержит термин, указанный в глоссарии.

Эту проверку можно включить с помощью флага `check-glossary` (см. *Настройка поведения с помощью флагов*). Но прежде чем делать это учтите следующее:

- Эта проверка использует строгое сравнение строк, ожидается, что глоссарий будет содержать термины со всеми их вариантами.
- Проверка каждой строки на соответствие словарю — дорогое удовольствие: это значительно замедлит все операции Weblate, которые подразумевают запуск проверок, включая импорт строк или переводов.

См.также:

Глоссарий, *Настройка поведения с помощью флагов*, *Флаги перевода*

Двойной пробел

Перевод содержит двойной пробел

Проверяет наличие в переводе двойного пробела, чтобы избежать ложных срабатываний на других проверках, связанных с пробелами.

Проверка возвращает ложь, если в исходной строке найден двойной пробел, что означает, что двойной пробел был внесён в неё преднамеренно.

Форматированные строки

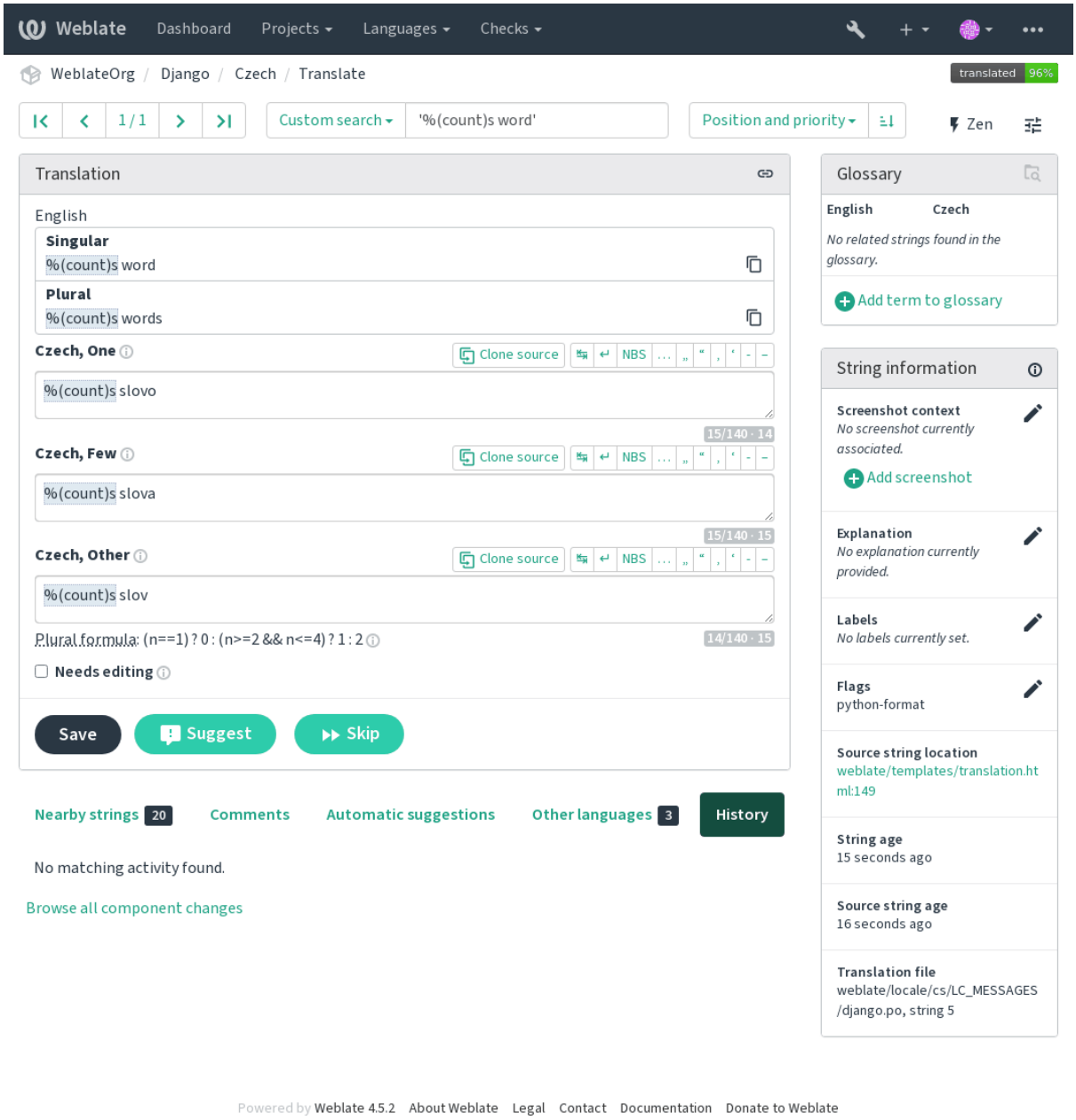
Проверяет, что форматирование в исходной строке и в переводе одинаково. Упущение в переводе строк формата обычно вызывает серьёзные проблемы, поэтому форматирование в переводе обычно должно соответствовать форматированию исходной строки.

Weblate поддерживает проверку строк формата на нескольких языках. Проверка не включается автоматически, только если строка соответствующим образом помечена (например, флагом *c-format* для C-формата). Gettext добавляет эту разметку автоматически, но для других форматов файлов или если ваши PO-файлы не сгенерированы программой **`xgettext`** вам, вероятно, придётся добавить её вручную.

Её можно добавлять как индивидуально в каждую единицу перевода (смотрите раздел *Дополнительная информация об исходных строках*), так и в *конфигурации компонента*. Определение формата для компонента целиком проще, но может привести к ложным срабатываниям, если строка не интерпретируется как строка формата, но при этом внутри неё используется синтаксис строки формата.

Подсказка: В случае, если конкретная проверка формата в Weblate отсутствует, вы можете использовать общую проверку *Символы-заполнители*.

Помимо проверки, эти флаги также включают подсветку строк формата, чтобы их было легко вставлять в перевод:



Интерполяционная строка AngularJS

Интерполяционные строки AngularJS не сооответствуют источнику

Именованная строка формата	`Your balance is {{amount}} {{ currency }}`
Флаг для включения	*angularjs-format*

См.также:

Интерполяция текста AngularJS

C-формат

Строка в формате C не соответствует источнику

Простая строка формата	`There are %d apples`
Позиционная строка формата	`Your balance is %1$d %2$s`
Флаг для включения	*c-format*

См.также:

Строки формата C, Синтаксис строки форматирования функции printf C

C#-формат

Строка в формате C# не соответствует источнику

Позиционная строка формата	`There are {0} apples`
Флаг для включения	*c-sharp-format*

См.также:

Строки формата C#

Шаблонные литералы ECMAScript

Шаблонные литералы ECMAScript не соответствует источнику

Интерполяция	`There are ${number} apples`
Флаг для включения	*es-format*

См.также:

Шаблонные литералы

Интерполяция i18next

Интерполяция i18next не соответствует источнику

Добавлено в версии 4.0.

Интерполяция	`There are {{number}} apples`
Вложение	`There are $t(number) apples`
Флаг для включения	*i18next-interpolation*

См.также:

Интерполяция i18next

Java-формат

Строка в формате Java не соответствует источнику

Простая строка формата	`There are %d apples`
Позиционная строка формата	`Your balance is %1$d %2$s`
Флаг для включения	*java-format*

См.также:

Строки в формате Java

Java MessageFormat

Строка в формате Java MessageFormat не соответствует источнику

Позиционная строка формата	`There are {0} apples`
Флаг для включения	Флаг *java-messageformat* включает эту проверку безусловно
	Флаг *auto-java-messageformat* включает проверку только если в исходной строке содержится строка формата

См.также:

Java MessageFormat

JavaScript-формат

Строка в формате JavaScript не соответствует источнику

Простая строка формата	`There are %d apples`
Флаг для включения	*javascript-format*

См.также:

Строки форматирования JavaScript

Lua-формат

Строка в формате Lua не соответствует источнику

Простая строка формата	`There are %d apples`
Флаг для включения	*lua-format*

См.также:

Строки форматирования Lua

Символы-заполнители в процентной нотации

Символы-заполнители в процентной нотации не соответствуют источнику

Добавлено в версии 4.0.

Простая строка формата	`There are %number% apples`
Флаг для включения	*percent-placeholders*

Perl-формат

Строка в формате Perl'a не соответствует источнику

Простая строка формата	`There are %d apples`
Позиционная строка формата	`Your balance is %1$d %2$s`
Флаг для включения	*perl-format*

См.также:

Функция sprintf Perl'a, Строки форматирования Perl'a

PHP- формат

Строка в формате PHP не соответствует источнику

Простая строка формата	`There are %d apples`
Позиционная строка формата	`Your balance is %1$d %2$s`
Флаг для включения	*php-format*

См.также:

Документация по функции sprintf PHP, Строки формата PHP

Python-формат: фигурные скобки

Форматирование фигурных скобок в Python не соответствует источнику

Простая строка формата	`There are {} apples`
Именованная строка формата	`Your balance is {amount} {currency}`
Флаг для включения	*python-brace-format*

См.также:

Формат фигурных скобок в Python'e, строки формата Python

Python-формат

Строка в формате Python'а не соответствует источнику

Простая строка формата	`There are %d apples`
Именованная строка формата	`Your balance is %(amount) %(currency)`
Флаг для включения	*python-format*

См.также:

Форматирование строк в Python'е, Python'ьи строки формата

Qt-формат

Строка в формате Qt не соответствует источнику

Позиционная строка формата	`There are %1 apples`
Флаг для включения	*qt-format*

См.также:

Qt QString::arg()

Формат множественного числа Qt

Строка в формате множественного числа Qt не соответствует источнику

Строка в формате множественного числа	`There are %Ln apple(s)`
Флаг для включения	*qt-plural-format*

См.также:

Руководство Qt по интернационализации

Ruby-формат

Строка формата Ruby не соответствует источнику

Простая строка формата	`There are %d apples`
Позиционная строка формата	`Your balance is %1$f %2$s`
Именованная строка формата	`Your balance is %+.2<amount>f %<currency>s`
Именованная шаблонная строка	`Your balance is %{amount} %{currency}`
Флаг для включения	*ruby-format*

См.также:

Ruby Kernel#sprintf

Scheme format

Scheme format string does not match source

Простая строка формата	`There are ~d apples`
Флаг для включения	*scheme-format*

См.также:

Srfi 28, Chicken Scheme format, Guile Scheme formatted output

Форматирование Vue I18n

Форматирование Vue I18n не соответствует источнику

Именованная строка формата	`There are {count} apples`
Форматирование локализации Rails	`There are %{count} apples`
Связанные сообщения локализации	`@:message.dio @:message.the_world!`
Флаг для включения	*vue-format*

См.также:

Формат сообщений локализации Vue I18n, **`Связанные сообщения локализации Vue I18n<https://kazupon.github.io/vue-i18n/ru/guide/messages.html#%D1%81%D0%B2%D1%8F%D0%B7%D0%B0%D0%%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F-n%D0%BE%D0%BA%D0%B0n%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8>`_**

Уже переведено

Эта строка уже переводилась

Означает, что строка уже переведена. Это может произойти, если переводы откатывались в системе контроля версий или иным образом терялись.

Противоречия

Эта строка имеет более одного перевода в этом проекте, либо же в некоторых компонентах она не переведена.

Weblate проверяет переводы одинаковых строк по всем переводам в проекте, чтобы помочь вам поддерживать переводы согласованными.

Проверка срабатывает, когда в проекте обнаруживаются различные переводы одной и той же строки. Также это может привести к несогласованности в отображаемых проверках. Другие переводы этой строки можно найти на вкладке *Другие вхождения*.

Примечание: Также эта проверка срабатывает в том случае, если строка в одном компоненте переведена, а в другом — нет. Она может использоваться, чтобы по-быстрому вручную обработать строки, не переведённые в некоторых компонентах; для этого надо просто на вкладке *Другие вхождения* нажать на кнопку *Использовать этот перевод*, отображаемую напротив каждой строки.

Для автоматизации перевода новых добавленных строк, которые уже переведены в другом компоненте, вы можете использовать надстройку *Автоматический перевод*.

См.также:

Поддержание единого перевода в разных компонентах

Используется кашида

Декоративные буквы кашиды не должны использоваться

Добавлено в версии 3.5.

Декоративные буквы кашида не должны использоваться в переводе. Также они известны как татвиль.

См.также:

Кашида в Википедии

Ссылки Markdown

Ссылки Markdown-разметки не соответствуют источнику

Добавлено в версии 3.5.

Ссылки Markdown-разметки не соответствуют источнику.

См.также:

Ссылки Markdown

Сноски Markdown

Ссылки на сноски Markdown не соответствуют источнику

Добавлено в версии 3.5.

Ссылки на сноски Markdown не соответствуют источнику.

См.также:

Markdown links

Синтаксис Markdown

Синтаксис Markdown-разметки не соответствует источнику

Добавлено в версии 3.5.

Синтаксис Markdown-разметки не соответствует источнику

См.также:

Markdown span elements

Максимальная длина перевода

Перевод не должен превышать заданную длину

Проверяет, что переводы имеют приемлемую длину, чтобы укладываться в доступное пространство. Эта проверка проверяет только длину символов перевода.

В отличие от других проверок, флаг для этой проверки должен быть установлен в виде пары `ключ:значение`, например, `max-length:100`.

Подсказка: Эта проверка смотрит на количество символов, что может быть не самой лучшей метрикой при использовании для отображения текста пропорциональных шрифтов. Проверка *Максимальный размер перевода* проверяет фактическую отрисовку текста.

Также для расширения позиций-заполнителей перед проверкой строки может быть полезен флаг `replacements:`.

Максимальный размер перевода

Отрисованный текст перевода не должен превышать заданный размер

Добавлено в версии 3.7.

Отрисованный текст перевода не должен превышать заданный размер. Эта проверка отрисовывает текст с переносом по строкам и проверяет, вписывается ли он в заданные границы.

Для этой проверки требуется один или два параметра — максимальная ширина и максимальное количество строк. В случае, если количество строк не указано, считается, что текст должен уместиться в одну строку.

Также директивами `font-*` вы можете настроить используемый шрифт (смотрите раздел *Настройка поведения с помощью флагов*), например, следующие флаги перевода говорят, что текст, отрисовываемый шрифтом ubuntu размера 22, должен вписаться в две строки и 500 пикселей:

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

Подсказка: Возможно, вы захотите установить директивы `font-*` в *настройках компонента*, чтобы для всех строк компонента был настроен один и тот же шрифт. В случае, если вам нужно будет настроить их для отдельной строки, вы сможете переопределить их значения для этой строки.

Также для расширения позиций-заполнителей перед проверкой строки может быть полезен флаг `replacements:`.

См.также:

Управление шрифтами, *Настройка поведения с помощью флагов*, *Максимальная длина перевода*

Несовпадающие \n

Количество \n в переводе не соответствует источнику

Экранированные символы новой строки обычно важны при форматировании вывода программы. Проверка срабатывает, если количество символов `\n` в переводе не совпадает с их количеством в исходной строке.

Несовпадающие двоеточия

Источник и перевод не заканчиваются оба двоеточием

Проверяет, что двоеточия в исходной строке и в переводе одинаковы. Также проверяется наличие двоеточий для тех языков, в которых они не используются (китайский или японский).

См.также:

Двоеточие в Википедии

Несовпадающие многоточия

Источник и перевод не заканчиваются оба многоточием

Проверяет, что конечные многоточия в исходной строке и в переводе одинаковы. Эта проверка проверяет только настоящие многоточия (символы …), но не три подряд идущих точки (`...`).

Как правило, на печати многоточие отрисовывается лучше, чем три подряд стоящих точки и лучше звучит при преобразовании текста в речь.

См.также:

Многоточие в Википедии

Несовпадающие восклицательные знаки

Источник и перевод не заканчиваются оба восклицательным знаком

Проверяет, что восклицательные знаки в исходной строке и в переводе одинаковы. Также проверяется наличие восклицательных знаков для тех языков, в которых они не используются (китайский, японский, корейский, армянский, лимбу, бирманский или нко).

См.также:

Восклицательный знак в Википедии

Несовпадающие точки

Источник и перевод не заканчиваются оба точкой

Проверяет, что точки в исходной строке и в переводе одинаковы. Также проверяется наличие точек для тех языков, в которых они не используются (китайский, японский, деванагари или урду).

См.также:

Точка в Википедии

Несовпадающие вопросительные знаки

Источник и перевод не заканчиваются оба вопросительным знаком

Проверяет, что вопросительные знаки в исходной строке и в переводе одинаковы. Также проверяется наличие вопросительных знаков для тех языков, в которых они не используются (армянский, арабский, китайский, корейский, японский, эфиопский, ваи или коптский).

См.также:

Вопросительный знак в Википедии

Несовпадающие точки с запятой

Источник и перевод не заканчиваются оба точкой с запятой

Проверяет, что точки с запятой в конце предложений в исходной строке и в переводе одинаковы. Эта проверка может быть полезна, например, для сохранения форматирования в записях файлов `.desktop`.

См.также:

Точка с запятой в Википедии

Несовпадающие переводы строк

Количество символов новой строки в переводе не соответствует источнику

Символы новой строки обычно важны при форматировании вывода программы. Проверка срабатывает, если количество литералов``n`` в переводе не совпадает с их количеством в исходной строке.

Нет форм множественного числа

Некоторые формы множественного числа не переведены

Проверяет, что были переведены все формы множественного числа исходной строки. Сведения о том, как используется каждая форма множественного числа, можно найти в определении строки.

Отсутствие перевода в форме множественного числа в некоторых случаях приводит к тому, что при использовании формы множественного числа ничего не будет отображаться.

Символы-заполнители

В переводе пропущены некоторые символы-заполнители:

Добавлено в версии 3.9.

Изменено в версии 4.3: В качестве заполнителя вы можете использовать регулярное выражение.

В переводе пропущены некоторые символы-заполнители. Эти символы либо извлекаются из файла перевода, либо определяются вручную с помощью флага `placeholders`; если их несколько, они разделяются двоеточием, а строки с пробелами могут быть заключены в кавычки:

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

Если у вас есть регулярный синтаксис для имён заполнителей, то вы можете применить регулярные выражения:

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

См.также:

Настройка поведения с помощью флагов

Пробелы и пунктуация

Отсутствует неразрывный пробел перед знаком двойной пунктуации

Добавлено в версии 3.9.

Проверяет наличие неразрывного пробела перед знаком двойной пунктуации (восклицательным знаком, вопросительным знаком, точкой с запятой или двоеточием). Это правило используется только в некоторых избранных языках, вроде французского или бретонского, где по типографским правилам обязателен пробел перед знаком двойной пунктуации.

См.также:

Пробелы во французском и английском языках в (англоязычной) Википедии

Регулярное выражение

Перевод не соответствует регулярному выражению:

Добавлено в версии 3.9.

Перевод не соответствует регулярному выражению. Это выражение либо извлекается из файла перевода, либо определяется вручную с помощью флага `regex`:

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

Одинаковые формы множественного числа

Некоторые формы множественного числа переведены одинаково

Проверка срабатывает, если некоторые формы множественного числа в переводе переведены одинаково. В большинстве языков они должны быть разными.

Пустая строка в начале

Источник и перевод не начинаются оба с новой строки

Обычно символы новой строки в исходных текстах появляются по уважительной причине, их пропуск или добавление могут привести к проблемам с форматированием при использовании переведённого текста.

См.также:

Пустая строка в конце

Пробелы в начале

Источник и перевод не начинаются оба одинаковым числом пробелов

Пробелы в начале строки обычно используются для набивания отступов в интерфейсе, поэтому их важно сохранить.

Пустая строка в конце

Источник и перевод не заканчиваются оба новой строкой

Обычно символы новой строки в исходных текстах появляются по уважительной причине, их пропуск или добавление могут привести к проблемам с форматированием при использовании переведённого текста.

См.также:

Пустая строка в начале

Пробелы в конце

Источник и перевод не заканчиваются оба пробелом

Проверяет, что количество конечных пробелов в исходной строке и в переводе одинаково.

Конечные пробелы обычно используется для добавления отступа до соседних элементов, поэтому их удаление может поломать макет.

Перевод без изменения

Источник и перевод тождественны

Возникает, если исходная строка и соответствующая строка перевода идентичны, по крайней мере в одной из форм множественного числа. Некоторые строки, обычно встречающиеся во всех языках, игнорируются, а различная разметка удаляется. Это уменьшает количество ложных срабатываний.

Эта проверка может помочь найти ошибочно не переведённые строки.

Поведение этой проверки по умолчанию заключается в исключении из проверки слов из встроенного чёрного списка. Это такие слова, которые зачастую не переводятся. Это полезно для того, чтобы избежать ложных срабатываний на коротких строках, которые состоят только из одного слова, одинакового в нескольких языках. Этот чёрный список можно отключить, добавив в строку или в компонент флаг `strict-same`.

См.также:

Настройки компонента, *Настройка поведения с помощью флагов*

Небезопасный HTML-код

Перевод использует небезопасную HTML-разметку

Добавлено в версии 3.9.

Перевод использует небезопасную HTML-разметку. Эта проверка должна быть включена с помощью флага `safe-html` (смотрите раздел *Настройка поведения с помощью флагов*). Также имеется соответствующий автоматический исправляльщик, который может автоматически почистить разметку.

См.также:

Проверка HTML осуществляется с помощью разработанной Mozilla библиотеки **`Bleach <https://bleach.readthedocs.io/`_**.

URL-адреса

Перевод не содержит URL

Добавлено в версии 3.5.

Перевод не содержит URL-адрес. Эта проверка срабатывает только в том случае, если единица перевода помечена как содержащая URL-адрес. В этом случае перевод должен быть корректным URL-адресом.

XML-разметка

XML-теги в переводе не соответствуют источнику

Обычно это означает, что полученный результат будет выглядеть по-другому. В большинстве случаев такое изменение из-за перевода не является желаемым результатом, но иногда именно это и надо.

Проверяет, что XML-теги в исходной строке и в переводе одинаковы.

Синтаксис XML

Перевод не является корректным XML

Добавлено в версии 2.8.

Разметка XML некорректна.

Пробел нулевой ширины

Перевод содержит лишние символы пробела нулевой ширины

Символы пробела нулевой ширины (<U+200B>) используются для разбиения сообщений по словам (переноса по словам).

Поскольку обычно они вставляются по ошибке, эта проверка срабатывает на их присутствие в переводе. В некоторых программах при использовании этого символа могут возникнуть проблемы.

См.также:

Пробел нулевой ширины в (англоязычной) Википедии

1.6.4 Проверки для исходных строк

Проверки для исходных строк могут помочь разработчикам улучшить качество исходных строк.

Многоточие

В этой строке используются три точки (…) вместо символа многоточия (…)

Эта проверка срабатывает, когда в строке используются три подряд стоящих точки (`...`), когда вместо них должен использоваться символ многоточия (…).

Использование юникодного символа в большинстве случаев является лучшим подходом; он лучше выглядит при отрисовке текста и может лучше звучать при преобразовании текста в речь.

См.также:

Многоточие в Википедии

Давно не переводилась

Строка давно не переводилась

Добавлено в версии 4.1.

Если строка долгое время не переводилась, это может указывать на проблему в исходной строке, затрудняющую её перевод.

Множество неудачных проверок

Переводы на несколько языков имеют неудачные проверки

Во многих переводах этой строки имеются неудачные проверки качества. Обычно это свидетельствует о том, что нужно что-то сделать для улучшения исходной строки.

Часто неудачная проверка может быть вызвана пропущенной точкой в конце предложения или подобными мелкими проблемами, которые переводчики склонны исправлять в переводе, в то время как лучше было бы их исправить в исходной строке.

Несколько переменных без названий

Строка содержит несколько неименованных переменных, что не позволяет переводчикам их переупорядочивать

Добавлено в версии 4.1.

Строка содержит несколько неименованных переменных, что не позволяет переводчикам их переупорядочивать.

Рассмотрите возможность использования именованных переменных вместо неименованных, чтобы позволить переводчикам менять их порядок в переводимом тексте.

Формы единственного числа

Строка содержит множественное число, но не использует формы множественного числа

Строка содержит множественное число, но не использует формы множественного числа. В случае, если ваша система перевода его поддерживает, вы должны использовать вариант, учитывающий множественное число.

Например, при использовании Python'а и Gettext'а это может выглядеть так:

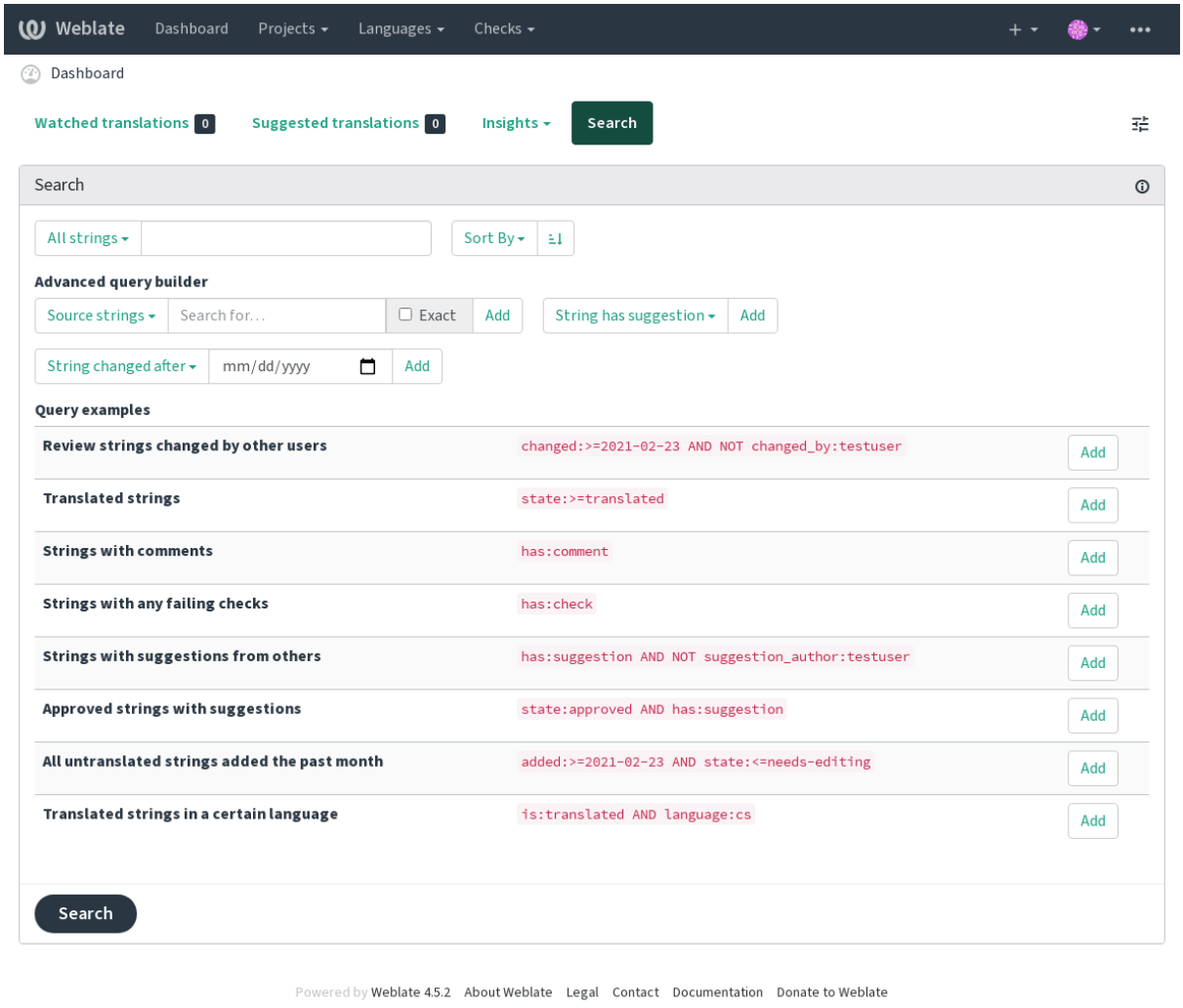
```
from gettext import ngettext

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

1.7 Поиск

Добавлено в версии 3.9.

Для поиска нужных строк можно использовать расширенные поисковые запросы, включающие в себя логические операции, круглые скобки или поиск по конкретным полям.

Если поле не определено, поиск проводится по полям *Источник*, *Перевод* и *Контекст*.



1.7.1 Простой поиск

Любая фраза, введённая в поле поиска, разбивается на слова. При поиске показываются строки, содержащие любое из этих слов. Чтобы найти точную фразу, поместите \»фразу для поиска\» в кавычки (работают как одинарные (\“), так и двойные (\») кавычки): `"это строка, заключённая в кавычки"` или `'другая закавыченная строка'`.

1.7.2 Поля

`source:ТЕКСТ` Регистронезависимый поиск по исходной строке.

`target:ТЕКСТ` Регистронезависимый поиск по целевой строке.

`context:ТЕКСТ` Регистронезависимый поиск по строке контекста.

`key:ТЕКСТ` Регистронезависимый поиск по строке ключа.

`note:ТЕКСТ` Регистронезависимый поиск по строке комментария.

`location:ТЕКСТ` Регистронезависимый поиск по строке местоположения.

`priority:ЧИСЛО` Приоритет строки.

`added:ДАТА_И_ВРЕМЯ` Метка времени, когда строка была добавлена в Weblate.

state:ТЕКСТ Поиск по состоянию (`approved`, `translated`, `needs-editing`, `empty`, `read-only`), поддерживает *Операторы поиска по полям*.

pending:ЛОГИЧЕСКОЕ_ЗНАЧЕНИЕ Строка, ожидающая сброса в систему контроля версий.

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

is:ТЕКСТ Поиск по состояниям строки (`pending`, `translated`, `untranslated`).

language:ТЕКСТ Целевой язык строки.

component:ТЕКСТ Плашка компонента, смотрите раздел *Плашка компонента*.

project:ТЕКСТ Плашка проекта, см. *URL-плашка*.

changed_by:ТЕКСТ Строка была изменена автором с указанным именем пользователя.

changed:ДАТА_И_ВРЕМЯ Дата изменения содержимого строки, поддерживает *операторы поиска по полям*.

change_time:ДАТА_И_ВРЕМЯ Дата изменения строки, поддерживает *операторы поиска по полям*; в отличии от `changed` это также включает и те события, которые не изменяют само содержимое строки, а также вы можете фильтровать события по конкретным действиям с помощью `change_action`.

change_action:ТЕКСТ 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 *Поиск по изменениям* for examples.

check:ТЕКСТ Строка имеет неудачные проверки.

dismissed_check:ТЕКСТ Строка имеет пропущенные проверки.

comment:ТЕКСТ Поиск в пользовательских комментариях.

comment_author:ТЕКСТ Фильтр по авторским комментариям.

suggestion:ТЕКСТ Поиск в предложениях.

suggestion_author:ТЕКСТ Фильтр по автору предложения.

explanation:TEXT Поиск в пояснениях.

1.7.3 Логические операторы

Вы можете комбинировать поисковые запросы, используя для формирования сложных запросов операторы `AND`, `OR` и `NOT` и круглые скобки. Например: `state:translated AND (source:hello OR source:bar)`

1.7.4 Операторы поиска по полям

Вы можете указать операторы, диапазоны или частичный поиск по дате или числам:

state:>=translated Состояние `translated` или лучше (`approved`).

changed:2019 Изменено в 2019 году.

changed:[2019-03-01 to 2019-04-01] Изменено между двумя указанными датами.

1.7.5 Оператор точного поиска

С помощью оператора = вы можете сделать запрос на точное совпадение различных строковых полей. Например, для нахождения всех исходных строк, точно равных строке `hello world`, используйте запрос `source:="hello world"`. При поиске односложных выражений кавычки можно опустить. Например, для нахождения всех исходных строк, точно равных строке `hello`, вы можете использовать запрос `source:=hello`.

1.7.6 Поиск по изменениям

Добавлено в версии 4.4.

Поиск по истории событий можно сделать с помощью операторов `change_action` и `change_time`.

Например, поиск по строкам, отмеченным для правки (*marked for edit*) в 2018 году, можно сделать с помощью: `change_time:2018 AND change_action:marked-for-edit` или `change_time:2018 AND change_action:"Marked for edit"`.

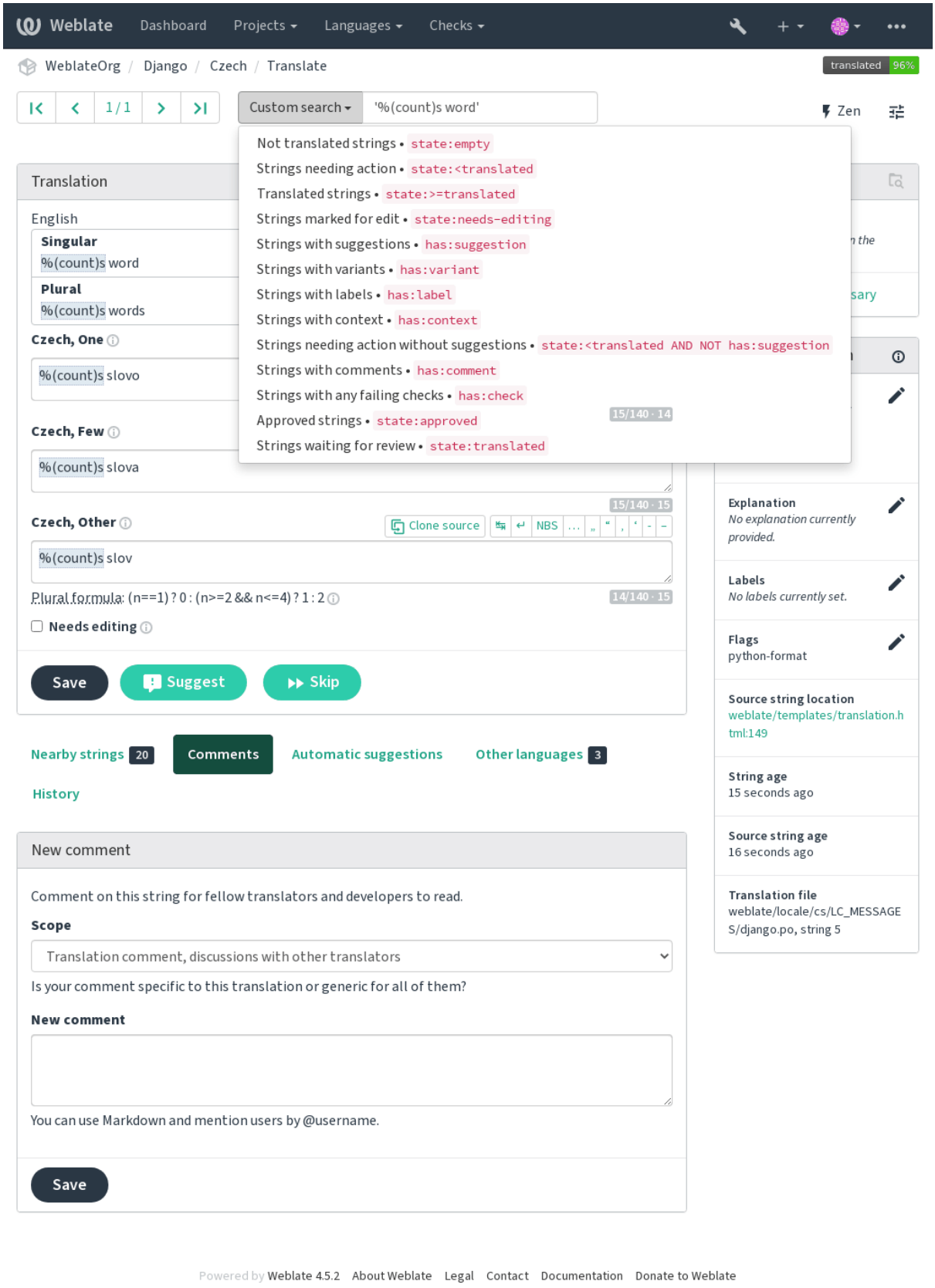
1.7.7 Регулярные выражения

Везде, где принимается текст, вы также можете задать и регулярное выражение в виде `r"выражение"`.

Например, для поиска по исходным строкам, содержащим любую цифру от 2 до 5, используйте запрос `source:r"[2-5]"`.

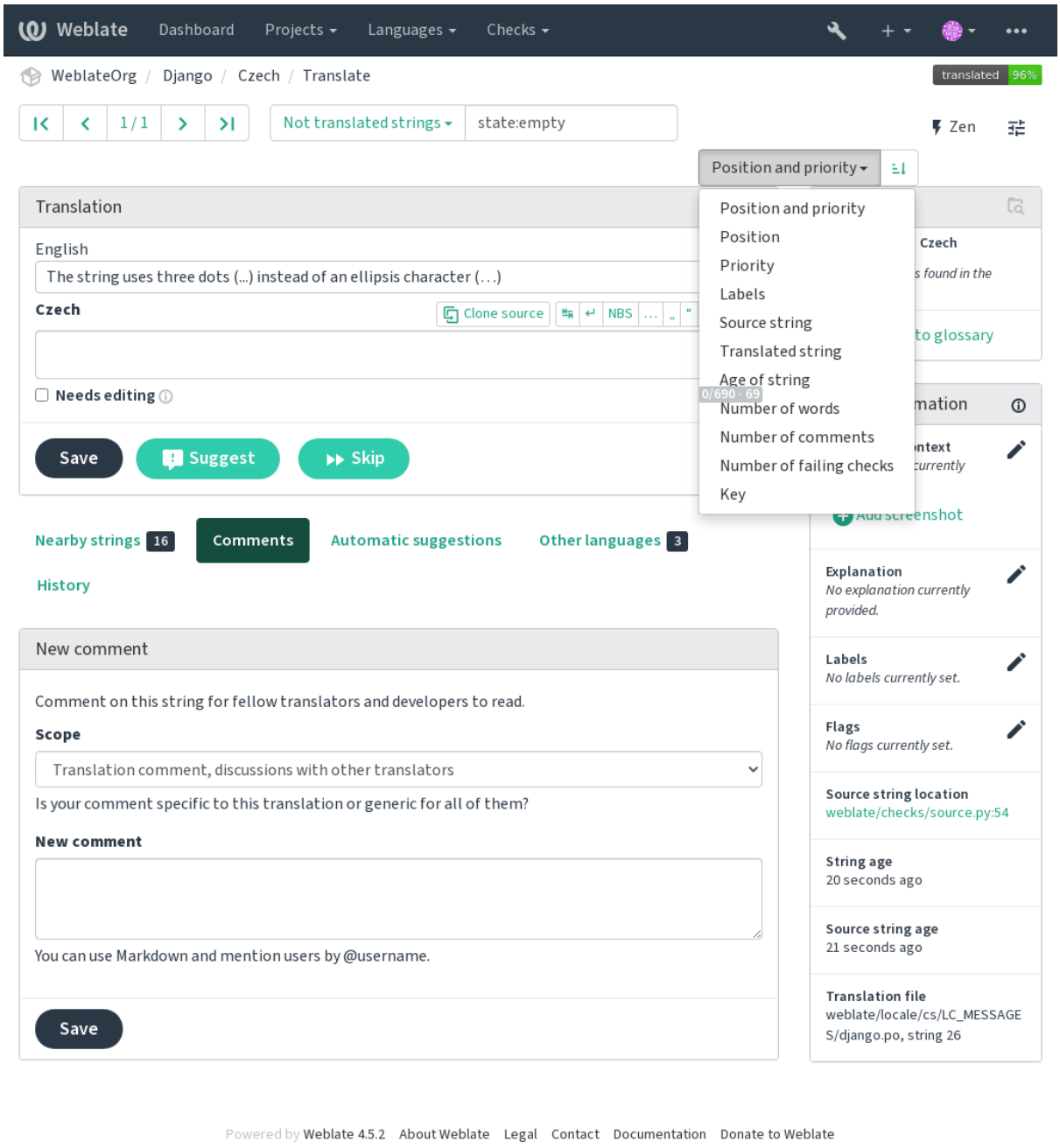
1.7.8 Предопределённые запросы

На странице поиска можно выбрать один из предопределённых запросов, которые позволяют вам быстро получить доступ к наиболее часто встречающимся поисковым запросам:


Weblate
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Czech / Translate
translated 96%
1 / 1
Custom search
'%(count)s word'
Zen
Not translated strings • state:empty
Strings needing action • state:<translated
Translated strings • state:>=translated
Strings marked for edit • state:needs-editing
Strings with suggestions • has:suggestion
Strings with variants • has:variant
Strings with labels • has:label
Strings with context • has:context
Strings needing action without suggestions • state:<translated AND NOT has:suggestion
Strings with comments • has:comment
Strings with any failing checks • has:check
Approved strings • state:approved
Strings waiting for review • state:translated
Translation
English
Singular
%(count)s word
Plural
%(count)s words
Czech, One
%(count)s slovo
Czech, Few
%(count)s slova
Czech, Other
Clone source
NBS
%(count)s slov
Plural formula: (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2
Needs editing
Save
Suggest
Skip
Nearby strings 20
Comments
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
15 seconds ago
Source string age
16 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5
Powered by Weblate 4.5.2
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.7.9 Упорядочивание результатов

Существует множество вариантов сортировки строк в соответствии с вашими потребностями:



1.8 Рабочие процессы перевода

Использование Weblate — это процесс, который приближает пользователей к вам путём приближения вас к вашим переводчикам. Вы сами решаете, какой объём его возможностей вы хотите использовать.

Ниже приведён не полный список способов настройки Weblate. Вы можете основывать другие рабочие процессы на перечисленных здесь самых обычных примерах.

1.8.1 Доступ к переводу

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.

В следующих главах *любой пользователь* означает пользователя, имеющего доступ к переводу. Это может быть любой аутентифицированный пользователь, если проект является публичным, или пользователь, имеющий в проекте разрешение *Перевод*.

1.8.2 Состояния перевода

Каждая переведённая строка может находиться в одном из следующих состояний:

Не переведено Перевод пуст, он может как храниться в файле, так и не храниться в нём, в зависимости от формата файла.

На правку 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).

Ожидает рецензирования Перевод выполнен, но ещё не проверен. Он хранится в файле как готовый перевод.

Одобрено Перевод был одобрен в процессе рецензирования. Он больше не может быть изменён переводчиками — только рецензентами. Переводчики могут только добавлять к нему предложения.

Предложения Предложения хранятся только в Weblate, а не в файле перевода.

The states are represented in the translation files when possible.

Подсказка: In case file format you use does not support storing states, you might want to use *Пометка неизменённых переводов флагом «На правку»* addon to flag unchanged strings as needing editing.

См.также:

Возможности типов переводов, *Рабочие процессы перевода*

1.8.3 Прямой перевод

Это самая обычная настройка для небольших команд, любой может переводить напрямую. Также она действует по умолчанию в Weblate.

- *Любой пользователь* может редактировать переводы.
- Предложения являются необязательным способом предложения изменений, когда переводчики не уверены в этом изменении.

Параметр	Значение	Примечание
Включить рецензии	выкл	Настраивается на уровне проекта.
Включить предложения	вкл	Полезны для пользователей в качестве способа предложения изменений, когда они в них не уверены.
Голосование по предложению	выкл	
Автоматическое принятие предложений	0	
Группа переводчиков	*Пользователи*	Or *Translate* with *per-project access control*.
Группа рецензентов	Н/Д	Не используется.

1.8.4 Коллегиальное рецензирование

В этом рабочем процессе любой может добавлять предложения и чтобы они стали переводом, они должны получить одобрение от другого участника (или участников).

- *Любой пользователь* может добавлять предложения.
- *Любой пользователь* может голосовать за предложения.
- Предложения становятся переводами, когда они получают заранее определённое количество голосов.

Параметр	Значение	Примечание
Включить рецензии	выкл	Настраивается на уровне проекта.
Включить предложения	вкл	
Голосование по предложению	выкл	
Автоматическое принятие предложений	1	Вы можете установить более высокое значение, чтобы потребовать больше рецензий от коллег.
Группа переводчиков	*Пользователи*	Or *Translate* with *per-project access control*.
Группа рецензентов	Н/Д	Не используется, рецензентами являются все переводчики.

1.8.5 Выделенные рецензенты

Добавлено в версии 2.18: Соответствующий рабочий процесс рецензирования поддерживается с версии Weblate 2.18.

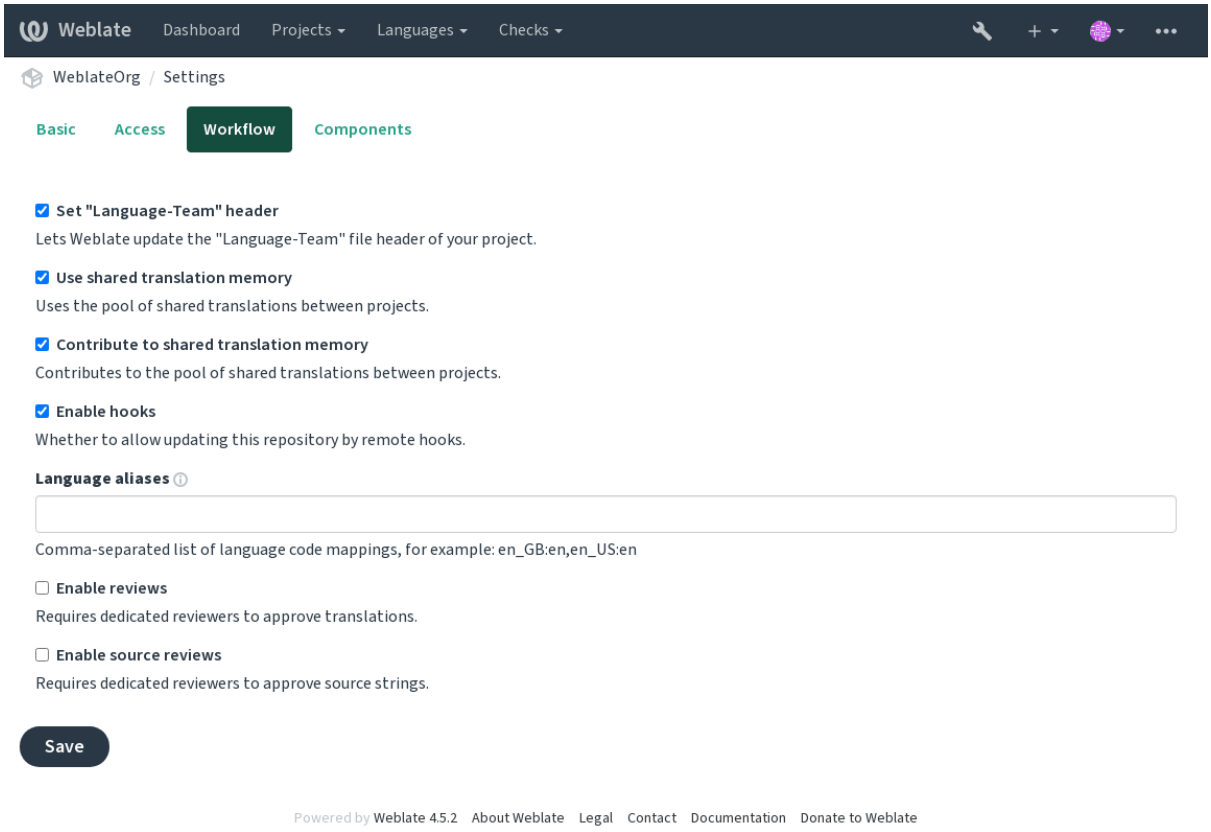
В рабочем процессе с выделенными рецензентами у вас есть две группы пользователей, одна из которых может отправлять переводы, а другая — их проверять, чтобы убедиться в том, что переводы согласованы и хорошего качества.

- *Любой пользователь* может редактировать не одобренные переводы.
- *Рецензент* может одобрять строки или снимать с них одобрение.
- *Рецензент* может редактировать все переводы (включая одобренные).
- Также для внесения предложений на изменение одобренных строк могут использовать предложения.

Параметр	Значение	Примечание
Включить рецензии	вкл	Настраивается на уровне проекта.
Включить предложения	выкл	Полезны для пользователей в качестве способа предложения изменений, когда они в них не уверены.
Голосование по предложению	выкл	
Автоматическое принятие предложений	0	
Группа переводчиков	*Пользователи*	Or *Translate* with *per-project access control*.
Группа рецензентов	*Рецензенты*	Or *Review* with *per-project access control*.

1.8.6 Включение рецензирования

Рецензирование можно включить в настройках проекта, на вкладке *Рабочий процесс* параметров проекта (*Управлять* → *Параметры* в меню проекта):

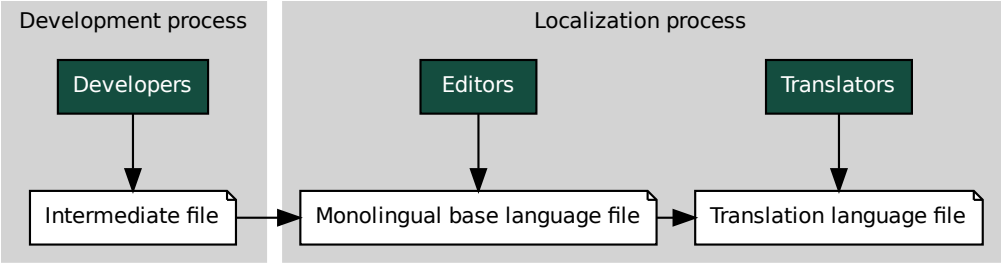


Примечание: В зависимости от настроек Weblate этот параметр может быть недоступен. Например, на Hosted Weblate он недоступен для проектов, размещённых бесплатно.

1.8.7 Контроль качества исходных строк

Во многих случаях строки на исходном языке поступают от разработчиков, так как именно они пишут код и придумывают новые исходные строки. Однако зачастую сами разработчики не являются носителями исходного языка и не могут обеспечить желаемого качества этих строк. Решить эту проблему может помочь промежуточный перевод, который будет являться неким дополнительным «шлюзом качества», расположенным между разработчиками, переводчиками и пользователями.

Если задан *файл промежуточного языка*, то он будет использоваться в качестве источника исходных строк, но их можно будет редактировать, дабы отшлифовать формулировки. Когда строка на исходном языке будет готова, она станет доступна также и для перевода на остальные языки.



См.также:

Файл промежуточного языка, *Одноязычный базовый файл*, *Двуязычные и одноязычные форматы*

1.8.8 Рецензирование исходных строк

При включённом параметре *Включить рецензии исходных строк* процесс рецензирования может быть применён также к исходным строкам. После включения этого параметра пользователи смогут сообщать о проблемах в исходных строках. Реальный процесс зависит от того, используете ли вы двуязычные или одноязычные форматы.

Для одноязычных форматов рецензирование исходной строки ничем не отличается от обычного *рецензирования* — когда сообщается о проблеме в исходной строке, она помечается флагом *На правку*.

Двуязычные форматы не позволяют напрямую редактировать исходные строки (они, как правило, извлекаются непосредственно из исходного кода). В этом случае к строкам, передаваемым переводчиками, прикрепляется метка *Исходная строка требует рецензирования*. Вы должны просмотреть такие строки и либо отредактировать их в исходном коде, либо удалить эту метку.

См.также:

Двуязычные и одноязычные форматы, *Выделенные рецензенты*, labels, *Комментарии*

1.9 Часто задаваемые вопросы

1.9.1 Конфигурация

Как создать автоматизированный рабочий процесс?

Weblate может выполнить все действия, связанные с переводом, в полуавтоматическом режиме. Если вы дадите ему доступ на отправку изменений в ваш репозиторий, переводы смогут в нём появляться без запроса подтверждения, если только не возникнут конфликты слияния.

1. Настройте свой Git-репозиторий так, чтобы он сообщал Weblate, когда происходят какие-либо изменения; для получения информации о том, как это сделать, смотрите раздел *Обработчики уведомлений*.
2. В *конфигурации вашего компонента* установите URL для отправки, это позволит Weblate'у отправлять изменения в ваш репозиторий.
3. Turn on *Отправлять при коммите* on your *Настройки компонента* in Weblate, this will make Weblate push changes to your repository whenever they happen at Weblate.

См.также:

Непрерывная локализация, *Предотвращение конфликтов слияния*

Как получить доступ к репозиториям по SSH?

Информацию о настройке SSH-ключей смотрите в разделе *Доступ к репозиториям*.

Как исправить в переводах конфликты слияния?

Время от времени, когда файл перевода изменяется одновременно и в Weblate, и в вышестоящем репозитории, происходят конфликты слияния. Обычно этого можно избежать, влив переводы из Weblate до внесения изменений в файлы перевода (например, перед запуском msgmerge). Просто скажите Weblate закоммитить все отложенные переводы (вы можете сделать это в подпункте *Обслуживание репозитория* меню *Управлять*) и влейте их в репозиторий (если не включена автоматическая отправка).

Если же вы уже нарвались на конфликт слияния, то самым простым способом решения всех конфликтов локально на вашей рабочей машине будет добавление Weblate в качестве удалённого репозитория, объединения изменений из него с изменениями из вышестоящего репозитория с исправлением возникших конфликтов. После того, как вы отправите изменения обратно, Weblate сможет использовать объединённую версию без каких-либо других специальных действий.

Примечание: В зависимости от ваших настроек, доступ к репозиторию Weblate может потребовать аутентификации. При использовании встроенного в Weblate *экспортера в Git* вы аутентифицируетесь с помощью своего имени пользователя и ключа API.

```
# 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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# 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
```

Если вы используете в Weblate несколько ветвей, то же самое вы можете сделать со всеми ними:

```
# 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
```

В случае использования файлов PO gettext'а существует способ полуавтоматического исправления конфликтов:

Извлеките и сохраните локальный клон Git-репозитория Weblate. Также возьмите второй свежий локальный клон вышестоящего Git-репозитория (то есть, вам нужно две копии вышестоящего Git-репозитория: нетронутая и рабочая):

```
# 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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
done
git commit

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

См.также:

Как экспортировать используемый Weblate репозиторий Git?, *Непрерывная локализация*, *Предотвращение конфликтов слияния*, *Клиент Weblate*

Как мне сразу перевести несколько веток?

Weblate поддерживает одновременную отправку изменений перевода внутри одного *проекта*. В каждом *компоненте*, в котором она включена (это поведение по умолчанию), внесённое изменение автоматически вносится и в прочие компоненты. Таким образом, переводы остаются синхронизированными, даже если сами ветки уже достаточно сильно разошлись, и простое объединение изменений перевода между ними невозможно.

После вливания изменений из Weblate, вам, возможно, придётся объединить эти ветви (в зависимости от вашего рабочего процесса разработки) для устранения различий:

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

См.также:

Поддержание единого перевода в разных компонентах

Как переводить многоплатформенные проекты?

Weblate поддерживает широкий диапазон форматов файлов (смотрите раздел *Поддерживаемые форматы файлов*) и самым простым подходом является использование родного формата для каждой платформы.

После того, как вы добавите все файлы перевода для платформ в качестве компонентов в один проект (смотрите раздел *Добавление проектов и компонентов перевода*), вы сможете использовать функцию распространения перевода (включена по умолчанию и может быть отключена в *настройках компонента*) для перевода строк одновременно для всех платформ.

См.также:

Поддержание единого перевода в разных компонентах

Как экспортировать используемый Weblate репозиторий Git?

Репозиторий ничем таким не выделяется, он живёт в каталоге `DATA_DIR` и называется `vcs/<проект>/<компонент>/`. Если у вас есть SSH-доступ к этой машине, вы можете работать с репозиторием напрямую.

Для анонимного доступа вы можете запустить Git-сервер и позволить ему открыть репозиторий во внешний мир.

В качестве альтернативы вы можете использовать *Экспортер Git* изнутри Weblate для автоматизации этого процесса.

Какие есть способы для отправки изменений в вышестоящий репозиторий?

В значительной степени это зависит от вашей установки, в этом плане Weblate достаточно гибок. Вот несколько примеров используемых с Weblate'ом рабочих процессов:

- Weblate автоматически отправляет и вливает изменения (смотрите раздел *Как создать автоматизированный рабочий процесс?*).
- Вы вручную приказываете Weblate отправить изменения (ему нужны права доступа на отправку изменений в вышестоящий репозиторий).
- Кто-то вручную вливает изменения из git-репозитория Weblate в вышестоящий репозиторий.
- Кто-то переписывает созданную Weblate историю (например, устраняя коммиты слияния), объединяет изменения и приказывает Weblate'у сбросить содержимое в вышестоящий репозиторий.

Конечно, вы можете смешивать все эти рабочие процессы по своему усмотрению.

Как мне ограничить Weblate доступом только к переводам, не раскрывая ему при этом исходный код?

Для отделения переводов от исходного кода вы можете использовать подмодули git, при этом вы по-прежнему держите их под контролем версий.

1. Создайте репозиторий с вашими файлами перевода.
2. Добавьте его в качестве подмодуля к своему коду:

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

3. Свяжите Weblate с этим репозиторием, ему больше не нужен доступ к репозиторию, содержащему ваш исходный код.
4. Вы можете обновлять основной репозиторий переводами из Weblate, выполняя команду:

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

Для получения более подробной информации обратитесь к документации по подмодулям git.

Как мне проверить, правильно ли настроен мой Weblate?

Weblate включает в себя набор проверок конфигурации, которые можно посмотреть, перейдите по ссылке *Отчёт о производительности* в интерфейсе администратора, или напрямую откройте адрес `/manage/performance/`.

Почему все коммиты от имени Weblate имеют электронную почту <noreply@weblate.org>?

Это умолчательное имя коммитящего, настроенное при создании компонента перевода. Вы в любое время можете изменить его в административной зоне сайта.

В любом случае, в качестве пользователя, сделавший перевод всё равно корректно записывается автор коммита (если это поддерживает нижележащая система контроля версий).

См.также:

Настройки компонента

1.9.2 Использование

Как мне отрецензировать переводы других переводчиков?

- В Weblate доступно на выбор несколько вариантов рабочего процесса, основанные на рецензировании, см. *Рабочие процессы перевода*.
- Вы можете подписаться на любые изменения на вкладке *Уведомления*, а затем проверять вклады других переводчиков по мере их поступления по электронной почте.
- В нижней части представления перевода имеется инструмент рецензирования, с помощью которого можно просматривать переводы, выполненные другими пользователями с заданной даты.

См.также:

Рабочие процессы перевода

Как мне предоставить обратную связь по исходной строке?

На контекстных вкладках, расположенных ниже перевода, вы можете использовать вкладку *Комментарии* для предоставления обратной связи по исходной строке или для её обсуждения с другими переводчиками.

См.также:

report-source, *Комментарии*

Как я могу использовать при переводе уже существующие переводы?

- Благодаря разделяемой памяти переводов могут быть использованы все переводы, загруженные в Weblate.
- Вы можете импортировать в Weblate свою уже существующую память переводов из файлов.
- Используйте функциональность импорта для загрузки компендиума в качестве переводов, предложений или переводов, нуждающихся в рецензировании. Это лучший подход для одноразового перевода с использованием компендиума или аналогичной базы данных переводов.
- Вы можете настроить *tmserver* со всеми имеющимися у вас базами данных и позволить Weblate'у его использовать. Это хороший вариант, если вы хотите использовать его при переводе несколько раз.
- Другим вариантом является перенос всех связанных проектов на один экземпляр Weblate, что позволит автоматически подбирать переводы из других проектов.

См.также:

Машинный перевод, *Автоматические предложения*, *Память переводов*

Обновляет ли Weblate помимо самих переводов также и файлы перевода?

Weblate пытается свести к минимуму изменения в файлах перевода. Для некоторых форматов файлов это, к сожалению, может привести к переформатированию файла. Если вы хотите, чтобы файл был отформатирован по-вашему, пожалуйста, используйте для этого перехватчик pre-commit.

См.также:

updating-target-files

Откуда берутся определения языка и как я могу добавить свои собственные определения?

Базовый набор определений языка входит в состав Weblate и Translate-toolkit. Он охватывает более 150 языков и включает информацию о множественных формах или направлении текста.

Вы можете определять свои собственные языки в интерфейсе администратора. Для этого вам нужно просто ввести необходимую информацию о них.

См.также:

Определение языков

Может ли Weblate подсветить изменения в строке «на правку»?

Weblate поддерживает такую подсветку, однако, чтобы показать различия, ему нужны данные.

К примеру, для PO-файлов Gettext при обновлении PO-файлов необходимо в вызов команды **msgmerge** передать параметр `--previous`:

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

Для одноязычных переводов Weblate может найти предыдущую строку по её идентификатору, поэтому различия он показывает автоматически.

Почему Weblate до сих пор показывает старые строки перевода, хотя я уже обновил шаблон?

Weblate не пытается манипулировать файлами перевода никаким другим способом, кроме как разрешения переводчикам их переводить. Поэтому он также не обновляет переводимые файлы при изменении шаблона или исходного кода. Вам просто нужно сделать это вручную и внести изменения в репозиторий, после чего Weblate автоматически подхватит изменения.

Примечание: Обычно перед обновлением файлов перевода полезно влить изменения, сделанные в Weblate, в основной репозиторий, так как в противном случае вы, вероятно, столкнётесь с некоторыми конфликтами при слиянии.

Например, если вы используете PO-файлы gettext, вы можете обновить файлы перевода с помощью инструмента **msgmerge**:

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

В случае, если вы хотите выполнять такое обновление автоматически, вы можете установить надстройку *Обновлять PO-файлы до соответствия POT (msgmerge)*.

См.также:

updating-target-files

1.9.3 Решение проблем

Запросы иногда завершаются с ошибкой «слишком много открытых файлов» (“too many open files”)

Такое иногда случается, когда ваш Git-репозиторий слишком разрастается и у вас их очень много. Сжатие Git-репозиториев улучшит эту ситуацию.

Простейший способ это сделать — запустить следующий скрипт:

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

См.также:

DATA_DIR

При доступе к сайту я получаю ошибку «Плохой запрос (400)» (“Bad Request (400)”)

Скорее всего это вызвано неправильной настройкой параметра *ALLOWED_HOSTS*. Он должен содержать все имена хостов, с которых вы хотите получать доступ к вашему Weblate’у. Например:

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

См.также:

Настройка разрешённых хостов

Что означает сообщение «Есть больше файлов для одного языка (en)»?

Обычно такое происходит, когда у вас есть файл перевода для исходного языка. Weblate отслеживает исходные строки и резервирует для этого язык оригинала. Дополнительный файл для того же самого языка не обрабатывается.

- В случае необходимости перевода на исходный язык в настройках компонента необходимо изменить параметр *Исходный язык*.
- Если файл перевода для исходного языка не нужен, пожалуйста, удалите его из репозитория.
- Если же файл перевода для исходного языка необходим, но должен игнорироваться Weblate, пожалуйста, настройте для его исключения *языковой фильтр*.

Подсказка: Вы также можете получить аналогичное сообщение об ошибке и для других языков. В таком случае, скорее всего, причина будет в том, что несколько разных файлов отображаются на один язык в Weblate.

Это может быть вызвано тем, что используются устаревшие коды языков одновременно с новыми (`ja` и `jp` для японского) или используется одновременно код, привязанный к стране, и основной (`ru` и `ru_RU`). Смотрите подробности в разделе «*Разбор кодов языков*».

1.9.4 Возможности

Поддерживает ли Weblate другие системы контроля версий, кроме Git и Mercurial?

В настоящее время Weblate не имеет родной поддержки ни для чего, кроме *Git* (с расширенной поддержкой *GitHub*, *Gerrit* и *Subversion*) и *Mercurial*, но вполне возможно дописать поддержку и других систем контроля версий.

Также для доступа к другим системам контроля версий вы можете использовать *Удалённые помощники Git* в Git'е .

Weblate also supports VCS-less operation, see *Локальные файлы*.

Примечание: Для родной поддержки других систем контроля версий Weblate требует использования распределённых СКВ, и, вероятно, может быть настроен на работу с любой системой, помимо Git или Mercurial, но кто-то должен эту поддержку реализовать.

См.также:

Интеграция с системой контроля версий

Как Weblate сообщает о том, кто какую строку перевёл?

Каждое изменение, сделанное через Weblate, фиксируется в системе контроля версий под именем переводчика. Таким образом, каждое изменение имеет надлежащее авторство, и вы можете отследить его с помощью стандартных инструментов СКВ, которые вы используете для кода.

Кроме того, если формат файла перевода это поддерживает, он обновляет заголовки файлов и включает в них имя переводчика.

См.также:

`list_translators`, ../devel/reporting

Почему Weblate принудительно показывает все PO-файлы в одном дереве?

Weblate был спроектирован таким образом, что каждый PO-файл представляется одним компонентом. Это выгодно переводчиками, поскольку они знают, что они переводят на самом деле.

Изменено в версии 4.2: Переводчики могут переводить все компоненты одного проекта на один конкретный язык как единое целое.

Почему Weblate использует такие коды языков, как sr_Latn или zh_Hant?

Это коды языков, определяемые **RFC 4646**, чтобы лучше показать, что это и в самом деле разные языки, вместо ранее ошибочно используемых модификаторов (для вариантов `@latin`) или кодов стран (для китайского языка).

Weblate по-прежнему понимает старые коды языков и сопоставляет их с текущими — например, `sr@latin` будет обрабатываться как `sr_Latn`, а `zh@CN` — как `zh_Hans`.

См.также:

Определение языков

1.10 Поддерживаемые форматы файлов

Weblate поддерживает большинство форматов перевода, понимаемых translate-toolkit, однако у каждого формата есть свои особенности, поэтому могут возникнуть некоторые проблемы с форматами, которые были протестированы недостаточно тщательно.

См.также:

Форматы файлов, относящиеся к переводу

Примечание: При выборе формата файла для вашего приложения лучше придерживаться какого-нибудь устоявшегося формата для используемого вами инструментария/платформы. Таким образом, ваши переводчики в дополнение смогут использовать любые инструменты, к которым они привыкли, и вероятность внесения ими своего вклада в ваш проект будет выше.

1.10.1 Двуязычные и одноязычные форматы

Поддерживаются как одноязычные, так и двуязычные форматы. Двуязычные форматы хранят в одном файле два языка — исходный язык и перевод (типовые примеры: *GNU gettext*, *XLIFF* или *Строки Apple iOS*). Одноязычные же форматы, напротив, идентифицируют строку по ID, и каждый файл перевода содержит только сопоставление этих ID с одним конкретным языком (например, *Строковые ресурсы Android*). Некоторые форматы файлов могут использоваться и так, и так; подробности смотрите ниже.

Для правильного использования одноязычных файлов Weblate требует доступа к файлу, содержащему полный список строк для перевода и их «перевод» на исходный язык — этот файл внутри Weblate называется *одноязычным базовым файлом*, хотя в вашей парадигме наименование этой сущности может отличаться.

Кроме того, этот рабочий процесс может быть расширен с помощью *файла промежуточного языка*, позволяющего включить в перевод строки, предоставленные разработчиками, но не использованными «как есть» в окончательных строках.

1.10.2 Автоматическое обнаружение

Weblate умеет автоматически обнаруживать некоторые широко распространённые форматы файлов, но это обнаружение может навредить вашей производительности и ограничено возможностями, специфичными для данного формата файла (к примеру, автоматическим добавлением новых переводов).

1.10.3 Возможности типов переводов

Возможности всех поддерживаемых форматов:

Формат	Язычность[c. 63, 1]	Множественные формы[c. 63, 2]	Комментарии[c. 63, 3]	Контекст[c. 63, 4]	Расположение[c. 63, 5]	Флаги[c. 63, 8]	Дополнительные состояния[c. 63, 6]
GNU gettext	двуязычный	да	да	да	да	да[9]	требуется правка
Одноязычный gettext	одноязычный	да	да	да	да	да[?]	требуется правка
XLIFF	оба	да	да	да	да	да[10]	требуется правка, одобрено

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Формат	Язычность[c. 63, 1]	Множественные формы[c. 63, 2]	Комментарии[c. 63, 3]	Контекст[c. 63, 4]	Расположение[c. 63, 5]	Флаги[c. 63, 8]	Дополнительные состояния[c. 63, 6]
Файлы свойств Java	оба	нет	да	нет	нет	нет	
Файлы свойств GWT	одноязычный	да	да	нет	нет	нет	
Переводы Joomla	одноязычный	нет	да	нет	да	нет	
Файлы .ts Qt Linguist	оба	да	да	нет	да	да[?]	требуется правка
Строковые ресурсы Android	одноязычный	да	да[7]	нет	нет	да[?]	
Строки Apple iOS	двуязычный	нет	да	нет	нет	нет	
Строки PHP	одноязычный	нет[11]	да	нет	нет	нет	
JSON-файлы	одноязычный	нет	нет	нет	нет	нет	
JSON -файлы i18next	одноязычный	да	нет	нет	нет	нет	
JSON-файлы go-i18n	одноязычный	да	нет	нет	нет	нет	
Файл ARB	одноязычный	да	да	нет	нет	нет	
JSON-файлы WebExtension	одноязычный	да	да	нет	нет	нет	
.XML-файлы ресурсов	одноязычный	нет	да	нет	нет	да[?]	
Файлы CSV	оба	нет	да	да	да	нет	требуется правка
Файлы YAML	одноязычный	нет	да	нет	нет	нет	
Файлы Ruby YAML	одноязычный	да	да	нет	нет	нет	
Файлы DTD	одноязычный	нет	нет	нет	нет	нет	
Плоские XML-файлы	одноязычный	нет	нет	нет	нет	да[?]	
Файлы ресурсов Windows	одноязычный	нет	да	нет	нет	нет	
Excel Open XML	одноязычный	нет	да	да	да	нет	требуется правка

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Формат	Язычность[c. 63, 1]	Множественные формы[c. 63, 2]	Комментарии[c. 63, 3]	Контекст[c. 63, 4]	Расположение[c. 63, 5]	Флаги[c. 63, 8]	Дополнительные состояния[c. 63, 6]
Файлы метаданных магазина приложений	одноязычный	нет	нет	нет	нет	нет	
Файлы субтитров	одноязычный	нет	нет	нет	да	нет	
Файлы HTML	одноязычный	нет	нет	нет	нет	нет	
Формат OpenDocument	одноязычный	нет	нет	нет	нет	нет	
Формат IDML	одноязычный	нет	нет	нет	нет	нет	
Переводы в INI-файлах	одноязычный	нет	нет	нет	нет	нет	
Переводы в INI-файлах Inno Setup	одноязычный	нет	нет	нет	нет	нет	
TermBase eXchange format	двуязычный	нет	да	нет	нет	да[?]	

1.10.4 GNU gettext

Самый широкоиспользуемый формат для перевода свободного програмного обеспечения.

Контекстная информация, хранящаяся в файле, поддерживается настройкой его заголовков или его привязкой к соответствующим исходным файлам.

Двуязычный PO-файл gettext обычно выглядит следующим образом:

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

#: weblate/media/js/bootstrap-datepicker.js:1421
```

(continues on next page)

[1] Смотрите раздел *Двуязычные и одноязычные форматы*

[2] Множественные формы необходимы для правильной локализации строк, содержащих переменную количества.

[3] Комментарии могут быть использованы для указания дополнительной информации о строке для перевода.

[4] Контекст используется для различения одинаковых строк, используемых в разных областях (например, *Sun* может означать как сокращённое название дня «воскресенье» — «Sunday» — так и название нашей ближайшей звезды).

[5] Расположение строки в исходном коде может помочь опытным переводчикам понять предназначение этой строки.

[8] Смотрите раздел *Настройка поведения с помощью флагов*

[6] Дополнительные состояния, поддерживаемые форматом файла в дополнение к состояниям «Не переведено» и «Переведено».

[9] В качестве флагов используются комментарии типа gettext.

[10] Для всех форматов на основе XML флаги извлекаются из нестандартного атрибута `weblate-flags`. Кроме того, флаг `max-length:N` поддерживается через атрибут `maxwidth`, как определено в стандарте XLIFF, смотрите раздел *Указание флагов перевода*.

[7] Комментарий XML, размещённый перед элементом `<string>`, трактуется как комментарий разработчика.

[11] Множественные формы поддерживаются только для Laravel, который использует для их определения внутристрочный синтаксис, смотрите документ Локализация в Laravel.

(продолжение с предыдущей страницы)

```
msgid "Tuesday"
msgstr "Úterý"

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

Типовые *настройки компонента* Weblate	
Маска файла	`po/*.po`
Одноязычный базовый файл	*Пусто*
Шаблон для новых переводов	`po/messages.pot`
Формат файла	*PO-файл gettext*

См.также:

devel/gettext, devel/sphinx, Gettext в Википедии, PO-файлы, *Обновление переменной ALL_LINGUAS в файле «configure»*, *Настройка вывода gettext*, *Обновление файла LINGUAS*, *Создание MO-файлов*, *Обновлять PO-файлы до соответствия POT (msgmerge)*

Одноязычный gettext

Некоторые проекты решают использовать gettext в качестве одноязычного формата — в их исходном коде присутствуют только идентификаторы и строка должна быть переведена на все языки, в том числе и на английский. Такая конфигурация поддерживается, хотя для этого и надо явно выбирать этот формат файлов при импорте компонентов в Weblate.

Одноязычный PO-файл gettext обычно выглядит следующим образом:

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

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

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

А базовый языковой файл — вот таким образом:

```
#: 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"
```

Типовые *настройки компонента* Weblate	
Маска файла	`po/*.po`
Одноязычный базовый файл	`po/en.po`
Шаблон для новых переводов	`po/messages.pot`
Формат файла	*PO-файл gettext (одноязычный)*

1.10.5 XLIFF

Формат на основе XML, созданный для стандартизации файлов перевода, но в конечном итоге ставший просто ещё одним из многочисленных стандартов в этой области.

Формат файла XML для обмена локализациями (XLIFF) обычно используется как двуязычный, но Weblate поддерживает его использование и в качестве одноязычного.

См.также:

Спецификация *формата файла XML для обмена локализациями (XLIFF)*

Состояния перевода

Изменено в версии 3.3: Weblate ignored the `state` attribute prior to the 3.3 release.

Атрибуту состояния `state` из XLIFF-файла частично соответствует состояние «На правку» в Weblate (строки в следующих состояниях считаются отмеченными на правку, если перевод на целевой язык уже присутствует в файле: `new`, `needs-translation`, `needs-adaptation`, `needs-l10n`). Если атрибут `state` отсутствует, а собственно перевод есть (существует элемент `<target>`), то строка считается переведённой.

Если строка перевода имеет атрибут `approved="yes"`, она также будет импортирована в Weblate в состоянии «Одобрено», все остальные строки будут импортированы в состоянии «Ожидает рецензирования» (что соответствует спецификации XLIFF).

При сохранении Weblate добавляет эти атрибуты только если они действительно необходимы:

- Атрибут `state` добавляется только в том случае, если строка помечена на правку.
- Атрибут `approved` добавляется только в том случае, если строка была отрецензирована.
- В остальных случаях эти атрибуты не добавляются, но если они присутствуют, то они будут обновлятся.

Таким образом, при использовании формата XLIFF настоятельно рекомендуется включить рецензирование в Weblate, чтобы видеть и изменять состояние одобрения строк.

Аналогично, при импорте таких файлов (в форме загрузки) необходимо выбрать пункт *Импортировать как переведённое* в выпадающем списке *Обработка строк, отмеченных «На правку»*.

См.также:

Выделенные рецензенты

Пробелы и переводы строк в файлах XLIFF

Как правило, тип или количество пробельных символов в XML-форматах не различаются. Если вы хотите их сохранить, вы должны добавить к строке флаг `xml:space="preserve"`.

Например:

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

Указание флагов перевода

С помощью атрибута `weblate-flags` вы можете указать дополнительные флаги перевода (смотрите раздел *Настройка поведения с помощью флагов*). Также Weblate понимает атрибуты `maxwidth` и `font` из спецификации XLIFF:

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

Из атрибута `font` извлекается название семейства шрифтов, а также размер и насыщенность шрифта. Приведённый выше пример показывает использование всех трёх параметров, хотя обязательным к указанию является только семейство шрифта. Любые пробельные символы в наименовании семейства шрифта преобразуются в подчёркивания, так что `Source Sans Pro` становится `Source_Sans_Pro`, поэтому при указании группы шрифтов (смотрите раздел *Управление шрифтами*) имейте это в виду.

Ключи строк

В качестве идентификаторов единиц перевода (*unit*) из XLIFF-файла Weblate использует значение атрибута `resname`, а если его нет, то `id` (совместно с тегом `file`, если есть).

То что именно аттрибут `resname` используется в качестве идентификатора, а не `id` вызвано тем, что для `resname` поддерживаются значения в виде, удобном для человека, что делает из более подходящими для отображения в интерфейсе Weblate. Значение `resname` должно быть уникальным для всего XLIFF-файла. Это требование Weblate, а не стандарта XLIFF, который не накладывает таких ограничений.

Типовые *настройки компонента* Weblate для двуязычного XLIFF	
Маска файла	`localizations/*.xliff`
Одноязычный базовый файл	*Пусто*
Шаблон для новых переводов	`localizations/en-US.xliff`
Формат файла	*Файл перевода XLIFF*

Типовые *настройки компонента* Weblate для одноязычного XLIFF	
Маска файла	`localizations/*.xliff`
Одноязычный базовый файл	`localizations/en-US.xliff`
Шаблон для новых переводов	`localizations/en-US.xliff`
Формат файла	*Файл перевода XLIFF*

См.также:

XLIFF в Википедии, XLIFF, Атрибут font в XLIFF 1.2, Атрибут maxwidth в XLIFF 1.2

1.10.6 Файлы свойств Java

Родной формат Java для переводов.

Файлы свойств Java обычно используются в качестве одноязычных переводов.

Weblate поддерживает варианты этого формата в кодировках ISO-8859-1, UTF-8 и UTF-16. Все они поддерживают сохранение всех символов юникода, просто в разных кодировках. В формате ISO-8859-1 используются юникодные экранирующие последовательности (например, `zkou\u0161ka`), остальные кодируют символы непосредственно в UTF-8 или UTF-16.

Примечание: Загрузка экранирующих последовательностей также работает в режиме UTF-8, поэтому, пожалуйста, будьте внимательны при выборе правильного набора кодировок в соответствии с потребностями вашего приложения.

Типовые *настройки компонента* Weblate	
Маска файла	`src/app/Bundle_*.properties`
Одноязычный базовый файл	`src/app/Bundle.properties`
Шаблон для новых переводов	*Пусто*
Формат файла	*Свойства Java (ISO 8859-1)*

См.также:

Свойства Java в Википедии, Файлы свойств Mozilla и Java, *Форматирование файла свойств Java*, *Очистка файлов перевода*

1.10.7 Файлы свойств GWT

Родной формат GWT для переводов.

Файлы свойств GWT обычно используются в качестве одноязычных переводов.

Типовые *настройки компонента* Weblate	
Маска файла	`src/app/Bundle_*.properties`
Одноязычный базовый файл	`src/app/Bundle.properties`
Шаблон для новых переводов	*Пусто*
Формат файла	*Свойства GWT*

См.также:

Руководство по локализации GWT, Пособие по интернализации GWT, Файлы свойств Mozilla и Java, *Форматирование файла свойств Java*, *Очистка файлов перевода*

1.10.8 Переводы в INI-файлах

Добавлено в версии 4.1.

Формат INI-файла для переводов.

Переводы в INI-файлах обычно используются в качестве одноязычных переводов.

Типовые *настройки компонента* Weblate	
Маска файла	`language/*.ini`
Одноязычный базовый файл	`language/en.ini`
Шаблон для новых переводов	*Пусто*
Формат файла	*INI-файл*

Примечание: Weblate извлекает ключи только из внутренних секций INI-файла. Если в вашем INI-файле нет секций, то, возможно, вместо INI вам стоит использовать формат переводов *Joomla* или *Java*.

См.также:

INI Files, *Файлы свойств Java*, *Переводы Joomla*, *Переводы в INI-файлах Inno Setup*

1.10.9 Переводы в INI-файлах Inno Setup

Добавлено в версии 4.1.

Формат INI-файлов Inno Setup для переводов.

Переводы в INI-файлах Inno Setup обычно используются в качестве одноязычных переводов.

Примечание: Единственным заметным отличием от *обычных переводов в INI-файлах* является поддержка заполнителей `%n` и `%t` для символов переноса строки и табуляции.

Типовые *настройки компонента* Weblate	
Маска файла	`language/*.islu`
Одноязычный базовый файл	`language/en.islu`
Шаблон для новых переводов	*Пусто*
Формат файла	*INI-файл Inno Setup*

Примечание: На текущий момент поддерживаются только юникодные файлы (`.islu`), файлы в кодировке ANSI (`.isl`) в настоящее время не поддерживаются.

См.также:

INI-файлы, *Переводы Joomla*, *Переводы в INI-файлах*

1.10.10 Переводы Joomla

Добавлено в версии 2.12.

Родной формат Joomla для переводов.

Переводы Joomla обычно используются в качестве одноязычных переводов.

Типовые *настройки компонента* Weblate	
Маска файла	`language/*/com_foobar.ini`
Одноязычный базовый файл	`language/en-GB/com_foobar.ini`
Шаблон для новых переводов	*Пусто*
Формат файла	*Языковой файл Joomla*

См.также:

Спецификация языковых файлов Joomla, Файлы свойств Mozilla и Java, *Переводы в INI-файлах*, *Переводы в INI-файлах Inno Setup*

1.10.11 Файлы .ts Qt Linguist

Формат перевода, используемый в приложениях на основе Qt.

Файлы Qt Linguist используются и как двуязычные, так и одноязычные переводы.

Типовые *настройки компонента* Weblate при использовании формата как двуязычного	
Маска файла	`i18n/app.*.ts`
Одноязычный базовый файл	*Пусто*
Шаблон для новых переводов	`i18n/app.ru.ts`
Формат файла	*Файл перевода Qt Linguist*

Типовые *настройки компонента* Weblate при использовании формата как одноязычного	
Маска файла	`i18n/app.*.ts`
Одноязычный базовый файл	`i18n/app.en.ts`
Шаблон для новых переводов	`i18n/app.en.ts`
Формат файла	*Файл перевода Qt Linguist*

См.также:

Руководство по Qt Linguist, Qt .ts, *Двуязычные и одноязычные форматы*

1.10.12 Строковые ресурсы Android

Специфичный для Android формат файла для перевода приложений.

Строковые ресурсы Android являются одноязычными. ref:*component-template* хранится в другом месте, нежели остальные строки — в `res/values/strings.xml`.

Типовые *настройки компонента* Weblate	
Маска файла	`res/values-*/strings.xml`
Одноязычный базовый файл	`res/values/strings.xml`
Шаблон для новых переводов	*Пусто*
Формат файла	*Строковый ресурс Android*

См.также:

Документация по строковым ресурсам Android, Строковые ресурсы Android

---

Примечание: Структуры строкового массива Android, *string-array*, в настоящее время не поддерживаются. Чтобы это обойти, вы можете разбить свои строковые массивы на части:

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

станет:

```
<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>
```

Элемент *string-array*, указывающий на элементы *string*, должен храниться в другом файле и не должен быть доступен для перевода.

Помочь в подготовке существующих файлов strings.xml и переводов вам может вот этот скрипт: https://gist.github.com/paour/11291062

1.10.13 Строки Apple iOS

Специфичный для Apple формат файла для перевода приложений, используемый для переводов приложений как для iOS, так и для iPhone'a/iPad'a.

Строки Apple iOS обычно используются в качестве двуязычных переводов.

Типовые *настройки компонента* Weblate	
Маска файла	`Resources/*.lproj/Localizable.strings`
Одноязычный базовый файл	`Resources/en.lproj/Localizable.strings` или `Resources/Base.lproj/Localizable.strings`
Шаблон для новых переводов	*Пусто*
Формат файла	*Строки iOS (UTF-8)*

См.также:

Документация Apple по «строковым файлам», Строки Mac OSX

1.10.14 Строки PHP

Переводы PHP обычно одноязычные, поэтому рекомендуется указывать базовый файл со строками на (как это чаще всего бывает) английском языке.

Пример файла:

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

Типовые *настройки компонента* Weblate	
Маска файла	`lang/*/texts.php`
Одноязычный базовый файл	`lang/en/texts.php`
Шаблон для новых переводов	`lang/en/texts.php`
Формат файла	*Строки PHP*

Строки Laravel PHP

Изменено в версии 4.1.

Файлы локализации Laravel PHP также поддерживают множественные формы:

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

См.также:

PHP, Локализация в Laravel

1.10.15 JSON-файлы

Добавлено в версии 2.0.

Изменено в версии 2.16: Начиная с Weblate версии 2.16 и translate-toolkit версии не ниже 2.2.4 также поддерживаются JSON-файлы с вложенной структурой.

Изменено в версии 4.3: Структура JSON-файла корректно сохраняется даже в сложных ситуациях (которые некорректно обрабатывались в предыдущих выпусках).

Формат JSON используется в основном для перевода приложений, написанных на JavaScript.

В настоящее время Weblate поддерживает несколько вариантов переводов в JSON:

- Простые файлы в формате ключ/значение, используемые, например, *vue-i18n* или *react-intl*.
- Файлы с вложенными ключами.
- *JSON -файлы i18next*
- *JSON-файлы go-i18n*
- *JSON-файлы WebExtension*
- *Файл ARB*

Переводы JSON обычно одноязычные, поэтому рекомендуется указывать базовый файл со строками на (как это чаще всего бывает) английском языке.

Пример файла:

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

Также поддерживается вложенная структура файлов (условия поддержки смотрите выше), такой файл может выглядеть следующим образом:

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

Подсказка: И формат *JSON-файл*, и *JSON-файл с вложенной структурой* могут обрабатывать одни и теже файлы.

Единственная разница между ними проявляется при добавлении новых строк через Weblate: варианта формата «со вложенной структурой» разбирает ключи вновь добавленных строк и вставляет их в файл в виде соответствующей структуры. Например, ключ `app.name` будет вставлен как:

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

Типовые *настройки компонента* Weblate	
Маска файла	`langs/translation-*.json`
Одноязычный базовый файл	`langs/translation-en.json`
Шаблон для новых переводов	*Пусто*
Формат файла	*Файл вложенной структуры JSON*

См.также:

JSON, *Настройка вывода в JSON*, *Очистка файлов перевода*,

1.10.16 JSON -файлы i18next

Изменено в версии 2.17: Начиная с Weblate версии 2.17 и translate-toolkit версии не ниже 2.2.5 также поддерживаются JSON-файлы i18next с формами множественного числа.

i18next — это фреймворк интернационализации, написанный на и для JavaScript. Weblate поддерживает его файлы локализации с такими возможностями, как формы множественного числа.

Переводы i18next обычно одноязычные, поэтому рекомендуется указывать базовый файл со строками на (как это чаще всего бывает) английском языке.

Примечание: Weblate поддерживает формат JSON-файла i18next версии v3. Версии v2 и v1 в основном совместимы, за исключением того, как они работают с формами множественного числа.

Пример файла:

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

Типовые *настройки компонента* Weblate	
Маска файла	`langs/*.json`
Одноязычный базовый файл	`langs/en.json`
Шаблон для новых переводов	*Пусто*
Формат файла	*JSON-файл i18next*

См.также:

JSON, Формат JSON i18next, *Настройка вывода в JSON*, *Очистка файлов перевода*

1.10.17 JSON-файлы go-i18n

Добавлено в версии 4.1.

Переводы go-i18n обычно одноязычные, поэтому рекомендуется указывать базовый файл со строками на (как это чаще всего бывает) английском языке.

Примечание: Weblate поддерживает формат JSON-файла go-i18n версии v1, для плоских JSON-файлов используйте формат *JSON-файлы*. Формат v2 с хэшами в настоящее время не поддерживается.

Типовые *настройки компонента* Weblate	
Маска файла	`langs/*.json`
Одноязычный базовый файл	`langs/en.json`
Шаблон для новых переводов	*Пусто*
Формат файла	*JSON -файл go-i18n*

См.также:

JSON, go-i18n, *Настройка вывода в JSON*, *Очистка файлов перевода*,

1.10.18 Файл ARB

Добавлено в версии 4.1.

Переводы ARB обычно одноязычные, поэтому рекомендуется указывать базовый файл со строками на (как это чаще всего бывает) английском языке.

Типовые *настройки компонента* Weblate	
Маска файла	`lib/l10n/intl_*.arb`
Одноязычный базовый файл	`lib/l10n/intl_en.arb`
Шаблон для новых переводов	*Пусто*
Формат файла	*Файл ARB*

См.также:

JSON, Спецификация пакета ресурсов приложения, Интернационализация приложений Flutter, *Настройка вывода в JSON*, *Очистка файлов перевода*

1.10.19 JSON-файлы WebExtension

Добавлено в версии 2.16: Поддерживаются с версии Weblate 2.16 и translate-toolkit версии не ниже 2.2.4.

Формат файла, используемый при переводе расширений для Mozilla Firefox или Google Chromium.

Примечание: Хотя этот формат называется JSON, его спецификация позволяет включать комментарии, которые не являются частью спецификации JSON. В настоящее время Weblate не поддерживает файл с комментариями.

Пример файла:

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

(continues on next page)

(продолжение с предыдущей страницы)

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

Типовые *настройки компонента* Weblate	
Маска файла	`_locales/*/messages.json`
Одноязычный базовый файл	`_locales/en/messages.json`
Шаблон для новых переводов	*Пусто*
Формат файла	*JSON-файл WebExtension*

См.также:

JSON, Google chrome.i18n, Интернационализация расширений Mozilla

1.10.20 .XML-файлы ресурсов

Добавлено в версии 2.3.

В .XML-файле ресурса (.resx) применяется формат одноязычного XML-файла, используемый в приложениях Microsoft .NET. Он взаимозаменяем с форматом .resw, который использует синтаксис, идентичный синтаксису .resx.

Типовые *настройки компонента* Weblate	
Маска файла	`Resources/Language.*.resx`
Одноязычный базовый файл	`Resources/Language.resx`
Шаблон для новых переводов	*Пусто*
Формат файла	*Файл ресурсов .NET*

См.также:

.NET Resource files (.resx), *Очистка файлов перевода*

1.10.21 Файлы CSV

Добавлено в версии 2.4.

Файлы CSV обычно содержат простой список исходных строк и переводов. Weblate поддерживает следующие форматы оных:

- Файлы с заголовком, определяющим поля в самом файле (`location` — расположение, `source` — исходная строка, `target` — перевод, `ID`, `fuzzy` — «на правку», `context` — контекст, `translator_comments` — комментарий переводчика, `developer_comments` — комментарий разработчика). Это рекомендуемый подход, так как при его использовании меньше вероятность ошибок. Выберите *CSV-файл* в качестве формата.
- 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.
- Не забывайте определить *одноязычный базовый файл*, если ваши файлы одноязычные (см. *Двуязычные и одноязычные форматы*).

Предупреждение: Формат CSV в настоящее время автоматически определяет диалект файла CSV. В некоторых случаях автоматическое определение может не сработать, и вы получите смешанные результаты. Особенно это касается файлов CSV с символами новых строк внутри значений. В качестве обходного пути рекомендуется не использовать кавычки.

Пример файла:

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

Типовые *настройки компонента* Weblate для двуязычного CSV	
Маска файла	`locale/*.csv`
Одноязычный базовый файл	*Пусто*
Шаблон для новых переводов	`locale/en.csv`
Формат файла	*Файл CSV*

Типовые *настройки компонента* Weblate для одноязычного CSV	
Маска файла	`locale/*.csv`
Одноязычный базовый файл	`locale/en.csv`
Шаблон для новых переводов	`locale/en.csv`
Формат файла	*Простой CSV-файл*

См.также:

CSV

1.10.22 Файлы YAML

Добавлено в версии 2.9.

Простые файлы YAML со строковыми ключами и значениями. Также Weblate извлекает строки из списков и словарей.

Пример файла YAML:

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

Типовые *настройки компонента* Weblate	
Маска файла	`translations/messages.*.yml`
Одноязычный базовый файл	`translations/messages.en.yml`
Шаблон для новых переводов	*Пусто*
Формат файла	*Файл YAML*

См.также:

YAML, *Файлы Ruby YAML*

1.10.23 Файлы Ruby YAML

Добавлено в версии 2.9.

Файлы YAML локализации Ruby с языком в качестве корневого узла.

Пример файла YAML локализации Ruby:

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

Типовые *настройки компонента* Weblate	
Маска файла	`translations/messages.*.yml`
Одноязычный базовый файл	`translations/messages.en.yml`
Шаблон для новых переводов	*Пусто*
Формат файла	*Файл Ruby YAML*

См.также:

YAML, *Файлы YAML*

1.10.24 Файлы DTD

Добавлено в версии 2.18.

Пример DTD-файла:

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

Типовые *настройки компонента* Weblate	
Маска файла	`locale/*.dtd`
Одноязычный базовый файл	`locale/en.dtd`
Шаблон для новых переводов	*Пусто*
Формат файла	*Файл DTD*

См.также:

Формат DTD Mozilla

1.10.25 Плоские XML-файлы

Добавлено в версии 3.9.

Пример плоского XML-файла:

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

Типовые *настройки компонента* Weblate	
Маска файла	`locale/*.xml`
Одноязычный базовый файл	`locale/en.xml`
Шаблон для новых переводов	*Пусто*
Формат файла	*Плоский XML-файл*

См.также:

Плоские XML-файлы

1.10.26 Файлы ресурсов Windows

Изменено в версии 4.1: Поддержка файлов ресурсов Windows была переписана.

Примечание: Поддержка этого формата в настоящее время находится в бета-режиме, приветствуются обратная связь от тестирования.

Пример файла ресурсов Windows:

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT

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

Типовые *настройки компонента* Weblate	
Маска файла	`lang/*.rc`
Одноязычный базовый файл	`lang/en-US.rc`
Шаблон для новых переводов	`lang/en-US.rc`
Формат файла	*Файл RC*

См.также:

Файлы ресурсов Windows

1.10.27 Файлы метаданных магазина приложений

Добавлено в версии 3.5.

Метаданные, используемые для публикации приложений в различных магазинах приложений, могут быть переведены. В настоящее время совместимы следующие инструменты:

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

Метаданные состоят из нескольких текстовых файлов, которые Weblate представляет для перевода в виде отдельных строк.

Типовые *настройки компонента* Weblate	
Маска файла	`fastlane/android/metadata/*`
Одноязычный базовый файл	`fastlane/android/metadata/en-US`
Шаблон для новых переводов	`fastlane/android/metadata/en-US`
Формат файла	*Файлы метаданных магазина приложений*

Подсказка: Если вы не хотите переводить определённые строки (например, историю изменений), пометьте их «только для чтения» (*read-only*, см. *Настройка поведения с помощью флагов*). Это также можно автоматизировать с помощью надстройки «*Массовая правка*».

1.10.28 Файлы субтитров

Добавлено в версии 3.7.

Weblate может переводить различные файлы субтитров:

- Файл субтитров SubRip (`*.srt`)
- Файл субтитров MicroDVD (`*.sub`)
- Файл субтитров Advanced SubStation Alpha (`*.ass`)
- Файл субтитров SubStation Alpha (`*.ssa`)

Типовые *настройки компонента* Weblate	
Маска файла	`path/*.srt`
Одноязычный базовый файл	`path/en.srt`
Шаблон для новых переводов	`path/en.srt`
Формат файла	*Файл субтитров SubRip*

См.также:

Субтитры

1.10.29 Excel Open XML

Добавлено в версии 3.2.

Можно импортировать и экспортировать файлы Excel Open XML (.xlsx).

Загружая для перевода файлы XLSX, имейте в виду, что учитывается только активный рабочий лист, и должны быть хотя бы колонки с названием `source` (содержащая исходную строку) и `target` (содержащая перевод). Дополнительно может быть колонка с названием `context` (содержащая путь к контексту строки перевода). Если вы используете скачанный XLSX для экспорта переводов в рабочую книгу Excel, вы уже получите файл в правильном формате.

1.10.30 Файлы HTML

Добавлено в версии 4.1.

Примечание: Поддержка этого формата в настоящее время находится в бета-режиме, приветствуются обратная связь от тестирования.

Переводимое содержимое извлекается из HTML-файлов и предлагается для перевода.

См.также:

HTML

1.10.31 Формат OpenDocument

Добавлено в версии 4.1.

Примечание: Поддержка этого формата в настоящее время находится в бета-режиме, приветствуются обратная связь от тестирования.

Переводимое содержимое извлекается из файлов OpenDocument и предлагается для перевода.

См.также:

Формат OpenDocument

1.10.32 Формат IDML

Добавлено в версии 4.1.

Примечание: Поддержка этого формата в настоящее время находится в бета-режиме, приветствуются обратная связь от тестирования.

Переводимое содержимое извлекается из файлов языка разметки Adobe InDesign и предлагается для перевода.

1.10.33 TermBase eXchange format

Добавлено в версии 4.5.

TBX — это XML-формат для обмена терминологическими данными.

Типовые *настройки компонента* Weblate	
Маска файла	`tbx/*.tbx`
Одноязычный базовый файл	*Пусто*
Шаблон для новых переводов	*Пусто*
Формат файла	*TermBase eXchange file*

См.также:

TBX в английской Википедии, TBX, *Глоссарий*

1.10.34 Другие

Большинство форматов, которые поддерживают сериализацию и которые поддерживает translate-toolkit, могут поддерживаться с лёгкостью, но они (пока) ни как не тестировались. В большинстве случаев, чтобы скрыть различия в поведении различных хранилищ translate-toolkit, в Weblate приходится добавлять тонкую прослойку.

См.также:

Форматы файлов, относящиеся к переводу

1.10.35 Строки только для чтения

Добавлено в версии 3.10.

Строки из файлов перевода, доступные только для чтения, будут включены, но не могут быть отредактированы в Weblate. Поддержка этой возможности уже встроена в некоторые форматы (*XLIFF* и *Строковые ресурсы Android*), а в других может быть сэмулирована добавлением к строке флага `read-only` — смотрите раздел *Настройка поведения с помощью флагов*.

1.11 Интеграция с системой контроля версий

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

1.11.1 Доступ к репозиториям

Репозиторий системы контроля версий, который вы хотите использовать, должен быть доступен для Weblate. При использовании общедоступного репозитория вам просто надо указать правильный URL-адрес (например, `https://github.com/WeblateOrg/weblate.git`), но для частных репозиториев или для URL-адресов отправки изменений настройка будет более сложной и потребует аутентификации.

Доступ к репозиториям из 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 *Репозиторий исходного кода* and *URL для отправки в репозиторий* using the SSH protocol (for example `git@github.com:WeblateOrg/weblate.git`).

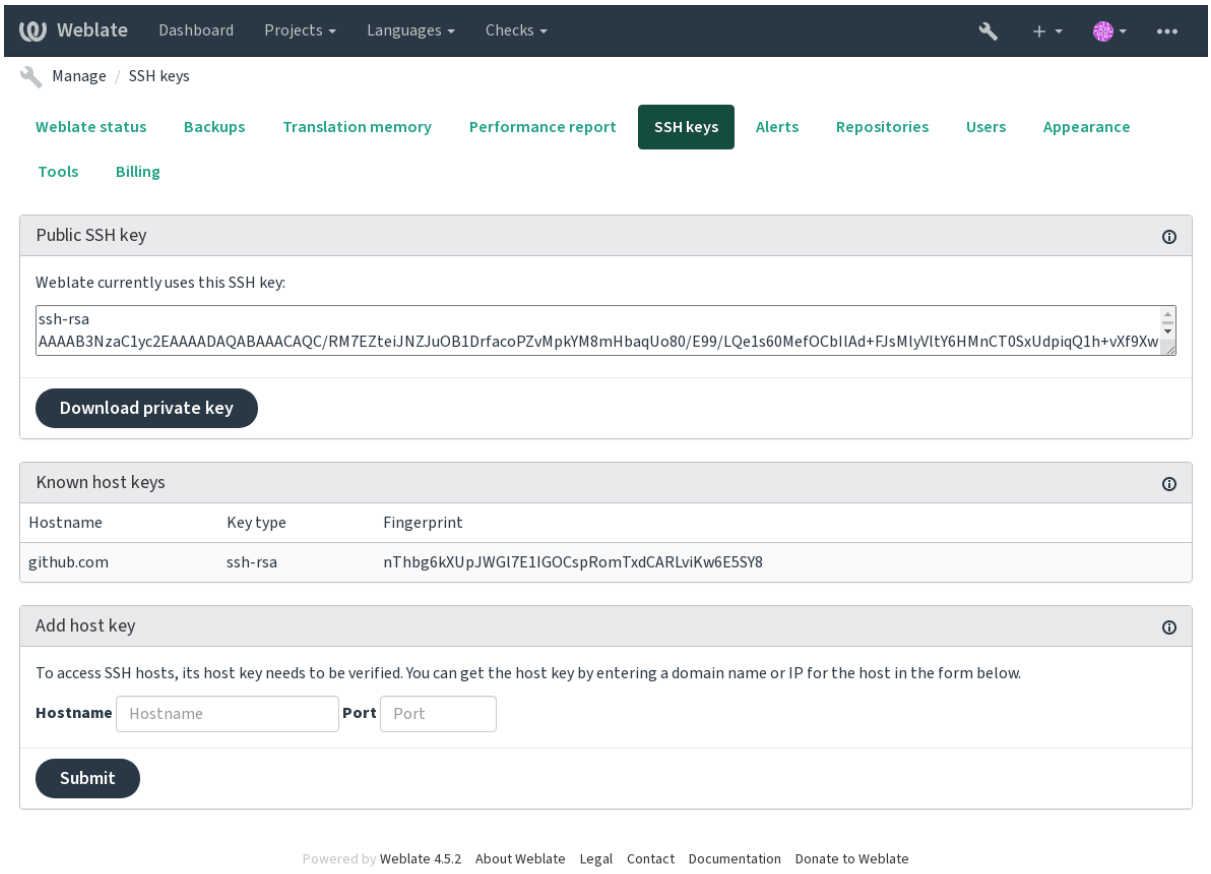
Репозитории по SSH

Наиболее часто используемый метод доступа к частным репозиториям — доступ по SSH. Чтобы дать Weblate такой доступ к вышестоящему репозиторию, авторизуйте его публичный SSH-ключ (смотрите раздел *SSH-ключ Weblate*).

> **Предупреждение:** On GitHub, each key can only be used once, see *Репозитории GitHub* and *Доступ к репозиториям из Hosted Weblate*.

Weblate также сохраняет отпечаток ключа сервера при первом подключении и не сможет подключиться к нему, если в дальнейшем ключ будет изменён (смотрите раздел *Проверка SSH-ключей сервера*).

В случае, если его будет необходимо изменить, это можно сделать из интерфейса администратора Weblate:



SSH-ключ Weblate

Открытый ключ Weblate виден всем пользователям, просматривающим страницу *О Weblate*.

Администраторы могут сгенерировать или посмотреть публичный ключ, который в настоящее время использует Weblate (из блока *SSH-ключи*) на странице входа в интерфейс администратора.

> **Примечание:** Соответствующий закрытый SSH-ключ в настоящее время не может быть запаролен, поэтому убедитесь, что он хорошо защищён.

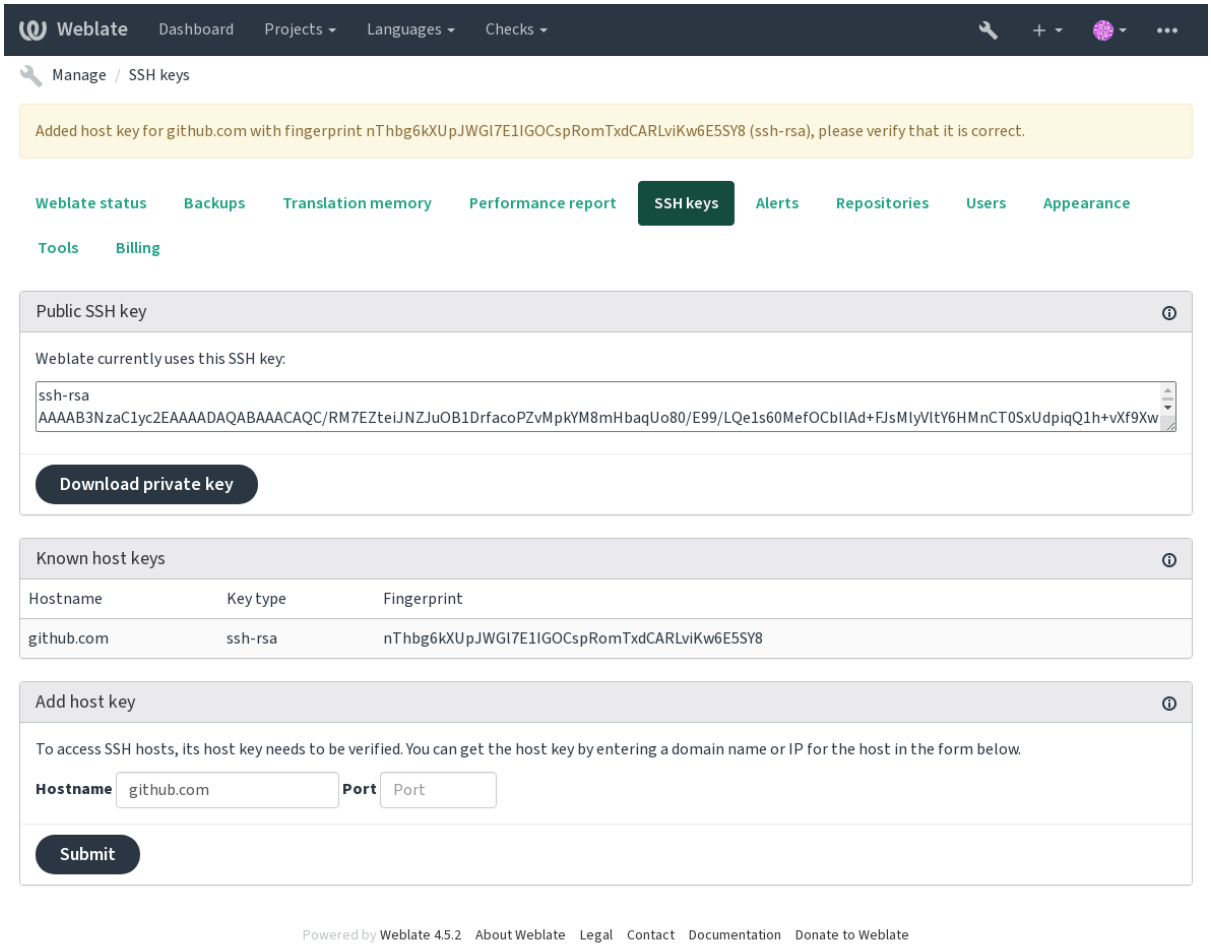
> **Подсказка:** Сделайте резервную копию сгенерированного закрытого SSH-ключа Weblate.

Проверка SSH-ключей сервера

Weblate автоматически сохраняет для дальнейшего использования SSH-ключи серверов при первом обращении к ним.

В случае, если вы хотите, подтвердить подлинность отпечатков ключей до первого подключения к репозиторию, добавьте SSH-ключ тех серверов, к которым вы собираетесь подключаться, через пункт *Добавить ключ хоста*, в том же разделе интерфейса администратора. Введите имя хоста, к которому вы собираетесь получить доступ (например, `gitlab.com`), и нажмите *Подтвердить*. После этого вы сможете убедится, что отпечаток этого ключа соответствует тому серверу, который вы добавляете.

Добавленные ключи и их отпечатки будут отображаться в сообщении о подтверждении:



Репозитории GitHub

Доступ по SSH возможен (смотрите раздел. *Репозитории по SSH*), но в случае, если вам необходимо получить доступ к более чем одному репозиторию, вы столкнётесь с ограничением GitHub'а по разрешённому использованию ключей SSH (поскольку один ключ может быть использован только один раз).

В случае, если *ветка для отправки* не задана, проект форкается и изменения отправляются через форк. В случае если она задана, изменения отправляются в вышестоящий репозиторий и выбранную ветку.

Для небольших проектов используйте HTTPS-аутентификацию с персональным токеном доступа и вашей GitHub'овской учётной записью, смотрите подробности в документации GitHub'а: Создание токена доступа для использования из командной строки.

Для более крупных конфигураций обычно лучше создать для Weblate специального пользователя, назначить ему открытый SSH-ключ, сгенерированный Weblate (смотрите раздел *SSH-ключ Weblate*), и предоставить

ему доступ ко всем репозиториям, которые вы собираетесь переводить. Этот подход также используется и наоблачном хостинге Weblate, для него на GitHub есть специальный пользователь *weblate*.

См.также:

Доступ к репозиториям из Hosted Weblate

Внутренние URL-адреса 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.

Предупреждение: 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.

Зачем это нужно:

- Экономия дискового пространства на сервере: хранится только одна копия репозитория.
- Ускорение обновлений: обновляется только один репозиторий.
- Существует только один экспортируемый репозиторий с переводами Weblate (смотрите раздел *Экспортер Git*).
- Некоторые надстройки могут работать с несколькими компонентами, совместно использующими один репозиторий, например, «*Уплотнение Git-коммитов*».

Репозитории по HTTPS

Для доступа к защищённым HTTPS-репозиториям включите в URL-адрес имя пользователя и пароль. Не волнуйтесь, Weblate обрежет эту информацию, когда адрес будет показываться пользователям (если им вообще будет разрешено просматривать адрес репозитория).

Например, URL-адрес GitHub с добавленной аутентификацией будет выглядеть примерно следующим образом: `https://пользователь:ваш_токен_доступа@github.com/WeblateOrg/weblate.git`.

Примечание: Если имя вашего пользователя или пароль содержат специальные символы, то они должны быть закодированы для использования в URL, например `https://user%40example.com:%24password%23@bitbucket.org/`….

Использование прокси

Если вам необходимо получить доступ к репозиториям системы контроля версий по HTTP/HTTPS через прокси-сервер, настройте СКВ на его использование.

Это можно сделать с помощью установки переменных окружения `http_proxy`, `https_proxy` и `all_proxy` (как описано в документации cURL) или принудительно включив его использование в конфигурации системы контроля версий, например:

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

Примечание: Настройка прокси должна выполняться под тем пользователем, от имени которого работает Weblate (смотрите также раздел *Права доступа к файлам*) и с `HOME=$DATA_DIR/home` (смотрите `DATA_DIR`), иначе процесс Git, запускаемый Weblate, не будет его использовать.

См.также:

Man-страница cURL, Документация команды git config

1.11.2 Git

См.также:

Для получения информации о том, как получить доступ к различным типам репозиториев, смотрите раздел *Доступ к репозиториям*.

Git с принудительной отправкой (force push)

Этот вариант ведёт себя точно так же, как и Git, с той лишь разницей, что он всегда выполняет принудительную отправку. Он предназначен только для тех случаев, когда вы используете отдельный репозиторий для переводов.

Предупреждение: Используйте с осторожностью, поскольку это может привести к потере коммитов в вашем репозитории.

Настройка конфигурации Git'а

Weblate выполняет все команды системы контроля версий с установкой `HOME=$DATA_DIR/home` (смотрите `DATA_DIR`), поэтому изменение пользовательских настроек нужно выполнять в `DATA_DIR/home/.git`.

Удалённые помощники Git

Вы также можете использовать удалённых помощников Git, чтобы получить поддержку других систем контроля версий, но будьте готовы самостоятельно отлаживать проблемы, которые это может вызвать.

В настоящее время в отдельных GitHub'овских репозиторияхдоступны помощники для Bazaar и Mercurial: git-remote-hg и git-remote-bzr. Скачайте их вручную и поместите куда-нибудь в пути поиска (например, в `~/bin`). Убедитесь, что у вас установлены соответствующие системы контроля версий.

После их установки ссылки на такие удалённые репозитории можно будет указывать в качестве репозитория в Weblate.

Для клонирования проекта `gnuhello` из Launchpad с помощью Bazaar используйте:

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

Для клонирования репозитория `hello` с selenic.com через Mercurial используйте:

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

Предупреждение: Неудобство использования удалённых помощников Git'а заключается , на примере Mercurial, в том, что удалённый помощник иногда создаёт новую оконечную фиксацию (tip) при отправке изменений обратно.

1.11.3 GitHub

Добавлено в версии 2.3.

Это просто добавляет тонкий слой логики поверх *Git* с помощью API GitHub'a, что позволяет отправлять изменения в переводе в виде запросов на извлечение вместо того, чтобы выполнять `push` непосредственно в репозиторий.

Git отправляет изменения непосредственно в репозиторий, в то время как *GitHub* создаёт запросы на извлечение. Для простого доступа к Git-репозиториям последние не нужны.

См.также:

Отправка изменений из Weblate'a

Отправка изменений в GitHub в виде запросов на извлечение

Если вы не хотите отправлять переводы непосредственно в репозиторий GitHub'a, их можно отправить в виде одного или нескольких запросов на извлечение.

Чтобы это заработало, нужно будет настроить учётные данные API.

См.также:

GITHUB_USERNAME, *GITHUB_TOKEN*, *GITHUB_CREDENTIALS*

1.11.4 GitLab

Добавлено в версии 3.9.

Это просто добавляет тонкий слой логики поверх *Git* с помощью API GitLab'a, что позволяет отправлять изменения в переводе в виде запросов на слияние вместо того, чтобы выполнять `push` непосредственно в репозиторий.

Нет необходимости использовать его для доступа к Git-репозиториям, обычный *Git* работает так же, единственная разница заключается в том, как выполняется отправка изменений в репозиторий. С помощью *Git* изменения отправляются непосредственно в репозиторий, в то время как *GitLab* создаёт запрос на слияние.

См.также:

Отправка изменений из Weblate'a

Отправка изменений в GitLab в виде запросов на слияние

Если вы не хотите отправлять переводы непосредственно в репозиторий GitLab, их можно отправить в виде одного или нескольких запросов на слияние.

Чтобы это заработало, нужно будет настроить учётные данные API.

См.также:

GITLAB_USERNAME, *GITLAB_TOKEN*, *GITLAB_CREDENTIALS*

1.11.5 Pagure

Добавлено в версии 4.3.2.

Это просто добавляет тонкий слой логики поверх *Git* с помощью API Pagure, что позволяет отправлять изменения в переводе в виде запросов на слияние вместо того, чтобы выполнять `push` непосредственно в репозиторий.

Нет необходимости использовать его для доступа к Git-репозиториям, обычный *Git* работает так же, единственная разница заключается в том, как выполняется отправка изменений в репозиторий. С помощью *Git* изменения отправляются непосредственно в репозиторий, в то время как *Pagure* создаёт запрос на слияние.

См.также:

Отправка изменений из Weblate'а

Отправка изменений в Pagure в виде запросов на слияние

Если вы не хотите отправлять переводы непосредственно в репозиторий Pagure, их можно отправить в виде одного или нескольких запросов на слияние.

Чтобы это заработало, нужно будет настроить учётные данные API.

См.также:

`PAGURE_USERNAME`, `PAGURE_TOKEN`, `PAGURE_CREDENTIALS`

1.11.6 Gerrit

Добавлено в версии 2.2.

Это просто добавляет тонкий слой логики поверх *Git* с помощью git-review , что позволяет отправлять изменения в переводе в виде запросов на рецензирование Gerrit вместо того, чтобы выполнять `push` непосредственно в репозиторий.

В документации Gerrit подробно описаны те настройки, которые необходимо сделать для использования таких репозиториев.

1.11.7 Mercurial

Добавлено в версии 2.1.

Mercurial — это ещё одна система контроля версий, которую вы можете использовать в Weblate напрямую.

Примечание: Weblate должен работать с любой версией Mercurial'а, но иногда в его интерфейсе командной строки происходят несовместимые изменения, которые нарушают его интеграцию с Weblate.

См.также:

Для получения информации о том, как получить доступ к различным типам репозиториев, смотрите раздел *Доступ к репозиториям*.

1.11.8 Subversion

Добавлено в версии 2.8.

Для взаимодействия с хранилищем subversion Weblate использует команду git-svn. Это Perl-скрипт, который позволяет использовать subversion из клиента Git'a, позволяя пользователям поддерживать полный клон внутреннего хранилища и коммитить в него локально.

Примечание: Weblate автоматически пытается определить компоновку хранилища Subversion — он поддерживает как прямые URL-адреса для веток, так и хранилища со стандартной компоновкой (каталоги branches/, tags/ и trunk/). Подробнее об этом можно прочитать в документации к git-svn. Если ваше хранилище не придерживается стандартной компоновки и вы столкнулись с ошибками, попробуйте включить в URL-адрес хранилища имя ветки, а саму ветку оставить пустой.

Изменено в версии 2.19: Before this, only repositories using the standard layout were supported.

Учётные данные 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
```

См.также:

`DATA_DIR`

1.11.9 Локальные файлы

Добавлено в версии 3.8.

Weblate также может работать без удалённой системы контроля версий. Первоначальные переводы импортируются путём их загрузки. Позже вы можете заменить отдельные файлы через загрузку файлов или добавить строки перевода непосредственно из Weblate (в настоящее время этот функционал доступен только для одноязычных переводов).

Weblate создаст свой внутренний Git-репозиторий, в котором будут отслеживаться все изменения. В случае, если вы позже решите использовать систему контроля версий для хранения переводов, у вас уже будет готовый репозиторий внутри Weblate, на основе которого можно будет создать свой собственный.

1.12 REST API Weblate'a

Добавлено в версии 2.6: REST API доступен с версии Weblate 2.6.

API доступен по адресу `/api/` и основан на фреймворке Django REST. Вы можете использовать его напрямую или через *Клиент Weblate*.

1.12.1 Аутентификация и общие параметры

API публичного проекта доступен без аутентификации, хотя количество неаутентифицированных запросов сильно ограничено (по умолчанию до 100 запросов в день), так что аутентификацию всё равно рекомендуется использовать. Аутентификация использует токен, который вы можете получить в своём профиле. Используйте его в заголовке `Authorization`:

ANY /

Описанные здесь общее поведение запросов для API, заголовки, коды состояния и параметры применяются ко всем конечным точкам.

Параметры запроса

- **format** – Формат ответа (переопределяет заголовок Accept). Возможные значения зависят от настройки REST-фреймворка, по умолчанию поддерживаются `json` и `api`. Последний предоставляет для API интерфейс веб-браузера.

Заголовки запроса

- Accept – тип содержимого ответа зависит от заголовка Accept
- Authorization – необязательный токен для аутентификации

Заголовки ответа

- Content-Type – зависит от заголовка Accept запроса
- Allow – список разрешённых для объекта методов HTTP

JSON -объект ответа

- **detail** (*string*) – verbose description of the result (for HTTP status codes other than 200 OK)
- **count** (*int*) – общее количество элементов для списков объектов
- **next** (*string*) – URL-адрес следующей страницы для списков объектов
- **previous** (*string*) – URL-адрес предыдущей страницы для списков объектов
- **results** (*array*) – результаты для списков объектов
- **url** (*string*) – URL-адрес для доступа к этому ресурсу с помощью API
- **web_url** (*string*) – URL-адрес для доступа к этому ресурсу через веб-браузер

Коды статусов

- 200 OK – при правильной обработке запроса
- 201 Created – when a new object was created successfully
- 204 No Content – when an object was created successfully
- 400 Bad Request – при отсутствии параметров формы
- 403 Forbidden – в случае отказа в доступе
- 429 Too Many Requests – при превышении количества запросов из-за ограничения на количество запросов

Примеры аутентификации

Пример запроса:

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

Пример ответа:

```
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/"
}
```

Пример CURL:

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

Примеры передачи параметров

Для метода POST параметры могут быть указаны либо в виде представления формы (*application/x-www-form-urlencoded*), либо в JSON (*application/json*).

Пример запроса формы:

```
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
```

Пример JSON запроса:

```
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"}
```

Пример CURL:

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

Пример CURL с JSON:

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

Ограничение частоты запросов API

Частота запросов к API ограничена; конфигурация по умолчанию ограничивает её до 100 запросов в день для анонимных пользователей и до 5000 запросов в час для аутентифицированных пользователей.

Это ограничение можно настроить в `settings.py`; подробнее о его настройке смотрите в разделе Throttling документации фреймворка Django REST.

О состоянии ограничения частоты запросов сообщается в следующих заголовках:

`X-RateLimit-Limit`	Ограничение на максимальное число выполняемых запросов
`X-RateLimit-Remaining`	Оставшийся лимит запросов
`X-RateLimit-Reset`	Количество секунд до сброса окна ограничения количества запросов

Изменено в версии 4.1: Добавлены заголовки состояния ограничения количества запросов.

См.также:

Ограничение частоты запросов, *Ограничение частоты запросов*

1.12.2 Точка входа в API

GET /api/

Корневая точка входа в API.

Пример запроса:

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

Пример ответа:

```
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/",
```

(continues on next page)

(продолжение с предыдущей страницы)

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

1.12.3 Пользователи

Добавлено в версии 4.0.

GET /api/users/

Возвращает список пользователей, если у вас есть разрешения на просмотр пользоватей для управления. Если их нет, то вы увидите только свои данные.

См.также:

Атрибуты объекта-пользователя описываются в разделе *GET /api/users/(str:username)/*.

POST /api/users/

Создаёт нового пользователя.

Параметры

- **username** (*string*) – Имя пользователя
- **full_name** (*string*) – Полное имя пользователя
- **email** (*string*) – Электронная почта пользователя
- **is_superuser** (*boolean*) – Является ли пользователь суперпользователем? (необязательный)
- **is_active** (*boolean*) – Активен ли пользователь? (необязательный)

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

Возвращает информацию о пользователях.

Параметры

- **username** (*string*) – Имя пользователя

JSON -объект ответа

- **username** (*string*) – имя пользователя
- **full_name** (*string*) – полное имя пользователя
- **email** (*string*) – электронная почта пользователя
- **is_superuser** (*boolean*) – является ли пользователь суперпользователем
- **is_active** (*boolean*) – активен ли пользователь
- **date_joined** (*string*) – дата создания пользователя
- **groups** (*array*) – ссылка на связанные группы; смотрите описание *GET /api/groups/(int:id)/*

Пример данных в 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,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "is_active": true,
    "date_joined": "2020-03-29T18:42:42.617681Z",
    "url": "http://example.com/api/users/exampleusername/",
    "statistics_url": "http://example.com/api/users/exampleusername/statistics/
↪"
}
```

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

Изменяет параметры пользователя.

Параметры

- **username** (*string*) – Имя пользователя

JSON -объект ответа

- **username** (*string*) – имя пользователя
- **full_name** (*string*) – полное имя пользователя
- **email** (*string*) – электронная почта пользователя
- **is_superuser** (*boolean*) – является ли пользователь суперпользователем
- **is_active** (*boolean*) – активен ли пользователь
- **date_joined** (*string*) – дата создания пользователя

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

Изменяет параметры пользователя.

Параметры

- **username** (*string*) – Имя пользователя

JSON -объект ответа

- **username** (*string*) – имя пользователя
- **full_name** (*string*) – полное имя пользователя
- **email** (*string*) – электронная почта пользователя
- **is_superuser** (*boolean*) – является ли пользователь суперпользователем
- **is_active** (*boolean*) – активен ли пользователь
- **date_joined** (*string*) – дата создания пользователя

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

Удаляет всю информацию о пользователе и помечает пользователя как неактивного.

Параметры

- **username** (*string*) – Имя пользователя

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

Связывает группы с пользователем.

Параметры

- **username** (*string*) – Имя пользователя

Параметры формы

- **string group_id** – Уникальный идентификатор группы

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

Список статистик пользователя.

Параметры

- **username** (*string*) – Имя пользователя

JSON -объект ответа

- **translated** (*int*) – Количество переводов пользователя
- **suggested** (*int*) – Количество предложений пользователя
- **uploaded** (*int*) – Количество загрузок пользователя
- **commented** (*int*) – Количество комментариев пользователя
- **languages** (*int*) – Количество языков, на которые пользователь может переводить

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

Список подписок пользователя.

Параметры

- **username** (*string*) – Имя пользователя

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

Связывает подписки с пользователем.

Параметры

- **username** (*string*) – Имя пользователя

JSON -объект запроса

- **notification** (*string*) – Название зарегистрированного уведомления
- **scope** (*int*) – Область действия уведомления из доступных вариантов
- **frequency** (*int*) – Выбор частоты уведомлений

GET /api/users/(**str:** *username*)**/notifications/**
int: *subscription_id***/** Возвращает связанную с пользователем подписку.

Параметры

- **username** (*string*) – Имя пользователя
- **subscription_id** (*int*) – Идентификатор зарегистрированного уведомления

PUT /api/users/(**str:** *username*)**/notifications/**
int: *subscription_id***/** Редактирует связанную с пользователем подписку.

Параметры

- **username** (*string*) – Имя пользователя
- **subscription_id** (*int*) – Идентификатор зарегистрированного уведомления

JSON -объект запроса

- **notification** (*string*) – Название зарегистрированного уведомления
- **scope** (*int*) – Область действия уведомления из доступных вариантов
- **frequency** (*int*) – Выбор частоты уведомлений

PATCH /api/users/(**str:** *username*)**/notifications/**
int: *subscription_id***/** Редактирует связанную с пользователем подписку.

Параметры

- **username** (*string*) – Имя пользователя
- **subscription_id** (*int*) – Идентификатор зарегистрированного уведомления

JSON -объект запроса

- **notification** (*string*) – Название зарегистрированного уведомления

- **scope** (*int*) – Область действия уведомления из доступных вариантов
- **frequency** (*int*) – Выбор частоты уведомлений

DELETE /api/users/(**str:** *username*)**/notifications/**
int: *subscription_id***/** Удаляет связанную с пользователем подписку.

Параметры

- **username** (*string*) – Имя пользователя
- **subscription_id** – Название зарегистрированного уведомления
- **subscription_id** – целое

1.12.4 Группы

Добавлено в версии 4.0.

GET /api/groups/
Возвращает список групп, если у вас есть разрешения на просмотр групп для управления. Если их нет, то вы можете видеть только те группы, в которые входит пользователь.

См.также:

Атрибуты объекта-группы описываются в разделе *GET /api/groups/(int:id)/*.

POST /api/groups/
Создаёт новую группу.

Параметры

- **name** (*string*) – Название группы
- **project_selection** (*int*) – Группа проекта, выбранная из предложенных вариантов
- **language_selection** (*int*) – Группа языков, выбранная из предложенных вариантов

GET /api/groups/(**int:** *id*)**/**
Возвращает информацию о группе.

Параметры

- **id** (*int*) – Идентификатор группы

JSON -объект ответа

- **name** (*string*) – название группы
- **project_selection** (*int*) – целое число, соответствующее группе проектов
- **language_selection** (*int*) – целое число, соответствующее группе языков
- **roles** (*array*) – ссылка на связанные роли; смотрите описание *GET /api/roles/(int:id)/*
- **projects** (*array*) – ссылка на связанные проекты; смотрите описание *GET /api/projects/(string:project)/*
- **components** (*array*) – ссылка на связанные компоненты; смотрите описание *GET /api/components/(string:project)/(string:component)/*
- **componentlist** (*array*) – ссылка на связанный список компонентов; смотрите описание *GET /api/component-lists/(str:slug)/*

Пример данных в JSON:

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

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

Изменяет параметры группы.

Параметры

- **id** (*int*) – Идентификатор группы

JSON -объект ответа

- **name** (*string*) – название группы
- **project_selection** (*int*) – целое число, соответствующее группе проектов
- **language_selection** (*int*) – целое число, соответствующее группе языков

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

Изменяет параметры группы.

Параметры

- **id** (*int*) – Идентификатор группы

JSON -объект ответа

- **name** (*string*) – название группы
- **project_selection** (*int*) – целое число, соответствующее группе проектов
- **language_selection** (*int*) – целое число, соответствующее группе языков

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

Удаляет группу.

Параметры

- **id** (*int*) – Идентификатор группы

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

Связывает роли с группой.

Параметры

- **id** (*int*) – Идентификатор группы

Параметры формы

- **string role_id** – Уникальный идентификатор роли

POST /api/groups/(**int:** *id*)**/components/**
Связывает компоненты с группой.

Параметры

- **id** (*int*) – Идентификатор группы

Параметры формы

- **string component_id** – Уникальный идентификатор компонента

DELETE /api/groups/(**int:** *id*)**/components/**
int: *component_id* Удаляет компонент из группы.

Параметры

- **id** (*int*) – Идентификатор группы
- **component_id** (*int*) – Уникальный идентификатор компонента

POST /api/groups/(**int:** *id*)**/projects/**
Связывает проекты с группой.

Параметры

- **id** (*int*) – Идентификатор группы

Параметры формы

- **string project_id** – Уникальный идентификатор проекта

DELETE /api/groups/(**int:** *id*)**/projects/**
int: *project_id* Удаляет проект из группы.

Параметры

- **id** (*int*) – Идентификатор группы
- **project_id** (*int*) – Уникальный идентификатор проекта

POST /api/groups/(**int:** *id*)**/languages/**
Связывает языки с группой.

Параметры

- **id** (*int*) – Идентификатор группы

Параметры формы

- **string language_code** – Уникальный код языка

DELETE /api/groups/(**int:** *id*)**/languages/**
string: *language_code* Удаляет язык из группы.

Параметры

- **id** (*int*) – Идентификатор группы
- **language_code** (*string*) – Уникальный код языка

POST /api/groups/(**int:** *id*)**/componentlists/**
Связывает списки компонентов с группой.

Параметры

- **id** (*int*) – Идентификатор группы

Параметры формы

- **string component_list_id** – Уникальный идентификатор списка компонентов

DELETE /api/groups/(**int:** *id*)**/componentlists/**
int: *component_list_id* Удаляет список компонентов из группы.

Параметры

- **id** (*int*) – Идентификатор группы
- **component_list_id** (*int*) – Уникальный идентификатор списка компонентов

1.12.5 Роли

GET /api/roles/

Возвращает список всех ролей, связанных с пользователем. Если пользователь является суперпользователем, то возвращается список всех существующих ролей.

См.также:

Атрибуты объекта-роли описываются в разделе *GET /api/roles/(int:id)/*.

POST /api/roles/

Создаёт новую роль.

Параметры

- **name** (*string*) – Название роли
- **permissions** (*array*) – Список кодовых названий разрешений

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

Возвращает информацию о роли.

Параметры

- **id** (*int*) – Идентификатор роли

JSON -объект ответа

- **name** (*string*) – Название роли
- **permissions** (*array*) – список кодовых названий разрешений

Пример данных в JSON:

```
{
    "name": "Access repository",
    "permissions": [
        "vcs.access",
        "vcs.view"
    ],
    "url": "http://example.com/api/roles/1/",
}
```

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

Изменяет параметры роли.

Параметры

- **id** (*int*) – Идентификатор роли

JSON -объект ответа

- **name** (*string*) – Название роли
- **permissions** (*array*) – список кодовых названий разрешений

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

Изменяет параметры роли.

Параметры

- **id** (*int*) – Идентификатор роли

JSON -объект ответа

- **name** (*string*) – Название роли
- **permissions** (*array*) – список кодовых названий разрешений

DELETE /api/roles/(int: *id*)/
Удаляет роль.

Параметры

- **id** (*int*) – Идентификатор роли

1.12.6 Языки

GET /api/languages/
Возвращает список всех языков.

См.также:

Атрибуты объекта-языка описываются в разделе *GET /api/languages/(string:language)/*.

POST /api/languages/
Создаёт новый язык.

Параметры

- **code** (*string*) – Название языка
- **name** (*string*) – Название языка
- **direction** (*string*) – Направление текста
- **plural** (*object*) – Формула множественного числа языка и количество форм множественных чисел

GET /api/languages/(string: *language*)/
Возвращает информацию о языке.

Параметры

- **language** (*string*) – Код языка

JSON -объект ответа

- **code** (*string*) – Код языка
- **direction** (*string*) – Направление текста
- **plural** (*object*) – Объект информации о формах множественного числа языка
- **aliases** (*array*) – Массив псевдонимов языка

Пример данных в JSON:

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

(continues on next page)

(продолжение с предыдущей страницы)

```
        "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*)/

Изменяет параметры языка.

Параметры

- **language** (*string*) – Код языка

JSON -объект запроса

- **name** (*string*) – Название языка
- **direction** (*string*) – Направление текста
- **plural** (*object*) – Данные по формам множественного числа языка

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

Изменяет параметры языка.

Параметры

- **language** (*string*) – Код языка

JSON -объект запроса

- **name** (*string*) – Название языка
- **direction** (*string*) – Направление текста
- **plural** (*object*) – Данные по формам множественного числа языка

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

Deletes the language.

Параметры

- **language** (*string*) – Код языка

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

Возвращает статистику по языку.

Параметры

- **language** (*string*) – Код языка

JSON -объект ответа

- **total** (*int*) – общее количество строк
- **total_words** (*int*) – общее количество слов
- **last_change** (*timestamp*) – последние изменения в языке
- **recent_changes** (*int*) – общее количество изменений
- **translated** (*int*) – количество переведённых строк
- **translated_percent** (*float*) – процент переведённых строк
- **translated_words** (*int*) – количество переведённых слов
- **translated_words_percent** (*int*) – процент переведённых слов
- **translated_chars** (*int*) – количество переведённых символов

- **translated_chars_percent** (*int*) – процент переведённых символов
- **total_chars** (*int*) – общее количество символов
- **fuzzy** (*int*) – количество неточных (отмеченных «на правку») переводов
- **fuzzy_percent** (*int*) – процент неточных (отмеченных «на правку») переводов
- **failing** (*int*) – количество строк с неудачными проверками
- **failing** – процент строк с неудачными проверками

1.12.7 Проекты

GET /api/projects/

Возвращает список всех проектов.

См.также:

Атрибуты объекта-проекта описываются в разделе *GET /api/projects/(string:project)/*.

POST /api/projects/

Добавлено в версии 3.9.

Создаёт новый проект.

Параметры

- **name** (*string*) – Название проекта
- **slug** (*string*) – Плашка проекта
- **web** (*string*) – Веб-сайт проекта

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

Возвращает информацию о проекте.

Параметры

- **project** (*string*) – URL-плашка проекта

JSON -объект ответа

- **name** (*string*) – название проекта
- **slug** (*string*) – плашка проекта
- **web** (*string*) – веб-сайт проекта
- **components_list_url** (*string*) – URL-адрес списка компонентов; смотрите описание *GET /api/projects/(string:project)/components/*
- **repository_url** (*string*) – URL-адрес статуса репозитория; смотрите описание *GET /api/projects/(string:project)/repository/*
- **changes_list_url** (*string*) – URL-адрес списка изменений; смотрите описание *GET /api/projects/(string:project)/changes/*
- **translation_review** (*boolean*) – *Включить рецензии*
- **source_review** (*boolean*) – *Включить рецензии исходных строк*
- **set_language_team** (*boolean*) – *Задать заголовок «Language-Team»*
- **enable_hooks** (*boolean*) – *Включить обработчики*
- **instructions** (*string*) – *Инструкции для переводчиков*
- **language_aliases** (*string*) – *Псевдонимы языка*

Пример данных в 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*)/

Добавлено в версии 4.3.

Вносит изменения в проект с помощью PATCH-запроса.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

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

Добавлено в версии 4.3.

Вносит изменения в проект с помощью PUT-запроса.

Параметры

- **project** (*string*) – URL-плашка проекта

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

Добавлено в версии 3.9.

Удаляет проект.

Параметры

- **project** (*string*) – URL-плашка проекта

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

Возвращает список изменений проекта. По сути, эта точка входа работает так же и принимает те же самые параметры, что и *GET /api/changes/*, только для проекта.

Параметры

- **project** (*string*) – URL-плашка проекта

JSON -объект ответа

- **results** (*array*) – массив объектов компонентов; смотрите описание *GET /api/changes/(int:id)/*

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

Возвращает информацию о состоянии репозитория системы контроля версий. Эта конечная точка содержит только общую сводку по всем репозиториям проекта. Для получения более подробной информации о состоянии используйте конечную точку *GET /api/components/(string:project)/(string:component)/repository/*.

Параметры

- **project** (*string*) – URL-плашка проекта

JSON -объект ответа

- **needs_commit** (*boolean*) – есть ли какие-нибудь отложенные изменения для коммита
- **needs_merge** (*boolean*) – есть ли какие-нибудь изменения в вышестоящем репозитории для объединения
- **needs_push** (*boolean*) – есть ли какие-нибудь локальные изменения для отправки

Пример данных в JSON:

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

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

Выполняет заданную операцию на репозитории системы контроля версий.

Параметры

- **project** (*string*) – URL-плашка проекта

JSON -объект запроса

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

JSON -объект ответа

- **result** (*boolean*) – результат операции

Пример CURL:

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

Пример JSON запроса:

```
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"}
```

Пример ответа в 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/

Возвращает список компонентов перевода в указанном проекте.

Параметры

- **project** (*string*) – URL-плашка проекта

JSON -объект ответа

- **results** (*array*) – массив объектов компонентов; смотрите описание *GET /api/components/(string:project)/(string:component)/*

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

Добавлено в версии 3.9.

Изменено в версии 4.3: The `zipfile` and `docfile` parameters are now accepted for VCS-less components, see *Локальные файлы*.

Создаёт компоненты перевода в указанном проекте.

Подсказка: Use *Внутренние URL-адреса Weblate* when creating multiple components from a single VCS repository.

Примечание: Большая часть процесса создания компонента происходит асинхронно, в фоновой задаче. Посмотрите значение атрибута `task_url` вновь созданного компонента и следите за ходом выполнения через него.

Параметры

- **project** (*string*) – URL-плашка проекта

Параметры формы

- **file zipfile** – ZIP-файл, который будет загружен в Weblate для инициализации переводов
- **file docfile** – Документ для перевода

JSON -объект ответа

- **result** (*object*) – Созданный объект компонента; смотрите описание *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*.

CURL form request example:

```
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/
```

CURL JSON request example:

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

(continues on next page)

(продолжение с предыдущей страницы)

```
    }' \
    -H "Content-Type: application/json" \
    -H "Authorization: Token TOKEN" \
    http://example.com/api/projects/hello/components/
```

Пример JSON запроса:

```
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",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "slug": "weblate",
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
    "new_base": "",
    "vcs": "git"
}
```

Пример ответа в 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",
            "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/",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "web_url": "http://example.com/projects/hello/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
    "new_base": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
}
```

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

Возвращает разбитую на страницы статистику по всем языкам проекта.

Добавлено в версии 3.8.

Параметры

- **project** (*string*) – URL-плашка проекта

JSON -объект ответа

- **results** (*array*) – массив объектов статистики перевода
- **language** (*string*) – название языка
- **code** (*string*) – код языка
- **total** (*int*) – общее количество строк
- **translated** (*int*) – количество переведённых строк
- **translated_percent** (*float*) – процент переведённых строк
- **total_words** (*int*) – общее количество слов
- **translated_words** (*int*) – количество переведённых слов
- **words_percent** (*float*) – процент переведённых слов

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

Возвращает статистику по проекту.

Добавлено в версии 3.8.

Параметры

- **project** (*string*) – URL-плашка проекта

JSON -объект ответа

- **total** (*int*) – общее количество строк
- **translated** (*int*) – количество переведённых строк
- **translated_percent** (*float*) – процент переведённых строк
- **total_words** (*int*) – общее количество слов
- **translated_words** (*int*) – количество переведённых слов
- **words_percent** (*float*) – процент переведённых слов

1.12.8 Компоненты

GET /api/components/

Возвращает список компонентов перевода.

См.также:

Атрибуты объекта-компонента описываются в разделе *GET /api/components/(string:project)/(string:component)/*.

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

Возвращает информацию о компоненте перевода.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект ответа

- **project** (*object*) – проект перевода; смотрите описание *GET /api/projects/(string:project)/*
- **name** (*string*) – *Название компонента*
- **slug** (*string*) – *Плашка компонента*
- **vcs** (*string*) – *Система контроля версий*
- **repo** (*string*) – *Репозиторий исходного кода*
- **git_export** (*string*) – *URL экспортированного репозитория*
- **branch** (*string*) – *Ветка репозитория*
- **push_branch** (*string*) – *Ветка для отправки*
- **filemask** (*string*) – *Маска файла*
- **template** (*string*) – *Одноязычный базовый файл*
- **edit_template** (*string*) – *Правка базового файла*
- **intermediate** (*string*) – *Файл промежуточного языка*
- **new_base** (*string*) – *Шаблон для новых переводов*
- **file_format** (*string*) – *Формат файла*
- **license** (*string*) – *Лицензия перевода*
- **agreement** (*string*) – *Соглашение участника*
- **new_lang** (*string*) – *Добавление нового перевода*
- **language_code_style** (*string*) – *Стиль кода языка*
- **source_language** (*object*) – объект исходного языка; смотрите описание *GET /api/languages/(string:language)/*
- **push** (*string*) – *URL для отправки в репозиторий*
- **check_flags** (*string*) – *Флаги перевода*
- **priority** (*string*) – *Приоритет*
- **enforced_checks** (*string*) – *Принудительные проверки*
- **restricted** (*string*) – *Ограниченный доступ*
- **repoweb** (*string*) – *Просмотрщик репозитория*

- **report_source_bugs** (*string*) – *Адрес для сообщения об ошибке в исходной строке*
- **merge_style** (*string*) – *Стиль слияния*
- **commit_message** (*string*) – *Сообщения коммита, добавления, удаления, слияния и сообщения надстроек*
- **add_message** (*string*) – *Сообщения коммита, добавления, удаления, слияния и сообщения надстроек*
- **delete_message** (*string*) – *Сообщения коммита, добавления, удаления, слияния и сообщения надстроек*
- **merge_message** (*string*) – *Сообщения коммита, добавления, удаления, слияния и сообщения надстроек*
- **addon_message** (*string*) – *Сообщения коммита, добавления, удаления, слияния и сообщения надстроек*
- **allow_translation_propagation** (*string*) – *Разрешить распространение перевода*
- **enable_suggestions** (*string*) – *Включить предложения*
- **suggestion_voting** (*string*) – *Голосование по предложению*
- **suggestion_autoaccept** (*string*) – *Автоматическое принятие предложений*
- **push_on_commit** (*string*) – *Отправлять при коммите*
- **commit_pending_age** (*string*) – *Возраст изменений для коммита*
- **auto_lock_error** (*string*) – *Блокировать при ошибке*
- **language_regex** (*string*) – *Языковой фильтр*
- **variant_regex** (*string*) – *Регулярное выражение вариантов*
- **repository_url** (*string*) – URL-адрес статуса репозитория; смотрите описание `GET /api/components/(string:project)/(string:component)/repository/`
- **translations_url** (*string*) – URL-адрес списка переводов; смотрите описание `GET /api/components/(string:project)/(string:component)/translations/`
- **lock_url** (*string*) – URL-адрес статуса блокировки; смотрите описание `GET /api/components/(string:project)/(string:component)/lock/`
- **changes_list_url** (*string*) – URL-адрес списка изменений; смотрите описание `GET /api/components/(string:project)/(string:component)/changes/`
- **task_url** (*string*) – URL фоновой задачи (если такая есть); см. `GET /api/tasks/(str:uuid)/`

Пример данных в JSON:

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

(continues on next page)

(продолжение с предыдущей страницы)

```
        "name": "Hello",
        "slug": "hello",
        "source_language": {
            "code": "en",
            "direction": "ltr",
            "name": "English",
            "url": "http://example.com/api/languages/en/",
            "web_url": "http://example.com/languages/en/"
        },
        "url": "http://example.com/api/projects/hello/",
        "web": "https://weblate.org/",
        "web_url": "http://example.com/projects/hello/"
    },
    "source_language": {
        "code": "en",
        "direction": "ltr",
        "name": "English",
        "url": "http://example.com/api/languages/en/",
        "web_url": "http://example.com/languages/en/"
    },
    "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*/ Вносит изменения в компонент с помощью PATCH-запроса.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **source_language** (*string*) – Код исходного языка проекта (необязательный)

JSON -объект запроса

- **name** (*string*) – название компонента
- **slug** (*string*) – плашка компонента
- **repo** (*string*) – URL-адрес репозитория системы контроля версий

Пример CURL:

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

Пример JSON запроса:

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

(continues on next page)

(продолжение с предыдущей страницы)

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

Пример ответа в 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",
            "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*/ Вносит изменения в компонент с помощью PATCH-запроса.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект запроса

- **branch** (*string*) – Ветка репозитория системы контроля версий
- **file_format** (*string*) – формат файла переводов
- **filemask** (*string*) – маска файлов перевода в репозитории
- **name** (*string*) – название компонента

- **slug** (*string*) – плашка компонента
- **repo** (*string*) – URL-адрес репозитория системы контроля версий
- **template** (*string*) – базовый файл для одноязычных переводов
- **new_base** (*string*) – базовый файл для добавления новых переводов
- **vcs** (*string*) – система контроля версий

DELETE /api/components/(**string:** *project*)**/**
string: *component***/** Добавлено в версии 3.9.

Удаляет компонент.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

GET /api/components/(**string:** *project*)**/**
string: *component***/changes/** Возвращает список изменений компонента. По сути, эта точка входа работает так же и принимает те же самые параметры, что и *GET /api/changes/*, только для компонента.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект ответа

- **results** (*array*) – массив объектов компонентов; смотрите описание *GET /api/changes/(int:id)/*

GET /api/components/(**string:** *project*)**/**
string: *component***/screenshots/** Возвращает список снимков экрана компонента.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект ответа

- **results** (*array*) – массив снимков экрана компонента; смотрите описание *GET /api/screenshots/(int:id)/*

GET /api/components/(**string:** *project*)**/**
string: *component***/lock/** Возвращает состояние блокировки компонента.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект ответа

- **locked** (*boolean*) – заблокирован ли компонент для обновлений

Пример данных в JSON:

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

POST /api/components/(string: *project*)/
string: *component*/lock/ Устанавливает статус блокировки компонента.

Ответ такой же, как и у *GET /api/components/(string:project)/(string:component)/lock/*.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект запроса

- **lock** – Логическое значение, стоит ли блокировка или нет.

Пример CURL:

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

Пример JSON запроса:

```
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}
```

Пример ответа в JSON:

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

{"locked":true}
```

GET /api/components/(string: *project*)/
string: *component*/repository/ Возвращает информацию о состоянии репозитория системы контроля версий.

Ответ такой же, как и у *GET /api/projects/(string:project)/repository/*.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект ответа

- **needs_commit** (*boolean*) – есть ли какие-нибудь отложенные изменения для коммита
- **needs_merge** (*boolean*) – есть ли какие-нибудь изменения в вышестоящем репозитории для объединения

- **needs_push** (*boolean*) – есть ли какие-нибудь локальные изменения для отправки
- **remote_commit** (*string*) – Информация о коммите из удалённого репозитория
- **status** (*string*) – Статус репозитория системы контроля версий по её же данным
- **merge_failure** – Текст, описывающий причину неудачи объединения или `null`, если её нет

POST /api/components/(string: *project*)/(string: *component*)/repository/ Выполняет заданную операцию на репозитории системы контроля версий.

Документацию смотрите в описании *POST /api/projects/(string:project)/repository/*.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект запроса

- **operation** (*string*) – Выполняемые операции: одна из `push`, `pull`, `commit`, `reset` или `cleanup`

JSON -объект ответа

- **result** (*boolean*) – результат операции

Пример CURL:

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

Пример JSON запроса:

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

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

Пример ответа в 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/ Скачивает базовый файл для одноязычных переводов.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

GET /api/components/(string: *project*)/
string: *component*/new_template/ Скачивает файл шаблона для новых переводов.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

GET /api/components/(string: *project*)/
string: *component*/translations/ Возвращает список объектов перевода в указанном компоненте.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект ответа

- **results** (*array*) – массив объектов перевода; смотрите описание *GET /api/translations/(string:project)/(string:component)/(string:language)/*

POST /api/components/(string: *project*)/
string: *component*/translations/ Создаёт новый перевод в указанном компоненте.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект запроса

- **language_code** (*string*) – код языка перевода; смотрите описание *GET /api/languages/(string:language)/*

JSON -объект ответа

- **result** (*object*) – новый созданный объект перевода

Пример CURL:

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

Пример JSON запроса:

```
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"}
```

Пример ответа в JSON:

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

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

GET /api/components/(string: *project*)/(string: *component*)/statistics/ Возвращает разбитую на страницы статистику по всем переводам компонента.

Добавлено в версии 2.7.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект ответа

- **results** (*array*) – массив объектов статистики перевода; смотрите описание *GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/*

GET /api/components/(string: *project*)/(string: *component*)/links/ Возвращает проекты, связанные с компонентом.

Добавлено в версии 4.5.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

JSON -объект ответа

- **projects** (*array*) – связанные проекты; смотрите описание *GET /api/projects/(string:project)/*

POST /api/components/(string: *project*)/string: *component*/links/ Связывает проект с компонентом.

Добавлено в версии 4.5.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента

Параметры формы

- **string project_slug** – Плашка проекта

DELETE /api/components/(string: *project*)/string: *component*/links/string: *project_slug*/ Удаляет связь проекта с компонентом.

Добавлено в версии 4.5.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **project_slug** (*string*) – Плашка удаляемого проекта

1.12.9 Переводы

GET /api/translations/
Возвращает список переводов.

См.также:

Атрибуты объекта-перевода описываются в разделе *GET /api/translations/(string:project)/(string:component)/(string:language)/*.

GET /api/translations/(string: *project*)/string: *component*/string: *language*/ Возвращает информацию о переводе.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

JSON -объект ответа

- **component** (*object*) – объект компонента; смотрите описание *GET /api/components/(string:project)/(string:component)/*
- **failing_checks** (*int*) – количество строк с неудачными проверками
- **failing_checks_percent** (*float*) – процент строк с неудачными проверками
- **failing_checks_words** (*int*) – количество слов с неудачными проверками

- **filename** (*string*) – имя файла перевода
- **fuzzy** (*int*) – количество неточных (отмеченных «на правку») переводов
- **fuzzy_percent** (*float*) – процент неточных (отмеченных «на правку») переводов
- **fuzzy_words** (*int*) – количество слов в неточных (отмеченных «на правку») строках
- **have_comment** (*int*) – количество строк с комментариями
- **have_suggestion** (*int*) – количество строк с комментариями
- **is_template** (*boolean*) – использует ли перевод одноязычные файлы
- **language** (*object*) – объект исходного языка; смотрите описание *GET /api/languages/(string:language)/*
- **language_code** (*string*) – код языка, используемый в репозитории; он может отличаться от кода языка в объекте языка
- **last_author** (*string*) – имя последнего автора
- **last_change** (*timestamp*) – метка времени последнего изменения
- **revision** (*string*) – хэш ревизии файла
- **share_url** (*string*) – URL-адрес для принятия участия в переводе, ведущий на вводную страницу
- **total** (*int*) – общее количество строк
- **total_words** (*int*) – общее количество слов
- **translate_url** (*string*) – URL-адрес для перевода
- **translated** (*int*) – количество переведённых строк
- **translated_percent** (*float*) – процент переведённых строк
- **translated_words** (*int*) – количество переведённых слов
- **repository_url** (*string*) – URL-адрес статуса репозитория; смотрите описание *GET /api/translations/(string:project)/(string:component)/(string:language)/repository/*
- **file_url** (*string*) – URL-адрес объекта файла; смотрите описание *GET /api/translations/(string:project)/(string:component)/(string:language)/file/*
- **changes_list_url** (*string*) – URL-адрес списка изменений; смотрите описание *GET /api/translations/(string:project)/(string:component)/(string:language)/changes/*
- **units_list_url** (*string*) – URL-адрес списка строк; смотрите описание *GET /api/translations/(string:project)/(string:component)/(string:language)/units/*

Пример данных в JSON:

```
{
    "component": {
        "branch": "main",
        "file_format": "po",
        "filemask": "po/*.po",
        "git_export": "",
        "license": "",
        "license_url": "",
        "name": "Weblate",
```

(continues on next page)

(продолжение с предыдущей страницы)


```
        "new_base": "",
        "project": {
            "name": "Hello",
            "slug": "hello",
            "source_language": {
                "code": "en",
                "direction": "ltr",
                "name": "English",
                "url": "http://example.com/api/languages/en/",
                "web_url": "http://example.com/languages/en/"
            },
            "url": "http://example.com/api/projects/hello/",
            "web": "https://weblate.org/",
            "web_url": "http://example.com/projects/hello/"
        },
        "repo": "file:///home/nijel/work/weblate-hello",
        "slug": "weblate",
        "template": "",
        "url": "http://example.com/api/components/hello/weblate/",
        "vcs": "git",
        "web_url": "http://example.com/projects/hello/weblate/"
    },
    "failing_checks": 3,
    "failing_checks_percent": 75.0,
    "failing_checks_words": 11,
    "filename": "po/cs.po",
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "fuzzy_words": 0,
    "have_comment": 0,
    "have_suggestion": 0,
    "is_template": false,
    "language": {
        "code": "cs",
        "direction": "ltr",
        "name": "Czech",
        "url": "http://example.com/api/languages/cs/",
        "web_url": "http://example.com/languages/cs/"
    },
    "language_code": "cs",
    "last_author": "Weblate Admin",
    "last_change": "2016-03-07T10:20:05.499",
    "revision": "7ddfafe6daaf57fc8654cc852ea6be212b015792",
    "share_url": "http://example.com/engage/hello/cs/",
    "total": 4,
    "total_words": 15,
    "translate_url": "http://example.com/translate/hello/weblate/cs/",
    "translated": 4,
    "translated_percent": 100.0,
    "translated_words": 15,
    "url": "http://example.com/api/translations/hello/weblate/cs/",
    "web_url": "http://example.com/projects/hello/weblate/cs/"
}
```


DELETE /api/translations/(string: *project*)**/**
string: *component*/**string:** *language*/ Добавлено в версии 3.9.

Удаляет перевод.

Параметры

- **project** (*string*) – URL-плашка проекта

- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

GET /api/translations/(**string:** *project*)**/**
string: *component***/string:** *language***/changes/** Возвращает список изменений перевода. По сути, эта точка входа работает так же и принимает те же самые параметры, что и *GET /api/changes/* , только для перевода.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

JSON -объект ответа

- **results** (*array*) – массив объектов компонентов; смотрите описание *GET /api/changes/(int:id)/*

GET /api/translations/(**string:** *project*)**/**
string: *component***/string:** *language***/units/** Возвращает список единиц перевода.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода
- **q** (*string*) – Строка поискового запроса (необязательный, см. *Поиск*)

JSON -объект ответа

- **results** (*array*) – массив объектов компонента; смотрите описание *GET /api/units/(int:id)/*

POST /api/translations/(**string:** *project*)**/**
string: *component***/string:** *language***/units/** Добавляет новую одноязычную единицу перевода.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

JSON -объект запроса

- **key** (*string*) – Название единицы перевода
- **value** (*string*) – Значение единицы перевода

См.также:

Управление строками, adding-new-strings

POST /api/translations/(**string:** *project*)**/**
string: *component***/string:** *language***/autotranslate/** Запускает автоматический перевод.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

JSON -объект запроса

- **mode** (*string*) – Режим автоматического перевода

- **filter_type** (*string*) – Тип фильтра автоматического перевода
- **auto_source** (*string*) – Источник автоматического перевода
- **component** (*string*) – Добавлять переводы этого компонента в общую память переводов проекта, дабы получить к ним доступ из других компонентов.
- **engines** (*string*) – Механизмы машинного перевода
- **threshold** (*string*) – Порог оценки

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 *Скачивание переводов*).

Примечание: Эта конечная точка API использует отличающуюся от остального API логику вывода, так как она работает не с данными, а с целым файлом. Отличается логика при установке допустимого параметра `format` — без него вы получите файл перевода, хранящийся в системе контроля версий.

Параметры запроса

- **format** – Используемый формат файла; если не указан, преобразование формата не происходит; поддерживаемые форматы файлов: `po`, `mo`, `xliff`, `xliff11`, `tbx`, `csv`, `xlsx`, `json`, `aresource` и `strings`

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

POST /api/translations/(string: *project*)/
string: *component*/string: *language*/file/ Загружает новый файл с переводами.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

Параметры формы

- **string conflicts** – Как поступать с конфликтами (`ignore`, `replace-translated` или `replace-approved`)
- **file file** – Загруженный файл
- **string email** – E-mail автора
- **string author** – Имя автора
- **string method** – Метод загрузки (`translate`, `approve`, `suggest`, `fuzzy`, `replace`, `source` или `add`), смотрите раздел *Способы импорта*
- **string fuzzy** – Что далать с неточными, отмеченными на правку, переводами (*пусто*, `process` или `approve`)

Пример 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/ Возвращает информацию о состоянии репозитория системы контроля версий.

Ответ такой же, как и у *GET /api/components/(string:project)/(string:component)/repository/*.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

POST /api/translations/(string: *project*)/string: *component*/string: *language*/repository/ Выполняет заданную операцию на репозитории системы контроля версий.

Документацию смотрите в описании *POST /api/projects/(string:project)/repository/*.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

JSON -объект запроса

- **operation** (*string*) – Выполняемые операции: одна из `push`, `pull`, `commit`, `reset` или `cleanup`

JSON -объект ответа

- **result** (*boolean*) – результат операции

GET /api/translations/(string: *project*)/string: *component*/string: *language*/statistics/ Возвращает подробную статистику по переводам.

Добавлено в версии 2.7.

Параметры

- **project** (*string*) – URL-плашка проекта
- **component** (*string*) – URL-плашка компонента
- **language** (*string*) – Код языка перевода

JSON -объект ответа

- **code** (*string*) – код языка
- **failing** (*int*) – количество неудачных проверок
- **failing_percent** (*float*) – процент неудачных проверок
- **fuzzy** (*int*) – количество неточных (отмеченных «на правку») переводов
- **fuzzy_percent** (*float*) – процент неточных (отмеченных «на правку») переводов
- **total_words** (*int*) – общее количество слов
- **translated_words** (*int*) – количество переведённых слов
- **last_author** (*string*) – имя последнего автора
- **last_change** (*timestamp*) – дата последнего изменения

- **name** (*string*) – название языка
- **total** (*int*) – общее количество строк
- **translated** (*int*) – количество переведённых строк
- **translated_percent** (*float*) – процент переведённых строк
- **url** (*string*) – URL-адрес для доступа к переводу (URL-адрес вводной страницы)
- **url_translate** (*string*) – URL-адрес для доступа к переводу (реальный URL-адрес перевода)

1.12.10 Единицы перевода

Единица перевода (*unit*) — это единичный фрагмент перевода, сопоставляющий исходную строку и соответствующий ей перевод, а также содержащий некоторые связанные с ними метаданные. Этот термин унаследован от Translate Toolkit и XLIFF.

Добавлено в версии 2.10.

GET /api/units/

Возвращает список единиц перевода.

См.также:

Атрибуты объекта единицы перевода описываются в разделе *GET /api/units/(int:id)/*.

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

Изменено в версии 4.3: `target` и `source` теперь являются массивами, чтобы корректно обрабатывать строки со множественным числом.

Возвращает информацию о единице перевода.

Параметры

- **id** (*int*) – Идентификатор единицы перевода

JSON -объект ответа

- **translation** (*string*) – URL-адрес связанного объекта перевода
- **source** (*array*) – исходная строка
- **previous_source** (*string*) – предыдущая исходная строка, использованная для поиска нечёткого соответствия
- **target** (*array*) – целевая строка
- **id_hash** (*string*) – уникальный идентификатор единицы перевода
- **content_hash** (*string*) – уникальный идентификатор исходной строки
- **location** (*string*) – местоположение единицы перевода в исходном коде
- **context** (*string*) – контекст единицы перевода
- **note** (*string*) – примечания к единице перевода
- **flags** (*string*) – флаги единицы перевода
- **state** (*int*) – состояние единицы перевода: 0 - не переведено, 10 - на правку, 20 - переведено, 30 - одобрено, 100 - только для чтения
- **fuzzy** (*boolean*) – является ли перевод единицы перевода неточным или отмеченным на правку
- **translated** (*boolean*) – переведена ли единица перевода
- **approved** (*boolean*) – одобрена ли единица перевода
- **position** (*int*) – позиция единицы перевода в файле перевода

- **has_suggestion** (*boolean*) – содержит ли единица перевода предложения
- **has_comment** (*boolean*) – содержит ли единица перевода комментарии
- **has_failing_check** (*boolean*) – имеются ли у единицы перевода неудачные проверки
- **num_words** (*int*) – количество слов в исходной строке
- **priority** (*int*) – приоритет перевода; по умолчанию равен 100
- **id** (*int*) – идентификатор единицы перевода
- **explanation** (*string*) – Пояснения для строки, доступные для данной единицы перевода, смотрите раздел *Дополнительная информация об исходных строках*
- **extra_flags** (*string*) – Дополнительные флаги строки, доступны на исходных единицах перевода, смотрите раздел *Настройка поведения с помощью флагов*
- **web_url** (*string*) – URL-адрес, по которому можно отредактировать единицу перевода
- **souce_unit** (*string*) – Ссылка на исходную единицу перевода; смотрите описание `GET /api/units/(int:id)/`

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

Добавлено в версии 4.3.

Выполняет частичное обновление единицы перевода.

Параметры

- **id** (*int*) – Идентификатор единицы перевода

JSON -объект запроса

- **state** (*int*) – состояние единицы перевода: 0 - не переведено, 10 - на правку, 20 - переведено, 30 - одобрено (требуется включить рецензирование, см. *Выделенные рецензенты*)
- **target** (*array*) – целевая строка
- **explanation** (*string*) – Пояснения для строки, доступные для данной единицы перевода, смотрите раздел *Дополнительная информация об исходных строках*
- **extra_flags** (*string*) – Дополнительные флаги строки, доступны на исходных единицах перевода, смотрите раздел *Настройка поведения с помощью флагов*

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

Добавлено в версии 4.3.

Выполняет полное обновление единицы перевода.

Параметры

- **id** (*int*) – Идентификатор единицы перевода

JSON -объект запроса

- **state** (*int*) – состояние единицы перевода: 0 - не переведено, 10 - на правку, 20 - переведено, 30 - одобрено (требуется включить рецензирование, см. *Выделенные рецензенты*)
- **target** (*array*) – целевая строка
- **explanation** (*string*) – Пояснения для строки, доступные для данной единицы перевода, смотрите раздел *Дополнительная информация об исходных строках*
- **extra_flags** (*string*) – Дополнительные флаги строки, доступны на исходных единицах перевода, смотрите раздел *Настройка поведения с помощью флагов*

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

Добавлено в версии 4.3.

Удаляет единицу перевода.

Параметры

- **id** (*int*) – Идентификатор единицы перевода

1.12.11 Изменения

Добавлено в версии 2.10.

GET /api/changes/

Изменено в версии 4.1: Фильтрация изменений была введена в выпуске 4.1.

Возвращает список изменений в переводе.

См.также:

Атрибуты объекта-изменения описываются в разделе *GET /api/changes/(int:id)/*.

Параметры запроса

- **user** (*string*) – Имя пользователя для фильтров
- **action** (*int*) – Действие для фильтра, может быть указано несколько раз
- **timestamp_after** (*timestamp*) – Временная метка в формате ISO 8601, изменения после которой нужно вернуть
- **timestamp_before** (*timestamp*) – Временная метка в формате ISO 8601, изменения до которой нужно вернуть

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

Возвращает информацию об изменении в переводе.

Параметры

- **id** (*int*) – Идентификатор изменения

JSON -объект ответа

- **unit** (*string*) – URL-адрес связанного объекта единицы перевода
- **translation** (*string*) – URL-адрес связанного объекта перевода
- **component** (*string*) – URL-адрес связанного объекта компонента
- **user** (*string*) – URL-адрес связанного объекта пользователя
- **author** (*string*) – URL-адрес связанного объекта автора
- **timestamp** (*timestamp*) – временная метка события
- **action** (*int*) – числовой идентификатор действия
- **action_name** (*string*) – текстовое описание действия
- **target** (*string*) – изменённый текст или информация о событии
- **id** (*int*) – идентификатор изменения

1.12.12 Снимки экрана

Добавлено в версии 2.14.

GET /api/screenshots/

Возвращает список снимков экрана для исходных строк.

См.также:

Атрибуты объекта снимка экрана описываются в разделе *GET /api/screenshots/(int:id)/*.

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

Возвращает информацию о снимке экрана.

Параметры

- **id** (*int*) – Идентификатор снимка экрана

JSON -объект ответа

- **name** (*string*) – название снимка экрана
- **component** (*string*) – URL-адрес связанного объекта компонента
- **file_url** (*string*) – URL-адрес для скачивания файла; смотрите описание *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ссылка на связанную информацию об исходной строке; смотрите описание *GET /api/units/(int:id)/*

GET /api/screenshots/(int: *id*)/file/

Скачивает изображение снимка экрана.

Параметры

- **id** (*int*) – Идентификатор снимка экрана

POST /api/screenshots/(int: *id*)/file/

Заменяет изображение снимка экрана.

Параметры

- **id** (*int*) – Идентификатор снимка экрана

Параметры формы

- **file image** – Загруженный файл

Пример 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/

Связывает исходную строку со снимком экрана.

Параметры

- **id** (*int*) – Идентификатор снимка экрана

Параметры формы

- **string unit_id** – Идентификатор единицы перевода

JSON -объект ответа

- **name** (*string*) – название снимка экрана
- **translation** (*string*) – URL-адрес связанного объекта перевода

- **file_url** (*string*) – URL-адрес для скачивания файла; смотрите описание *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ссылка на связанную информацию об исходной строке; смотрите описание *GET /api/units/(int:id)/*

DELETE /api/screenshots/(int: *id*)/units/
int: *unit_id* Удаляет связь исходной строки со снимком экрана.

Параметры

- **id** (*int*) – Идентификатор снимка экрана
- **unit_id** – ИД единицы перевода исходной строки

POST /api/screenshots/
Создаёт новый снимок экрана.

Параметры формы

- **file image** – Загруженный файл
- **string name** – Название снимка экрана
- **string project_slug** – Плашка проекта
- **string component_slug** – Плашка компонента
- **string language_code** – Код языка

JSON -объект ответа

- **name** (*string*) – название снимка экрана
- **component** (*string*) – URL-адрес связанного объекта компонента
- **file_url** (*string*) – URL-адрес для скачивания файла; смотрите описание *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ссылка на связанную информацию об исходной строке; смотрите описание *GET /api/units/(int:id)/*

PATCH /api/screenshots/(int: *id*)/
Редактирует часть информации о снимке экрана.

Параметры

- **id** (*int*) – Идентификатор снимка экрана

JSON -объект ответа

- **name** (*string*) – название снимка экрана
- **component** (*string*) – URL-адрес связанного объекта компонента
- **file_url** (*string*) – URL-адрес для скачивания файла; смотрите описание *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ссылка на связанную информацию об исходной строке; смотрите описание *GET /api/units/(int:id)/*

PUT /api/screenshots/(int: *id*)/
Редактирует всю информацию о снимке экрана.

Параметры

- **id** (*int*) – Идентификатор снимка экрана

JSON -объект ответа

- **name** (*string*) – название снимка экрана
- **component** (*string*) – URL-адрес связанного объекта компонента

- **file_url** (*string*) – URL-адрес для скачивания файла; смотрите описание *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ссылка на связанную информацию об исходной строке; смотрите описание *GET /api/units/(int:id)/*

DELETE /api/screenshots/(int: *id*)/
Удаляет снимок экрана.

Параметры

- **id** (*int*) – Идентификатор снимка экрана

1.12.13 Надстройки

Добавлено в версии 4.4.1.

GET /api/addons/
Возвращает список надстроек.

См.также:

Атрибуты объекта-надстройки, описываются в разделе *GET /api/units/(int:id)/*.

GET /api/addons/(int: *id*)/
Возвращает информацию о надстройке.

Параметры

- **id** (*int*) – ID надстройки

JSON -объект ответа

- **name** (*string*) – название надстройки
- **component** (*string*) – URL-адрес связанного объекта компонента
- **configuration** (*object*) – Необязательные настройки надстройки

POST /api/components/(string: *project*)/
string: *component*/addons/ Создаёт новую надстройку.

Параметры

- **project_slug** (*string*) – Плашка проекта
- **component_slug** (*string*) – Плашка компонента

JSON -объект запроса

- **name** (*string*) – название надстройки
- **configuration** (*object*) – Необязательные настройки надстройки

PATCH /api/addons/(int: *id*)/
Редактирует часть информации о надстройке.

Параметры

- **id** (*int*) – ID надстройки

JSON -объект ответа

- **configuration** (*object*) – Необязательные настройки надстройки

PUT /api/addons/(int: *id*)/
Редактирует всю информацию о надстройке.

Параметры

- **id** (*int*) – ID надстройки

JSON -объект ответа

- **configuration** (*object*) – Необязательные настройки надстройки

DELETE /api/addons/(int: *id*)/
Удаляет надстройку.

Параметры

- **id** (*int*) – ID надстройки

1.12.14 Списки компонентов

Добавлено в версии 4.0.

GET /api/component-lists/
Возвращает список списков компонентов.

См.также:

Атрибуты объекта списка компонентов описываются в разделе *GET /api/component-lists/(str:slug)/*.

GET /api/component-lists/(str: *slug*)/
Возвращает информацию о списке компонентов.

Параметры

- **slug** (*string*) – Плашка списка компонентов

JSON -объект ответа

- **name** (*string*) – название списка компонентов
- **slug** (*string*) – плашка списка компонентов
- **show_dashboard** (*boolean*) – показывать ли его на панели управления
- **components** (*array*) – ссылка на связанные компоненты; смотрите описание *GET /api/components/(string:project)/(string:component)/*
- **auto_assign** (*array*) – правила автоматического назначения

PUT /api/component-lists/(str: *slug*)/
Изменяет параметры списка компонентов.

Параметры

- **slug** (*string*) – Плашка списка компонентов

JSON -объект запроса

- **name** (*string*) – название списка компонентов
- **slug** (*string*) – плашка списка компонентов
- **show_dashboard** (*boolean*) – показывать ли его на панели управления

PATCH /api/component-lists/(str: *slug*)/
Изменяет параметры списка компонентов.

Параметры

- **slug** (*string*) – Плашка списка компонентов

JSON -объект запроса

- **name** (*string*) – название списка компонентов
- **slug** (*string*) – плашка списка компонентов
- **show_dashboard** (*boolean*) – показывать ли его на панели управления

DELETE /api/component-lists/(**str:** *slug*)**/**
Удаляет список компонентов.

Параметры

- **slug** (*string*) – Плашка списка компонентов

POST /api/component-lists/(**str:** *slug*)**/components/**
Связывает компонент со списком компонентов.

Параметры

- **slug** (*string*) – Плашка списка компонентов

Параметры формы

- **string component_id** – Идентификатор компонента

DELETE /api/component-lists/(**str:** *slug*)**/components/**
str: *component_slug* Отвязывает компонент от списка компонентов.

Параметры

- **slug** (*string*) – Плашка списка компонентов
- **component_slug** (*string*) – Плашка компонента

1.12.15 Глоссарий

Изменено в версии 4.5: Глоссарии теперь хранятся как обычные компоненты, переводы и строки, поэтому используйте для работы с ними соответствующие API.

1.12.16 Задачи

Добавлено в версии 4.4.

GET /api/tasks/
На данный момент получение списка задач недоступно.

GET /api/tasks/(**str:** *uuid*)**/**
Возвращает информацию о задаче

Параметры

- **uuid** (*string*) – UUID задачи

JSON -объект ответа

- **completed** (*boolean*) – Завершилась ли задача
- **progress** (*int*) – Ход выполнения задачи (в процентах)
- **result** (*object*) – Результат выполнения задачи или дополнительная информация о ходе её выполнения
- **log** (*string*) – Журнал выполнения задачи

1.12.17 Обработчики уведомлений

Обработчики уведомлений позволяют внешним приложениям уведомлять Weblate об обновлении репозитория системы контроля версий.

Вы можете использовать конечные точки репозитория для проектов, компонентов и переводов для обновления отдельных репозиториев; документацию смотрите в описании `POST /api/projects/(string:project)/repository/`.

GET /hooks/update/(**string:** *project*)**/**
string: *component***/** Не рекомендуется, начиная с версии 2.6: Пожалуйста, используйте вместо этой конечной точки точку `POST /api/components/(string:project)/(string:component)/repository/`, которая корректно работает с аутентификацией для проектов, ограниченных списками контроля доступа.

Запускает обновление компонента (извлечение кода из системы контроля версий и поиск в нём изменений перевода).

GET /hooks/update/(**string:** *project*)**/**
Не рекомендуется, начиная с версии 2.6: Пожалуйста, используйте вместо этой конечной точки точку `POST /api/projects/(string:project)/repository/`, которая корректно работает с аутентификацией для проектов, ограниченных списками контроля доступа.

Запускает обновление всех компонентов в проекте (извлечение кода из системы контроля версий и поиск в нём изменений перевода).

POST /hooks/github/
Специальный обработчик для обработки уведомлений GitHub и автоматического обновления соответствующих компонентов.

Примечание: GitHub включает прямую поддержку уведомлений Weblate: включите сервисный обработчик Weblate в настройках репозитория и установите URL-адрес в соответствие с URL-адресом вашей установки Weblate.

См.также:

Автоматическое получение изменений из GitHub'a Инструкция по настройке интеграции с GitHub

https://docs.github.com/en/github/extending-github/about-webhooks Общая информация о веб-обработчиках GitHub

ENABLE_HOOKS Для включения обработчиков для всего Weblate

POST /hooks/gitlab/
Специальный обработчик для обработки уведомлений GitLab и автоматического обновления соответствующих компонентов.

См.также:

Автоматическое получение изменений из GitLab'a Инструкция по настройке интеграции с GitLab

https://docs.gitlab.com/ce/user/project/integrations/webhooks.html Общая информация о веб-обработчиках GitLab

ENABLE_HOOKS Для включения обработчиков для всего Weblate

POST /hooks/bitbucket/
Специальный обработчик для обработки уведомлений Bitbucket и автоматического обновления соответствующих компонентов.

См.также:

Автоматическое получение изменений из Bitbucket'a Инструкция по настройке интеграции с Bitbucket

https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/ Общая информация о веб-обработчиках Bitbucket

ENABLE_HOOKS Для включения обработчиков для всего Weblate

POST /hooks/pagure/

Добавлено в версии 3.3.

Специальный обработчик для обработки уведомлений Pagure и автоматического обновления соответствующих компонентов.

См.также:

Автоматическое получение изменений из Pagure Инструкция по настройке интеграции с Pagure

https://docs.pagure.org/pagure/usage/using_webhooks.html Общая информация о веб-обработчиках Pagure

ENABLE_HOOKS Для включения обработчиков для всего Weblate

POST /hooks/azure/

Добавлено в версии 3.8.

Специальный обработчик для обработки уведомлений Azure Repos и автоматического обновления соответствующих компонентов.

См.также:

Автоматическое получение изменений из Azure Repos Инструкция по настройке интеграции с Azure

https://docs.microsoft.com/ru-ru/azure/devops/service-hooks/services/webhooks?view=azure-devops Общая информация о веб-обработчиках Azure Repos

ENABLE_HOOKS Для включения обработчиков для всего Weblate

POST /hooks/gitea/

Добавлено в версии 3.9.

Специальный обработчик для обработки уведомлений Gitea и автоматического обновления соответствующих компонентов.

См.также:

Автоматическое получение изменений из репозиториев Gitea Инструкция по настройке интеграции с Gitea

https://docs.gitea.io/en-us/webhooks/ Общая информация о веб-обработчиках Gitea

ENABLE_HOOKS Для включения обработчиков для всего Weblate

POST /hooks/gitee/

Добавлено в версии 3.9.

Специальный обработчик для обработки уведомлений Gitee и автоматического обновления соответствующих компонентов.

См.также:

Автоматическое получение изменений из репозиториев Gitee Инструкция по настройке интеграции с Gitee

https://gitee.com/help/categories/40 Общая информация о веб-обработчиках Gitee

ENABLE_HOOKS Для включения обработчиков для всего Weblate

1.12.18 Экспорты

Чтобы позволить вам и дальше обрабатывать данные, Weblate предоставляет вам различные виды экспорта.

GET /exports/stats/(**string:** *project*)**/**
string: *component***/**

Параметры запроса

- **format** (*string*) – Выходной формат: `json` или `csv`

Не рекомендуется, начиная с версии 2.6: Пожалуйста, используйте вместо этой конечной точки точки *GET /api/components/(string:project)/(string:component)/statistics/* или *GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/*; они также разрешает доступ к проектам, контролируемым списками контроля доступа.

Возвращает статистику по указанному компоненту в заданном формате.

Пример запроса:

```
GET /exports/stats/weblate/main/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Пример ответа:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

[
    {
        "code": "cs",
        "failing": 0,
        "failing_percent": 0.0,
        "fuzzy": 0,
        "fuzzy_percent": 0.0,
        "last_author": "Michal Čihař",
        "last_change": "2012-03-28T15:07:38+00:00",
        "name": "Czech",
        "total": 436,
        "total_words": 15271,
        "translated": 436,
        "translated_percent": 100.0,
        "translated_words": 3201,
        "url": "http://hosted.weblate.org/engage/weblate/cs/",
        "url_translate": "http://hosted.weblate.org/projects/weblate/main/cs/"
    },
    {
        "code": "nl",
        "failing": 21,
        "failing_percent": 4.8,
        "fuzzy": 11,
        "fuzzy_percent": 2.5,
        "last_author": null,
        "last_change": null,
        "name": "Dutch",
        "total": 436,
        "total_words": 15271,
        "translated": 319,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "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/"
    }
]
```

1.12.19 RSS-каналы

Изменения в переводах экспортируются в RSS-каналы.

GET /exports/rss/(**string:** *project*)**/**
string: *component***/string:** *language***/** Возвращает RSS-ленту с последними изменениями в переводе.

GET /exports/rss/(**string:** *project*)**/**
string: *component***/** Возвращает RSS-ленту с последними изменениями в компоненте.

GET /exports/rss/(**string:** *project*)**/**
Возвращает RSS-ленту с последними изменениями в проекте.

GET /exports/rss/language/(**string:** *language*)**/**
Возвращает RSS-ленту с последними изменениями в языке.

GET /exports/rss/
Возвращает RSS-ленту с последними изменениями в экземпляре Weblate.

См.также:

RSS в Википедии

1.13 Клиент Weblate

Добавлено в версии 2.7: Полная поддержка вспомогательной команды wlc существует с версии Weblate 2.7. Если вы используете более старую версию, могут возникнуть некоторые несовместимости с API.

1.13.1 Установка

Клиент Weblate поставляется отдельно и включает в себя модуль Python. Для использования нижеприведённых команд необходимо установить модуль `wlc`:

```
pip3 install wlc
```

1.13.2 Использование Docker

Клиент Weblate также доступен в виде образа Docker.

Этот образ публикуется на Docker Hub: https://hub.docker.com/r/weblate/wlc

Установка:

```
docker pull weblate/wlc
```

Контейнер Docker использует настройки по умолчанию Weblate и соединяется с API, развёрнутом на локальной машине. URL-адрес API и API_KEY можно настроить через параметры, принимаемые Weblate.

У команды запуска контейнера следующий синтаксис:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Пример:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

Также вы можете пробросить *файлы настроек* в контейнер Docker. Проще всего будет это сделать добавив текущий каталог в качестве тома `/home/weblate`:

```
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

1.13.3 Начало работы

Настройки wlc хранятся в `~/.config/weblate` (другие возможные расположения смотрите в разделе *Файлы настроек*), создайте этот файл в соответствии со своим окружением:

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

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

После этого вы сможете вызывать команды на умолчательном сервере:

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

См.также:

Файлы настроек

1.13.4 Обзор

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

Какая операция должна быть выполнена на самом деле указывает параметр `command`.

1.13.5 Описание

Weblate Client — это библиотека Python и инструмент командной строки для удалённого управления Weblate с помощью его *REST API Weblate'a*. Инструмент командной строки может быть вызван как команда **wlc**, которая построена на основе модуля *wlc*.

Аргументы

Программа принимает следующие аргументы, определяющие формат вывода или то, какой надо использовать экземпляр Weblate'a. Они должны быть введены до какой-либо команды.

--format `{csv,json,text,html}`
Определяет формат вывода.

--url `URL`
Задаёт URL-адрес API. Переопределяет любое значение, найденное в файле настроек, смотрите раздел *Файлы настроек*. URL-адрес должен заканчиваться на `/api/`, например, `https://hosted.weblate.org/api/`.

--key `KEY`
Задаёт пользовательский ключ API. Переопределяет любое значение из файла настроек, смотрите раздел *Файлы настроек*. Свой ключ вы можете найти в своём профиле на Weblate.

--config `PATH`
Переопределяет путь к файлу настроек, смотрите раздел *Файлы настроек*.

--config-section `SECTION`
Переопределяет используемый раздел файла настроек, смотрите раздел *Файлы настроек*.

Команды

Доступны следующие команды:

version
Распечатать текущую версию.

list-languages
Вывести список используемых в Weblate языков.

list-projects
Вывести список существующих в Weblate проектов.

list-components
Вывести список существующих в Weblate компонентов.

list-translations
Вывести список существующих в Weblate переводов.

show
Показать объект Weblate (перевод, компонент или проект).

ls
Вывести список объектов Weblate (переводов, компонентов или проектов).

commit
Зафиксировать изменения, внесённые в объект Weblate (перевод, компонент или проект).

pull
Извлечь изменения из удалённого репозитория в объект Weblate (перевод, компонент или проект).

push
Отправить изменения объекта Weblate (перевода, компонента или проекта) в удалённый репозиторий.

reset
Добавлено в версии 0.7: Поддерживается с версии wlc 0.7.

Сбросить изменения в объекте Weblate (переводе, компоненте или проекте) для приведения их в соответствие с удалённым репозиторием.

cleanup
Добавлено в версии 0.9: Поддерживается с версии wlc 0.9.

Удалить любые неотслеживаемые изменения в объекте Weblate (переводе, компоненте или проекте) для приведения их в соответствие с удалённым репозиторием.

repo
Вывести статус репозитория для данного объекта Weblate (перевода, компонента или проекта).

statistics
Вывести подробную статистику по данному объекту Weblate (переводу, компоненту или проекту).

lock-status
Добавлено в версии 0.5: Поддерживается с версии wlc 0.5.

Вывести статус блокировки.

lock
Добавлено в версии 0.5: Поддерживается с версии wlc 0.5.

Заблокировать компонент в Weblate от дальнейшего перевода.

unlock
Добавлено в версии 0.5: Поддерживается с версии wlc 0.5.

Разблокировать перевод компонента Weblate.

changes
Добавлено в версии 0.7: Поддерживается с версии wlc 0.7 и Weblate 2.10.

Вывести список изменений для данного объекта.

download
Добавлено в версии 0.7: Поддерживается с версии wlc 0.7.

Скачать файл перевода.

--convert
Преобразовывать формат файла; если не указан, то ни каких преобразований на стороне сервера не происходит и файл скачивается в том виде, в котором он сохранён в репозитории.

--output
Задаёт файл, в который нужно сохранить файл перевода, если не указан, он будет распечатан в стандартный поток вывода.

upload
Добавлено в версии 0.9: Поддерживается с версии wlc 0.9.

Загрузить файл перевода.

--overwrite
Перезаписывать существующие переводы во время загрузки.

--input
Файл, из которого читается содержимое, если не указан, чтение будет производиться со стандартного потока ввода.

Подсказка: Более подробную информацию по выполнению конкретных команд можно получить с помощью параметра `--help`, например, `wlc ls --help`.

1.13.6 Файлы настроек

`.weblate`, `.weblate.ini`, `weblate.ini` Изменено в версии 1.6: Также допустимы и файлы с расширением *.ini*.

Проектный файл настроек

`C:\Users\ИМЯ\AppData\weblate.ini` Добавлено в версии 1.6.

Пользовательский файл настроек в Windows.

`~/.config/weblate` Пользовательский файл настроек

`/etc/xdg/weblate` Системный файл настроек

Программа следует спецификации XDG, поэтому вы можете изменить пути размещения файлов настроек с помощью переменных окружения `XDG_CONFIG_HOME` и `XDG_CONFIG_DIRS`. В Windows предпочтительным местом для файла настроек является каталог `APPDATA`.

Можно настроить следующие параметры, находящиеся в разделе `[weblate]` (изменить этот раздел вы можете ключом командной строки *`--config-section`*):

`key`
Ключ API для доступа к Weblate.

`url`
Адрес API сервера, по умолчанию установлен в `http://127.0.0.1:8000/api/`.

`translation`
Путь к переводу по умолчанию — компоненту или проекту.

Файл настроек является INI-файлом, например:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

Дополнительно ключи API могут быть сохранены в разделе `[keys]`:

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

Это позволяет вам сохранять ключи в ваших личных настройках, а файл настроек `.weblate` хранить в системе контроля версий, чтобы wlc знал, с каким сервером ему следует общаться.

1.13.7 Примеры

Печать текущей версии программы:

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

Список всех проектов:

```
$ 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/
```

Также вы можете указать, с каким проектом должен работать wlc:

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

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

При такой настройке закоммитить отложенные изменения в текущем проекте проще простого:

```
$ wlc commit
```

1.14 Python API Weblate'a

1.14.1 Установка

Python API поставляется отдельно, чтобы он был у вас появился, вам нужно установить *клиент Weblate*: `wlc`.

```
pip install wlc
```

1.14.2 `wlc`

`WeblateException`

exception `wlc.`**`WeblateException`**

Базовый класс для всех исключений.

Weblate

class wlc.**Weblate**(*key=''*, *url=None*, *config=None*)

Параметры

- **key** (*str*) – Ключ пользователя
- **url** (*str*) – URL-адрес сервера API, если не указан, используется адрес по умолчанию
- **config** (wlc.config.WeblateConfig) – Объект-конфигурация, может переопределять все остальные параметры.

Класс доступа к API, определения ключа API и, необязательно, URL-адреса API.

get(*path*)

Параметры **path** (*str*) – Путь запроса

Тип результата object

Выполняет единичный GET-вызов API.

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

Параметры **path** (*str*) – Путь запроса

Тип результата object

Выполняет единичный GET-вызов API.

1.14.3 wlc.config

WeblateConfig

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

Параметры **section** (*str*) – Используемый раздел конфигурации

Парсер файлов настроек, реализующий спецификацию XDG.

load(*path=None*)

Параметры **path** (*str*) – Путь к файлу, из которого загружаются настройки.

Загружает настройки из файла; если файл не указан, загружает их из файла настроек *wlc* (`~/.config/wlc`), расположенного в вашем каталоге настроек XDG (`/etc/xdg/wlc`).

1.14.4 wlc.main

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

Параметры

- **settings** (*list*) – Параметры для переопределения в виде списка кортежей
- **stdout** (*object*) – Файловый объект стандартного потока вывода печати вывода, по умолчанию используется `sys.stdout`
- **args** (*list*) – Аргументы командной строки для обработки, по умолчанию используется `sys.args`

Главная точка входа для интерфейса командной строки.

@wlc.main.**register_command**(*command*)

Декоратор для регистрации класса *Command* в основном анализаторе, используемом функцией *main()*.

Command

class `wlc.main.`**`Command`**(*args*, *config*, *stdout=None*)

Главный класс для вызова команд.

ГЛАВА 2

Руководство администратора

2.1 Инструкции по настройке

2.1.1 Установка Weblate

Установка с помощью Docker'а

С помощью докеризированного развёртывания Weblate вы можете обзавестись своим личный экземпляр Weblate за считаные секунды. Все зависимости Weblate уже включены в образ. В качестве базы данных по умолчанию настроена PostgreSQL.

Требования к оборудованию

Weblate должен без проблем работать на любом современном оборудовании, ниже приведена минимальная конфигурация, необходимая для запуска Weblate на одном сервере (сам Weblate, база данных и веб-сервер):

- 2 ГБ оперативной памяти
- 2-х ядерный процессор
- 1 ГБ дискового пространства

Чем больше памяти, тем лучше — она используется для кэширования на всех уровнях (на уровне файловой системы, уровне базы данных и уровне Weblate).

Множество одновременно работающих пользователей увеличивают количество необходимых ядер процессора. Для сотен компонентов перевода рекомендуется не менее 4 Гб оперативной памяти.

Типовое использование дискового пространства базой данной находится в районе 300MB на 1 миллион хранимых слов. Пространство необходимое для клонирования репозиториев разнится, хотя Weblate и пытается поддерживать их размер минимальным, делая поверхностные (shallow) копии.

Примечание: Фактические требования к вашей установке Weblate сильно зависят от размера управляемых ею переводов.

Установка

В следующих примерах предполагается, что у вас есть рабочее окружение Docker'а, в котором установлен `docker-compose`. Для получения конкретных инструкций ознакомьтесь с документацией к Docker'у.

1. Склонируйте репозиторий weblate-docker:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Создайте файл `docker-compose.override.yml` со своими настройками. Полный список переменных окружения приведён в разделе *Переменные окружения Docker'а*.

```
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
```

Примечание: Если переменная *WEBLATE_ADMIN_PASSWORD* не установлена, то пользователь-администратор создаётся со случайным паролем, отображаемым при первом запуске.

Приведённый пример настраивает Weblate на прослушку порта 80, чтобы его изменить, отредактируйте отображение портов в файле `docker-compose.override.yml`.

3. Запустите контейнеры Weblate:

```
docker-compose up
```

Наслаждайтесь своим развёрнутым Weblate'ом, он доступен на порту 80 контейнера `weblate`.

Изменено в версии 2.15-2: Недавно настройка изменилась, ранее существовал отдельный контейнер веб-сервера, но с версии 2.15-2 веб-сервер встроен в контейнер Weblate.

Изменено в версии 3.7.1-6: С июля 2019 года (начиная с метки 3.7.1-6) контейнеры больше не запускают своё содержимое под пользователем root. Это изменение привело к изменению открытого порта с 80 на 8080.

См.также:

Вызов команд управления

Контейнер Docker с поддержкой HTTPS

Общие инструкции по развертыванию, пожалуйста, смотрите в разделе *Установка*, в этом разделе упоминаются только отличия от той инструкции.

Использование собственных SSL-сертификатов

Добавлено в версии 3.8-3.

Если вы хотите использовать свой собственный SSL-сертификат, просто поместите его файлы в том данных Weblate (смотрите раздел *Тома контейнеров Docker'а*):

- `ssl/fullchain.pem`, содержащий сертификат и все необходимые сертификаты центров сертификации
- `ssl/privkey.pem`, содержащий закрытый ключ

Обеими этими файлами должен владеть тот же самый пользователь, который запускает контейнер docker 'а, а маска этих файлов должна быть равна `600` (читать и писать в них может только владелец).

Кроме того, контейнер Weblate теперь будет принимать SSL-соединения на порт 4443 и вам нужно будет включить перенаправление портов для HTTPS в переопределении Docker Compose:

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
      - 443:4443
```

Если на этом же сервере у вас уже размещены другие сайты, скорее всего, порты `80` и `443` используются обратным прокси, например, NGINX. Для передачи HTTPS соединения из NGINX'а в контейнер docker 'а вы можете использовать следующую конфигурацию:

```
server {
    listen 443;
    listen [::]:443;

    server_name <SITE_URL>;
    ssl_certificate /etc/letsencrypt/live/<SITE>/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/<SITE>/privkey.pem;

    location / {
            proxy_set_header HOST $host;
            proxy_set_header X-Forwarded-Proto https;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Host $server_name;
            proxy_pass https://127.0.0.1:<EXPOSED_DOCKER_PORT>;
    }
}
```

Замените заполнители `<SITE_URL>`, `<SITE>` и `EXPOSED_DOCKER_PORT` на фактические значения из вашего окружения.

Автоматический выпуск SSL-сертификатов с помощью сервиса Let's Encrypt

Если вы хотите использовать Let's Encrypt для автоматической генерации SSL-сертификатов для общедоступной установки, вам нужно будет добавить дополнительный контейнер Docker с обратным HTTPS прокси — https-portal. Он используется в файле `docker-compose-https.yml`. Затем создайте файл `docker-compose-https.override.yml` со своими настройками:

```
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'
```

При каждом вызове **`docker-compose`** вам нужно будет передавать ему оба этих файла, а затем сделать следующее:

```
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
```

Обновление контейнера Docker'а

Обычно рекомендуется обновлять только контейнер Weblate, а контейнер PostgreSQL сохранять той версии, которая у вас есть, поскольку обновление PostgreSQL довольно болезненно и в большинстве случаев не приносит особых преимуществ.

Вы можете сделать это, оставаясь с существующим docker-compose и просто извлекая последние образы с последующим перезапуском контейнера:

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

При первом запуске база данных Weblate должна быть перенесена автоматически, надобности в дополнительных ручных действиях не возникнет.

Примечание: Обновления через версию 3.0 Weblate не поддерживает. Если вы используете версию 2.x и хотите обновиться до версии 3.x, сначала выполните обновление до последней версии 3.0.1-x (на момент написания этого документа это образ `3.0.1-7`), которая выполнит миграцию, а затем продолжите обновление до более новых версий.

Также вы можете захотеть обновить репозиторий `docker-compose`, хотя в большинстве случаев необходимости в этом нет. Пожалуйста, остерегайтесь в этом случае изменений версии PostgreSQL, поскольку обновление базы данных является не простым делом, более подробную информацию смотрите в этом замечании на GitHub'е.

Вход от имени администратора

После настройки контейнера вы можете войти в систему под пользователем *admin* с паролем, указанным в переменной окружения `WEBLATE_ADMIN_PASSWORD`, или случайным паролем, сгенерированным при первом запуске, если он не был установлен явно.

Для сброса пароля пользователя *admin* перезапустите контейнер с переменной `WEBLATE_ADMIN_PASSWORD`, установленной в новое значение пароля.

См.также:

`WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

Переменные окружения Docker'а

Многие *параметры конфигурации* Weblate в контейнере Docker могут быть установлены через переменные окружения:

Общие параметры

WEBLATE_DEBUG

Настраивает отладочный режим Django, используя её переменную `DEBUG`.

Пример:

```
environment:
  WEBLATE_DEBUG: 1
```

См.также:

Отключение отладочного режима

WEBLATE_LOGLEVEL

Настраивает подробность ведения журнала.

WEBLATE_SITE_TITLE

Изменяет название сайта, показываемое в заголовке всех страниц.

WEBLATE_SITE_DOMAIN

Настраивает домен сайта. Этот параметр является обязательным.

См.также:

Установка правильного домена сайта, `SITE_DOMAIN`

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Настраивает имя и электронную почту администратора сайта. Используется как для параметра `ADMINS`, так и для создания пользователя *admin* (подробнее смотрите в описании переменной окружения `WEBLATE_ADMIN_PASSWORD`).

Пример:

```
environment:
  WEBLATE_ADMIN_NAME: Weblate admin
  WEBLATE_ADMIN_EMAIL: noreply@example.com
```

См.также:

Вход от имени администратора, *Правильная настройка администраторов*, `ADMINS`

WEBLATE_ADMIN_PASSWORD

Устанавливает пароль для пользователя *admin*.

- Если не установлен и пользователя *admin* не существует, он создаётся со случайным паролем, который показывается при первом запуске контейнера.
- Если не установлен и пользователя *admin* не существует, никакие действия не выполняются.
- Если установлен, при каждом запуске контейнера пользователю *admin* устанавливаются соответствующие пароль `WEBLATE_ADMIN_PASSWORD`, имя `WEBLATE_ADMIN_NAME` и адрес электронной почты `WEBLATE_ADMIN_EMAIL`.

> **Предупреждение:** Хранение пароля в файле настроек может представлять из себя угрозу безопасности. Используйте эту переменную только для начальной установки (или пусть Weblate сгенерирует при начальной загрузке случайный пароль) или для восстановления пароля.

См.также:

Вход от имени администратора, `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

WEBLATE_SERVER_EMAIL

WEBLATE_DEFAULT_FROM_EMAIL

Настраивает адрес для исходящих сообщений электронной почты.

См.также:

Настройка отправки электронной почты

WEBLATE_ALLOWED_HOSTS

Настраивает разрешённые имена HTTP-хостов с помощью параметра `ALLOWED_HOSTS`.

По умолчанию установлен в `*`, что позволяет использовать все имена хостов.

Пример:

```
environment:
  WEBLATE_ALLOWED_HOSTS: weblate.example.com,example.com
```

См.также:

`ALLOWED_HOSTS`, *Настройка разрешённых хостов*, *Установка правильного домена сайта*

WEBLATE_REGISTRATION_OPEN

Настраивает статус открытия регистрации, переключая параметр `REGISTRATION_OPEN`.

Пример:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Настраивает через параметр `REGISTRATION_ALLOW_BACKENDS` методы аутентификации, которые можно использовать для создания новой учётной записи.

Пример:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
  WEBLATE_REGISTRATION_ALLOW_BACKENDS: azuread-oauth2,azuread-tenant-
↪oauth2
```

WEBLATE_TIME_ZONE

Настраивает используемый в Weblate часовой пояс, смотрите описание параметра `TIME_ZONE`.

Примечание: Для изменения часового пояса самого контейнера Docker используйте переменную окружения `TZ`.

Пример:

```
environment:
  WEBLATE_TIME_ZONE: Europe/Prague
```

WEBLATE_ENABLE_HTTPS

Заставляет Weblate думать, что он работает за обратным HTTPS-прокси, что принуждает Weblate использовать HTTPS в ссылках писем электронной почты и API или устанавливать у кук флаги безопасности.

Подсказка: Смотрите возможные топкие места в документацию к параметру *ENABLE_HTTPS*.

Примечание: Этот параметр не включает разрешение на приём контейнером Weblate'а соединений по HTTPS, его нужно настроить отдельно, для примеров смотрите раздел *Контейнер Docker с поддержкой HTTPS*.

Пример:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

См.также:

ENABLE_HTTPS *Установка правильного домена сайта*, *WEBLATE_SECURE_PROXY_SSL_HEADER*

WEBLATE_IP_PROXY_HEADER

Позволяет Weblate'у получать IP-адрес из любого заданного HTTP-заголовка. Используйте его при использовании обратного прокси перед контейнером Weblate.

Включает параметр *IP_BEHIND_REVERSE_PROXY* и устанавливает параметр *IP_PROXY_HEADER*.

Примечание: Формат параметра должен соответствовать ожиданиям Django. Django преобразует необработанные имена HTTP-заголовков следующим образом:

- переводит все символы в верхний регистр
- все дефисы заменяет на подчеркивания
- добавляет префикс `` HTTP_``

Таким образом, заголовок `X-Forwarded-For` отображается на ``HTTP_X_FORWARDED_FOR.

Пример:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

Кортеж, представляющий собой комбинацию HTTP-заголовков/значений, указывающую, что запрос является безопасным. Он необходим, когда Weblate работает за обратным прокси, выполняющим SSL-терминацию, которая не передаёт стандартные HTTPS-заголовки.

Пример:

```
environment:
  WEBLATE_SECURE_PROXY_SSL_HEADER: HTTP_X_FORWARDED_PROTO,https
```

См.также:

SECURE_PROXY_SSL_HEADER

WEBLATE_REQUIRE_LOGIN

Включает параметр *REQUIRE_LOGIN*, в результате чего аутентификация будет требоваться для всего Weblate.

Пример:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

Добавляет исключения из URL-адресов (для которых требование аутентификации установлено глобально для всего Weblate) с помощью параметра *LOGIN_REQUIRED_URLS_EXCEPTIONS*.

Вы можете либо заменить все настройки целиком, либо изменить значение по умолчанию с помощью переменных ADD и REMOVE.

WEBLATE_GOOGLE_ANALYTICS_ID

Настраивает идентификатор для Google Analytics, изменяя параметр *GOOGLE_ANALYTICS_ID*.

WEBLATE_GITHUB_USERNAME

Настраивает имя пользователя GitHub для GitHub'овских запросов на извлечение, изменяя параметр *GITHUB_USERNAME*.

См.также:

GitHub

WEBLATE_GITHUB_TOKEN

Добавлено в версии 4.3.

Настраивает GitHub'овский токен персонального доступа для GitHub'овских запросов на извлечение через API, изменяя параметр *GITHUB_TOKEN*.

См.также:

GitHub

WEBLATE_GITLAB_USERNAME

Настраивает имя пользователя GitLab для GitLab'овских запросов на слияние, изменяя параметр *GITLAB_USERNAME*

См.также:

GitLab

WEBLATE_GITLAB_TOKEN

Настраивает GitLab'овский токен персонального доступа для GitLab'овских запросов на слияние через API, изменяя параметр *GITLAB_TOKEN*

См.также:

GitLab

WEBLATE_PAGURE_USERNAME

Настраивает имя пользователя Pagure для Pagure'вских запросов на слияние, изменяя параметр *PAGURE_USERNAME*

См.также:

Pagure

WEBLATE_PAGURE_TOKEN

Настраивает Pagure'вский токен персонального доступа для Pagure'вских запросов на слияние через API, изменяя параметр `Pagure_TOKEN`

См.также:

Pagure

WEBLATE_SIMPLIFY_LANGUAGES

Настраивает политику упрощения языка, смотрите описание параметра *SIMPLIFY_LANGUAGES*.

WEBLATE_DEFAULT_ACCESS_CONTROL

Настраивает значение по умолчанию для *управления доступом* в новых проектах, смотрите описание параметра *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Настраивает значение по умолчанию для *ограниченного доступа* в новых компонентах, смотрите описание параметра *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Настраивает значение по умолчанию для *разрешения распространения переводов* в новых компонентах, смотрите описание параметра *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

Настраивает параметр *DEFAULT_COMMITER_EMAIL*.

WEBLATE_DEFAULT_COMMITER_NAME

Настраивает параметр *DEFAULT_COMMITER_NAME*.

WEBLATE_DEFAULT_SHARED_TM

Configures *DEFAULT_SHARED_TM*.

WEBLATE_AKISMET_API_KEY

Настраивает API-ключ Akismet, смотрите описание параметра *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

Настраивает GPG для подписи коммитов, смотрите описание параметра *WEBLATE_GPG_IDENTITY*.

См.также:

Подписание коммитов Git с помощью GnuPG

WEBLATE_URL_PREFIX

Настраивает префикс URL-адреса, на котором запущен Weblate, смотрите описание параметра *URL_PREFIX*.

WEBLATE_SILENCED_SYSTEM_CHECKS

Настраивает проверки, которые вы не хотите видеть, смотрите описание параметра `SILENCED_SYSTEM_CHECKS`.

WEBLATE_CSP_SCRIPT_SRC

WEBLATE_CSP_IMG_SRC

WEBLATE_CSP_CONNECT_SRC

WEBLATE_CSP_STYLE_SRC

WEBLATE_CSP_FONT_SRC

Позволяет настраивать HTTP-заголовок политики безопасности содержимого `Content-Security-Policy`.

См.также:

Политика безопасности содержимого, *CSP_SCRIPT_SRC*, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

WEBLATE_LICENSE_FILTER
Настраивает параметр *LICENSE_FILTER*.

WEBLATE_LICENSE_REQUIRED
Configures *LICENSE_REQUIRED*

WEBLATE_WEBSITE_REQUIRED
Configures *WEBSITE_REQUIRED*

WEBLATE_HIDE_VERSION
Настраивает параметр *HIDE_VERSION*.

WEBLATE_BASIC_LANGUAGES
Настраивает параметр *BASIC_LANGUAGES*.

WEBLATE_DEFAULT_AUTO_WATCH
Настраивает параметр *DEFAULT_AUTO_WATCH*.

Параметры машинного перевода

WEBLATE_MT_APERTIUM_APY
Включает машинный перевод от *Apertium* и устанавливает параметр *MT_APERTIUM_APY*

WEBLATE_MT_AWS_REGION

WEBLATE_MT_AWS_ACCESS_KEY_ID

WEBLATE_MT_AWS_SECRET_ACCESS_KEY
Настраивает машинный перевод от *AWS*.

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

WEBLATE_MT_DEEPL_KEY
Включает машинный перевод от *DeepL* и устанавливает параметр *MT_DEEPL_KEY*

WEBLATE_MT_DEEPL_API_VERSION
Настраивает используемую версию API *DeepL*, смотрите описание параметра *MT_DEEPL_API_VERSION*.

WEBLATE_MT_GOOGLE_KEY
Включает машинный перевод от *Google Translate* и устанавливает параметр *MT_GOOGLE_KEY*

WEBLATE_MT_MICROSOFT_COGNITIVE_KEY
Включает машинный перевод от *Переводчика Microsoft Cognitive Services* и устанавливает параметр *MT_MICROSOFT_COGNITIVE_KEY*

WEBLATE_MT_MICROSOFT_ENDPOINT_URL
Устанавливает параметр *MT_MICROSOFT_ENDPOINT_URL*, обратите внимание, что он должен содержать только доменное имя.

WEBLATE_MT_MICROSOFT_REGION
Устанавливает параметр *MT_MICROSOFT_REGION*

WEBLATE_MT_MICROSOFT_BASE_URL
Устанавливает параметр *MT_MICROSOFT_BASE_URL*

WEBLATE_MT_MODERNMT_KEY
Включает машинный перевод от *ModernMT* и устанавливает параметр *MT_MODERNMT_KEY*.

WEBLATE_MT_MYMEMORY_ENABLED

Включает машинный перевод от *MyMemory* и устанавливает параметр *MT_MYMEMORY_EMAIL* в значение переменной окружения *WEBLATE_ADMIN_EMAIL*.

Пример:

```
environment:
  WEBLATE_MT_MYMEMORY_ENABLED: 1
```

WEBLATE_MT_GLOSBE_ENABLED

Включает машинный перевод от *Glosbe*.

```
environment:
  WEBLATE_MT_GLOSBE_ENABLED: 1
```

WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED

Включает машинный перевод от *терминологической службы Майкрософт*.

```
environment:
  WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED: 1
```

WEBLATE_MT_SAP_BASE_URL

WEBLATE_MT_SAP_SANDBOX_APIKEY

WEBLATE_MT_SAP_USERNAME

WEBLATE_MT_SAP_PASSWORD

WEBLATE_MT_SAP_USE_MT

Настраивает машинный перевод от *SAP Translation Hub*.

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

Параметры аутентификации

LDAP

WEBLATE_AUTH_LDAP_SERVER_URI

WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE

WEBLATE_AUTH_LDAP_USER_ATTR_MAP

WEBLATE_AUTH_LDAP_BIND_DN

WEBLATE_AUTH_LDAP_BIND_PASSWORD

WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS

WEBLATE_AUTH_LDAP_USER_SEARCH

WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER

WEBLATE_AUTH_LDAP_USER_SEARCH_UNION

WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER

Настройка аутентификации через LDAP.

Пример прямой привязки:

```
environment:
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE: uid=%(user)s,ou=People,dc=example,dc=net
  # map weblate 'full_name' to ldap 'name' and weblate 'email' attribute to
↪'mail' ldap attribute.
  # another example that can be used with OpenLDAP: 'full_name:cn,email:mail'
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
```

Пример поиска и привязки:

```
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
```

Пример объединения поиска и привязки:

```
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
```

Пример поиска и привязки в 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)
```

См.также:

Аутентификация через LDAP

GitHub

WEBLATE_SOCIAL_AUTH_GITHUB_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_SECRET

Включает *аутентификацию через GitHub*.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET

Включает *аутентификацию через Bitbucket*.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET
Включает *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
Включает *Google OAuth 2*.

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY

WEBLATE_SOCIAL_AUTH_GITLAB_SECRET

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL
Включает *GitLab OAuth 2*.

Azure Active Directory

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET
Включает аутентификацию через Azure Active Directory, смотрите раздел *Microsoft Azure Active Directory*.

Azure Active Directory с поддержкой Tenant

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID
Включает аутентификацию через Azure Active Directory с поддержкой Tenant, смотрите раздел *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
Включает аутентификацию через Keycloak, смотрите документацию.

Поставщики Linux

Вы можете включить аутентификацию с помощью сервисов аутентификации поставщиков Linux, установив следующие переменные в любое значение.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET
Включает аутентификацию через Slack, смотрите раздел *Slack*.

SAML

Самоподписанные ключи SAML генерируются автоматически при первом запуске контейнера. Если вы хотите использовать собственные ключи, поместите сертификат и закрытый ключ в файлы `/app/data/ssl/saml.crt` и `/app/data/ssl/saml.key`.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT
Настройки провайдера идентификационных данных SAML, смотрите раздел *Аутентификация через SAML*.

Другие параметры аутентификации

WEBLATE_NO_EMAIL_AUTH
При установке в любое значение отключает аутентификацию через электронную почту.

Настройки базы данных PostgreSQL

База данных создается в файле `docker-compose.yml`, поэтому эти настройки влияют как на контейнер Weblate, так и на контейнер PostgreSQL.

См.также:

Настройка базы данных для Weblate

POSTGRES_PASSWORD
Пароль PostgreSQL.

POSTGRES_USER
Имя пользователя PostgreSQL.

POSTGRES_DATABASE
Имя базы данных PostgreSQL.

POSTGRES_HOST
Имя хоста или IP-адрес сервера PostgreSQL. По умолчанию равен `database`.

POSTGRES_PORT
Порт сервера PostgreSQL. По умолчанию не установлен (используется значение по умолчанию).

POSTGRES_SSL_MODE
Настраивает обработку SSL сервером PostgreSQL при соединении с сервером, возможные варианты настройки смотрите документе Описания режимов SSL

POSTGRES_ALTER_ROLE
Устанавливает имя роли, которую Weblate будет настраивать во время миграции, см. *Настройка Weblate для использования PostgreSQL*.

Параметры резервного копирования базы данных

См.также:

Дампы данных для резервного копирования

WEBLATE_DATABASE_BACKUP
Настраивает ежедневный дамп базы данных с помощью параметра `DATABASE_BACKUP`. По умолчанию установлен в `plain`.

Настройка сервера кэширования

Weblate настоятельно рекомендует использовать Redis и при запуске Weblate'a в Docker'e вы должны предоставить экземпляр Redis'a.

См.также:

Включение кэширования

REDIS_HOST
Имя хоста или IP-адрес сервера Redis. По умолчанию установлен в `cache`.

REDIS_PORT
Порт сервера Redis. По умолчанию установлен в `6379`.

REDIS_DB
Номер базы данных Redis, по умолчанию установлен в `1`.

REDIS_PASSWORD
Пароль сервера Redis, по умолчанию не используется.

REDIS_TLS
Включает использование SSL для соединения с Redis.

REDIS_VERIFY_SSL
Может использоваться для отключения проверки SSL-сертификата для соединениий с Redis.

Настройка почтового сервера

Чтобы работала отправка писем электронной почты, необходимо предоставить почтовый сервер.

Пример конфигурации TLS:

```
environment:
    WEBLATE_EMAIL_HOST: smtp.example.com
    WEBLATE_EMAIL_HOST_USER: user
    WEBLATE_EMAIL_HOST_PASSWORD: pass
```

Пример конфигурации SSL:

```
environment:
    WEBLATE_EMAIL_HOST: smtp.example.com
    WEBLATE_EMAIL_PORT: 465
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    WEBLATE_EMAIL_HOST_USER: user
    WEBLATE_EMAIL_HOST_PASSWORD: pass
    WEBLATE_EMAIL_USE_TLS: 0
    WEBLATE_EMAIL_USE_SSL: 1
```

См.также:

Настройка исходящей почты

WEBLATE_EMAIL_HOST

Имя хоста или IP-адрес почтового сервера.

См.также:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_SSL*, *WEBLATE_EMAIL_USE_TLS*, EMAIL_HOST

WEBLATE_EMAIL_PORT

Порт почтового сервера, по умолчанию установлен в 25.

См.также:

EMAIL_PORT

WEBLATE_EMAIL_HOST_USER

Пользователь для аутентификации по электронной почте.

См.также:

EMAIL_HOST_USER

WEBLATE_EMAIL_HOST_PASSWORD

Пароль для аутентификации по электронной почте.

См.также:

EMAIL_HOST_PASSWORD

WEBLATE_EMAIL_USE_SSL

Использовать ли неявное TLS (безопасное) соединение при общении с сервером SMTP. В большей части документации по электронной почте этот тип TLS-соединения называется SSL. Он обычно используется на 465-м порту. Если у вас возникли проблемы, попробуйте включить явный TLS параметром *WEBLATE_EMAIL_USE_TLS*.

См.также:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_TLS*, EMAIL_USE_SSL

WEBLATE_EMAIL_USE_TLS

Использовать ли TLS (безопасное) соединение при общении с сервером SMTP. Он используется для явных TLS-соединений, обычно на портах 587 или 25. Если ваши соединения зависают, попробуйте включить неявный TLS параметром *WEBLATE_EMAIL_USE_SSL*.

См.также:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_SSL*, EMAIL_USE_TLS

WEBLATE_EMAIL_BACKEND

Настраивает бэкенд Django, используемый для отправки сообщений электронной почты.

См.также:

Настройка отправки электронной почты, EMAIL_BACKEND

Site integration

WEBLATE_GET_HELP_URL
Configures *GET_HELP_URL*.

WEBLATE_STATUS_URL
Configures *STATUS_URL*.

WEBLATE_LEGAL_URL
Configures *LEGAL_URL*.

Отчёты об ошибках

Рекомендуется систематически собирать ошибки от установки, смотрите раздел *Сбор отчётов об ошибках*.

Для включения поддержки Rollbar установите следующие переменные:

ROLLBAR_KEY
Ваш токен доступа к серверу Rollbar.

ROLLBAR_ENVIRONMENT
Ваше окружение Rollbar, по умолчанию установлена в `production`.

Для включения поддержки Sentry установите следующие переменные:

SENTRY_DSN
Ваш DSN Sentry.

SENTRY_ENVIRONMENT
Ваше окружение Sentry (необязательная).

CDN локализации

WEBLATE_LOCALIZE_CDN_URL

WEBLATE_LOCALIZE_CDN_PATH
Добавлено в версии 4.2.1.

Настройки для *CDN локализации JavaScript'а*.

Переменная *WEBLATE_LOCALIZE_CDN_PATH* указывает путь внутри контейнера. Он должен указывать на постоянный том, а не на временное хранилище.

Одной из возможных настроек является направление пути внутрь каталога данных Weblate:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

Примечание: За раздачу файлов, генерируемых Weblate, отвечаете вы, эта переменная только указывает местоположение, куда их надо сохранять.

См.также:

weblate-cdn, *LOCALIZE_CDN_URL*, *LOCALIZE_CDN_PATH*

Изменение включённых приложений, проверок, надстроек или автоматических исправлений

Добавлено в версии 3.8-5.

Встроенную конфигурация включённых проверок, надстроек или автоматических исправлений можно изменить с помощью следующих переменных:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Пример:

```
environment:
  WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
↪SameBookendingWhitespace
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon,customize.addons.OtherAddon
```

См.также:

CHECK_LIST, *AUTOFIX_LIST*, *WEBLATE_ADDONS*, INSTALLED_APPS

Параметры контейнера

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

Эти переменные позволяют настроить параметры рабочего Celery. Может быть полезно подправить степень параллелизма (`--concurrency 16`) или включить использование другой реализации пула (`--pool=gevent`).

По умолчанию количество параллельных рабочих соответствует количеству процессоров (за исключением рабочего создания резервных копий, который должен быть запущен в единственном экземпляре).

Пример:

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

См.также:

Параметры рабочих Celery, *Фоновые задачи с использованием Celery*

UWSGI_WORKERS

Настраивает количество выполняемых рабочих процессов uWSGI.

По умолчанию установлена в количество процессоров + 1.

Пример:

```
environment:
  UWSGI_WORKERS: 32
```

Если на вашей машине много процессорных ядер и возникают проблемы, вызванные нехваткой памяти, то попробуйте уменьшить количество рабочих процессов:

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

Тома контейнеров Docker'a

Контейнер Weblate экспортирует только один том данных. Контейнеры других сервисов (PostgreSQL или Redis) имеют свои собственные тома данных, которые в настоящем документе не рассматриваются.

Том данных используется для хранения таких постоянных данных Weblate, как склонированные репозитории, или для настройки установки Weblate.

Непосредственное размещение этого тома в хост-системе зависит от настроек самого Docker'a, но обычно он хранится в каталоге `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/`. В контейнере он монтируется под именем `/app/data`.

См.также:

Документация по томам Docker'a

Дальнейшая настройка

Вы можете и дальше настраивать установку Weblate в томе данных, смотрите раздел *Тома контейнеров Docker'a*.

Пользовательские файлы конфигурации

В дополнителнение ко всему вы можете переопределить параметры в файле `/app/data/settings-override.py` (см. *Тома контейнеров Docker'a*). Этот файл выполняется в самом конце встроенных настроек, после загрузки всех настроек из переменных окружения, так что вы можпте подстроить или переопределить их.

Замена логотипа и других статических файлов

Добавлено в версии 3.8-5.

Статические файлы, поставляемые с Weblate, можно переопределить, поместив их в каталог `/app/data/python/customize/static` (смотрите раздел *Тома контейнеров Docker'a*). Например, создав файл `/app/data/python/customize/static/favicon.ico`, вы замените им иконку сайта.

Подсказка: При запуске контейнера файлы копируются в соответствующее место, поэтому после изменения содержимого тома Weblate требуется перезапустить.

В качестве альтернативы вы также можете включить свой собственный модуль (смотрите раздел *Настройка Weblate*) и добавить его в контейнер Docker'a отдельным томом, например:

```
weblate:
  volumes:
    - weblate-data:/app/data
    - ./weblate_customization/weblate_customization:/app/data/python/weblate_
↪customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

Добавление собственных модулей Python

Добавлено в версии 3.8-5.

Собственные модули Python вы можете разместить в каталоге `/app/data/python/` (смотрите раздел *Тома контейнеров Docker'а*) — в дальнейшем они могут быть загружены в Weblate, скорее всего, через *пользовательские файлы настроек*.

См.также:

Настройка Weblate

Выбор вашей машины — локальные или облачные провайдеры

С помощью Docker Machine вы можете развернуть Weblate как на локальной машине, так и на любом большом количестве облачных развёртываний, например на Amazon AWS, Greenhost и на многих других провайдерах.

Установка на Debian и Ubuntu

Требования к оборудованию

Weblate должен без проблем работать на любом современном оборудовании, ниже приведена минимальная конфигурация, необходимая для запуска Weblate на одном сервере (сам Weblate, база данных и веб-сервер):

- 2 ГБ оперативной памяти
- 2-х ядерный процессор
- 1 ГБ дискового пространства

Чем больше памяти, тем лучше — она используется для кэширования на всех уровнях (на уровне файловой системы, уровне базы данных и уровне Weblate).

Множество одновременно работающих пользователей увеличивают количество необходимых ядер процессора. Для сотен компонентов перевода рекомендуется не менее 4 Гб оперативной памяти.

Типовое использование дискового пространства базой данной находится в районе 300MB на 1 миллион хранимых слов. Пространство необходимое для клонирования репозиториев разнится, хотя Weblate и пытается поддерживать их размер минимальным, делая поверхностные (shallow) копии.

Примечание: Фактические требования к вашей установке Weblate сильно зависят от размера управляемых ею переводов.

Установка

Системные требования

Установите зависимости, необходимые для сборки модулей Python (смотрите раздел *Требования к программному обеспечению*):

```
apt install \
   libxml2-dev libxslt-dev libfreetype6-dev libjpeg-dev libz-dev libyaml-dev \
   libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev libacl1-dev libssl-dev \
   build-essential python3-gdbm python3-dev python3-pip python3-virtualenv␣
↪virtualenv git
```

Установите желаемые необязательные зависимости в зависимости от функций, которые вы собираетесь использовать (смотрите раздел *Необязательные зависимости*):

```
apt install tesseract-ocr libtesseract-dev libleptonica-dev
```

При желании установите программное обеспечение для запуска рабочего сервера, смотрите разделы *Запуск сервера*, *Настройка базы данных для Weblate* и *Фоновые задачи с использованием Celery*. В зависимости от размера вашей установки вы можете захотеть запустить эти компоненты на выделенных серверах.

Инструкции по установке на местах:

```
# Web server option 1: NGINX and uWSGI
apt install nginx uwsgi uwsgi-plugin-python3

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

# Caching backend: Redis
apt install redis-server

# Database server: PostgreSQL
apt install postgresql postgresql-contrib

# SMTP server
apt install exim4
```

Модули Python

Подсказка: Для установки Weblate мы используем виртуальное окружение, предоставляемое пакетом virtualenv, чтобы отделить Weblate от вашей системы. Если вы не знакомы с этим понятием, обратитесь к документации пакета.

1. Создайте виртуальное окружение для Weblate:

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

2. Активируйте виртуальное окружение для Weblate:

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

3. Установите Weblate, включая все его зависимости:

   ```
   pip install Weblate
   ```

4. Установите драйвер базы данных:

```
pip install psycopg2-binary
```

5. Установите желаемые необязательные зависимости в зависимости от функций, которые вы собираетесь использовать (некоторые из них могут потребовать дополнительных системных библиотек, проверьте раздел *Необязательные зависимости*):

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

Настройка Weblate

Примечание: Следующие шаги подразумевают, что виртуальное окружение, используемое Weblate'ом, активно (активировать его можно командой `. ~/weblate-env/bin/activate`). В случае, если это не так, вам придётся указывать полный путь к команде **`weblate`**: `~/weblate-env/bin/weblate`.

1. Скопируйте файл `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` в `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. По своему усмотрению подкорректируйте значения в новом файле `settings.py`. Для тестирования вы можете использовать поставляемый пример, но для рабочей среды вы захотите внести в него изменения, смотрите раздел *Изменение конфигурации под свои нужды*.
3. Создайте базу данных и её структуру для Weblate (пример настройки использует PostgreSQL, для установки в рабочую среду ознакомьтесь разделом *Настройка базы данных для Weblate*):

```
weblate migrate
```

4. Создайте учётную запись администратора и скопируйте сгенерированный ей пароль в буфер обмена, также сохраните его для дальнейшего использования:

```
weblate createadmin
```

5. Соберите статические файлы для веб-сервера (смотрите разделы *Запуск сервера* и *Обслуживание статических файлов*):

```
weblate collectstatic
```

6. Сожмите файлы JavaScript и CSS (необязательно, смотрите раздел *Сжатие клиентских ресурсов*):

```
weblate compress
```

7. Запустите рабочие процессы Celery. Для целей разработки это делать не обязательно, но настоятельно рекомендуется. Дополнительную информацию смотрите в разделе *Фоновые задачи с использованием Celery*:

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

8. Запустите сервер разработки (шаги для запуска рабочего сервера смотрите в разделе *Запуск сервера*):

```
weblate runserver
```

После установки

Поздравляем, теперь ваш сервер Weblate запущен и вы можете начать его использовать.

- Теперь вы можете получить доступ к Weblate по адресу `http://localhost:8000/`.
- Войдите в систему с учётными данными администратора, полученными во время установки, или зарегистрируйтесь в качестве нового пользователя.
- Теперь при помощи команды **`weblate`** и когда активно виртуальное окружение Weblate вы можете запускать его команды, смотрите раздел *Команды управления*.
- Остановить тестовый сервер можно комбинацией `Ctrl+C`.
- Просмотрите список вероятных проблем с вашей установкой или по адресу `/manage/performance/` в URL, или с помощью команды **`weblate check --deploy`**; см. также *Рабочая среда*.

Добавление перевода

1. Откройте интерфейс администратора (`http://localhost:8000/create/project/`) и создайте проект, который вы хотите перевести. Подробнее смотрите в разделе *Настройки проекта*.

   Всё, что вам нужно здесь указать — это название проекта и его веб-сайт.

2. Создайте компонент, который является реальным объектом для перевода — он указывает на репозиторий системы контроля версий и задаёт файлы для перевода. Подробнее смотрите в разделе *Настройки компонента*.

   Здесь важны следующие поля: название компонента, адрес репозитория системы контроля версий и маска для поиска переводимых файлов. Weblate поддерживает широкий спектр форматов, включая PO-файлы gettext, строки ресурсов Android'а, строки свойств iOS, файлы свойств Java или Qt Linguist, подробнее смотрите в разделе *Поддерживаемые форматы файлов*.

3. После того, как вышеперечисленные действия будут завершены (это может быть довольно длительный процесс в зависимости от размера вашего репозитория системы контроля версий и количества сообщений для перевода), вы сможете начать перевод.

Установка на SUSE и openSUSE

Требования к оборудованию

Weblate должен без проблем работать на любом современном оборудовании, ниже приведена минимальная конфигурация, необходимая для запуска Weblate на одном сервере (сам Weblate, база данных и веб-сервер):

- 2 ГБ оперативной памяти
- 2-х ядерный процессор
- 1 ГБ дискового пространства

Чем больше памяти, тем лучше — она используется для кэширования на всех уровнях (на уровне файловой системы, уровне базы данных и уровне Weblate).

Множество одновременно работающих пользователей увеличивают количество необходимых ядер процессора. Для сотен компонентов перевода рекомендуется не менее 4 Гб оперативной памяти.

Типовое использование дискового пространства базой данной находится в районе 300MB на 1 миллион хранимых слов. Пространство необходимое для клонирования репозиториев разнится, хотя Weblate и пытается поддерживать их размер минимальным, делая поверхностные (shallow) копии.

Примечание: Фактические требования к вашей установке Weblate сильно зависят от размера управляемых ею переводов.

Установка

Системные требования

Установите зависимости, необходимые для сборки модулей Python (смотрите раздел *Требования к программному обеспечению*):

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

Установите желаемые необязательные зависимости в зависимости от функций, которые вы собираетесь использовать (смотрите раздел *Необязательные зависимости*):

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

При желании установите программное обеспечение для запуска рабочего сервера, смотрите разделы *Запуск сервера*, *Настройка базы данных для Weblate* и *Фоновые задачи с использованием Celery*. В зависимости от размера вашей установки вы можете захотеть запустить эти компоненты на выделенных серверах.

Инструкции по установке на местах:

```
# 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
```

Модули Python

Подсказка: Для установки Weblate мы используем виртуальное окружение, предоставляемое пакетом virtualenv, чтобы отделить Weblate от вашей системы. Если вы не знакомы с этим понятием, обратитесь к документации пакета.

1. Создайте виртуальное окружение для Weblate:

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

2. Активируйте виртуальное окружение для Weblate:

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

3. Установите Weblate, включая все его зависимости:

```
pip install Weblate
```

4. Установите драйвер базы данных:

```
pip install psycopg2-binary
```

5. Установите желаемые необязательные зависимости в зависимости от функций, которые вы собираетесь использовать (некоторые из них могут потребовать дополнительных системных библиотек, проверьте раздел *Необязательные зависимости*):

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

Настройка Weblate

Примечание: Следующие шаги подразумевают, что виртуальное окружение, используемое Weblate'ом, активно (активировать его можно командой `. ~/weblate-env/bin/activate`). В случае, если это не так, вам придётся указывать полный путь к команде **`weblate`**: `~/weblate-env/bin/weblate`.

1. Скопируйте файл `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` в `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. По своему усмотрению подкорректируйте значения в новом файле `settings.py`. Для тестирования вы можете использовать поставляемый пример, но для рабочей среды вы захотите внести в него изменения, смотрите раздел *Изменение конфигурации под свои нужды*.
3. Создайте базу данных и её структуру для Weblate (пример настройки использует PostgreSQL, для установки в рабочую среду ознакомьтесь разделом *Настройка базы данных для Weblate*):

```
weblate migrate
```

4. Создайте учётную запись администратора и скопируйте сгенерированный ей пароль в буфер обмена, также сохраните его для дальнейшего использования:

```
weblate createadmin
```

5. Соберите статические файлы для веб-сервера (смотрите разделы *Запуск сервера* и *Обслуживание статических файлов*):

```
weblate collectstatic
```

6. Сожмите файлы JavaScript и CSS (необязательно, смотрите раздел *Сжатие клиентских ресурсов*):

```
weblate compress
```

7. Запустите рабочие процессы Celery. Для целей разработки это делать не обязательно, но настоятельно рекомендуется. Дополнительную информацию смотрите в разделе *Фоновые задачи с использованием Celery*:

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

8. Запустите сервер разработки (шаги для запуска рабочего сервера смотрите в разделе *Запуск сервера*):

```
weblate runserver
```

После установки

Поздравляем, теперь ваш сервер Weblate запущен и вы можете начать его использовать.

- Теперь вы можете получить доступ к Weblate по адресу `http://localhost:8000/`.
- Войдите в систему с учётными данными администратора, полученными во время установки, или зарегистрируйтесь в качестве нового пользователя.
- Теперь при помощи команды **`weblate`** и когда активно виртуальное окружение Weblate вы можете запускать его команды, смотрите раздел *Команды управления*.
- Остановить тестовый сервер можно комбинацией `Ctrl+C`.
- Просмотрите список вероятных проблем с вашей установкой или по адресу `/manage/performance/` в URL, или с помощью команды **`weblate check --deploy`**; см. также *Рабочая среда*.

Добавление перевода

1. Откройте интерфейс администратора (`http://localhost:8000/create/project/`) и создайте проект, который вы хотите перевести. Подробнее смотрите в разделе *Настройки проекта*.

   Всё, что вам нужно здесь указать — это название проекта и его веб-сайт.

2. Создайте компонент, который является реальным объектом для перевода — он указывает на репозиторий системы контроля версий и задаёт файлы для перевода. Подробнее смотрите в разделе *Настройки компонента*.

   Здесь важны следующие поля: название компонента, адрес репозитория системы контроля версий и маска для поиска переводимых файлов. Weblate поддерживает широкий спектр форматов, включая PO-файлы gettext, строки ресурсов Android'a, строки свойств iOS, файлы свойств Java или Qt Linguist, подробнее смотрите в разделе *Поддерживаемые форматы файлов*.

3. После того, как вышеперечисленные действия будут завершены (это может быть довольно длительный процесс в зависимости от размера вашего репозитория системы контроля версий и количества сообщений для перевода), вы сможете начать перевод.

Установка на RedHat, Fedora и CentOS

Требования к оборудованию

Weblate должен без проблем работать на любом современном оборудовании, ниже приведена минимальная конфигурация, необходимая для запуска Weblate на одном сервере (сам Weblate, база данных и веб-сервер):

- 2 ГБ оперативной памяти
- 2-х ядерный процессор
- 1 ГБ дискового пространства

Чем больше памяти, тем лучше — она используется для кэширования на всех уровнях (на уровне файловой системы, уровне базы данных и уровне Weblate).

Множество одновременно работающих пользователей увеличивают количество необходимых ядер процессора. Для сотен компонентов перевода рекомендуется не менее 4 Гб оперативной памяти.

Типовое использование дискового пространства базой данной находится в районе 300MB на 1 миллион хранимых слов. Пространство необходимое для клонирования репозиториев разнится, хотя Weblate и пытается поддерживать их размер минимальным, делая поверхностные (shallow) копии.

Примечание: Фактические требования к вашей установке Weblate сильно зависят от размера управляемых ею переводов.

Установка

Системные требования

Установите зависимости, необходимые для сборки модулей Python (смотрите раздел *Требования к программному обеспечению*):

```
dnf install \
   libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel libyaml-
↪devel \
   cairo-devel pango-devel gobject-introspection-devel libacl-devel \
   python3-pip python3-virtualenv python3-devel git
```

Установите желаемые необязательные зависимости в зависимости от функций, которые вы собираетесь использовать (смотрите раздел *Необязательные зависимости*):

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

При желании установите программное обеспечение для запуска рабочего сервера, смотрите разделы *Запуск сервера*, *Настройка базы данных для Weblate* и *Фоновые задачи с использованием Celery*. В зависимости от размера вашей установки вы можете захотеть запустить эти компоненты на выделенных серверах.

Инструкции по установке на местах:

```
# 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
```

Модули Python

Подсказка: Для установки Weblate мы используем виртуальное окружение, предоставляемое пакетом virtualenv, чтобы отделить Weblate от вашей системы. Если вы не знакомы с этим понятием, обратитесь к документации пакета.

1. Создайте виртуальное окружение для Weblate:

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

2. Активируйте виртуальное окружение для Weblate:

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

3. Установите Weblate, включая все его зависимости:

```
pip install Weblate
```

4. Установите драйвер базы данных:

```
pip install psycopg2-binary
```

5. Установите желаемые необязательные зависимости в зависимости от функций, которые вы собираетесь использовать (некоторые из них могут потребовать дополнительных системных библиотек, проверьте раздел *Необязательные зависимости*):

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

Настройка Weblate

Примечание: Следующие шаги подразумевают, что виртуальное окружение, используемое Weblate'ом, активно (активировать его можно командой `. ~/weblate-env/bin/activate`). В случае, если это не так, вам придётся указывать полный путь к команде **`weblate`**: `~/weblate-env/bin/weblate`.

1. Скопируйте файл `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` в `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. По своему усмотрению подкорректируйте значения в новом файле `settings.py`. Для тестирования вы можете использовать поставляемый пример, но для рабочей среды вы захотите внести в него изменения, смотрите раздел *Изменение конфигурации под свои нужды*.
3. Создайте базу данных и её структуру для Weblate (пример настройки использует PostgreSQL, для установки в рабочую среду ознакомьтесь разделом *Настройка базы данных для Weblate*):

```
weblate migrate
```

4. Создайте учётную запись администратора и скопируйте сгенерированный ей пароль в буфер обмена, также сохраните его для дальнейшего использования:

```
weblate createadmin
```

5. Соберите статические файлы для веб-сервера (смотрите разделы *Запуск сервера* и *Обслуживание статических файлов*):

```
weblate collectstatic
```

6. Сожмите файлы JavaScript и CSS (необязательно, смотрите раздел *Сжатие клиентских ресурсов*):

```
weblate compress
```

7. Запустите рабочие процессы Celery. Для целей разработки это делать не обязательно, но настоятельно рекомендуется. Дополнительную информацию смотрите в разделе *Фоновые задачи с использованием Celery*:

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

8. Запустите сервер разработки (шаги для запуска рабочего сервера смотрите в разделе *Запуск сервера*):

```
weblate runserver
```

После установки

Поздравляем, теперь ваш сервер Weblate запущен и вы можете начать его использовать.

- Теперь вы можете получить доступ к Weblate по адресу `http://localhost:8000/`.
- Войдите в систему с учётными данными администратора, полученными во время установки, или зарегистрируйтесь в качестве нового пользователя.
- Теперь при помощи команды **`weblate`** и когда активно виртуальное окружение Weblate вы можете запускать его команды, смотрите раздел *Команды управления*.
- Остановить тестовый сервер можно комбинацией `Ctrl+C`.
- Просмотрите список вероятных проблем с вашей установкой или по адресу `/manage/performance/` в URL, или с помощью команды **`weblate check --deploy`**; см. также *Рабочая среда*.

Добавление перевода

1. Откройте интерфейс администратора (`http://localhost:8000/create/project/`) и создайте проект, который вы хотите перевести. Подробнее смотрите в разделе *Настройки проекта*.

   Всё, что вам нужно здесь указать — это название проекта и его веб-сайт.

2. Создайте компонент, который является реальным объектом для перевода — он указывает на репозиторий системы контроля версий и задаёт файлы для перевода. Подробнее смотрите в разделе *Настройки компонента*.

   Здесь важны следующие поля: название компонента, адрес репозитория системы контроля версий и маска для поиска переводимых файлов. Weblate поддерживает широкий спектр форматов, включая PO-файлы gettext, строки ресурсов Android'а, строки свойств iOS, файлы свойств Java или Qt Linguist, подробнее смотрите в разделе *Поддерживаемые форматы файлов*.

3. После того, как вышеперечисленные действия будут завершены (это может быть довольно длительный процесс в зависимости от размера вашего репозитория системы контроля версий и количества сообщений для перевода), вы сможете начать перевод.

Установка на macOS

Требования к оборудованию

Weblate должен без проблем работать на любом современном оборудовании, ниже приведена минимальная конфигурация, необходимая для запуска Weblate на одном сервере (сам Weblate, база данных и веб-сервер):

- 2 ГБ оперативной памяти
- 2-х ядерный процессор
- 1 ГБ дискового пространства

Чем больше памяти, тем лучше — она используется для кэширования на всех уровнях (на уровне файловой системы, уровне базы данных и уровне Weblate).

Множество одновременно работающих пользователей увеличивают количество необходимых ядер процессора. Для сотен компонентов перевода рекомендуется не менее 4 Гб оперативной памяти.

Типовое использование дискового пространства базой данной находится в районе 300MB на 1 миллион хранимых слов. Пространство необходимое для клонирования репозиториев разнится, хотя Weblate и пытается поддерживать их размер минимальным, делая поверхностные (shallow) копии.

Примечание: Фактические требования к вашей установке Weblate сильно зависят от размера управляемых ею переводов.

Установка

Системные требования

Установите зависимости, необходимые для сборки модулей Python (смотрите раздел *Требования к программному обеспечению*):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
pip3 install virtualenv
```

Убедитесь, что pip сможет найти версию `libffi`, предоставляемую homebrew — она понадобится на этапе сборки установки.

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

Установите желаемые необязательные зависимости в зависимости от функций, которые вы собираетесь использовать (смотрите раздел *Необязательные зависимости*):

```
brew install tesseract
```

При желании установите программное обеспечение для запуска рабочего сервера, смотрите разделы *Запуск сервера*, *Настройка базы данных для Weblate* и *Фоновые задачи с использованием Celery*. В зависимости от размера вашей установки вы можете захотеть запустить эти компоненты на выделенных серверах.

Инструкции по установке на местах:

```
# 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
```

Модули Python

Подсказка: Для установки Weblate мы используем виртуальное окружение, предоставляемое пакетом virtualenv, чтобы отделить Weblate от вашей системы. Если вы не знакомы с этим понятием, обратитесь к документации пакета.

1. Создайте виртуальное окружение для Weblate:

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

2. Активируйте виртуальное окружение для Weblate:

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

3. Установите Weblate, включая все его зависимости:

```
pip install Weblate
```

4. Установите драйвер базы данных:

```
pip install psycopg2-binary
```

5. Установите желаемые необязательные зависимости в зависимости от функций, которые вы собираетесь использовать (некоторые из них могут потребовать дополнительных системных библиотек, проверьте раздел *Необязательные зависимости*):

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

Настройка Weblate

Примечание: Следующие шаги подразумевают, что виртуальное окружение, используемое Weblate'ом, активно (активировать его можно командой `. ~/weblate-env/bin/activate`). В случае, если это не так, вам придётся указывать полный путь к команде **`weblate`**: `~/weblate-env/bin/weblate`.

1. Скопируйте файл `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` в `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. По своему усмотрению подкорректируйте значения в новом файле `settings.py`. Для тестирования вы можете использовать поставляемый пример, но для рабочей среды вы захотите внести в него изменения, смотрите раздел *Изменение конфигурации под свои нужды*.
3. Создайте базу данных и её структуру для Weblate (пример настройки использует PostgreSQL, для установки в рабочую среду ознакомьтесь разделом *Настройка базы данных для Weblate*):

```
weblate migrate
```

4. Создайте учётную запись администратора и скопируйте сгенерированный ей пароль в буфер обмена, также сохраните его для дальнейшего использования:

```
weblate createadmin
```

5. Соберите статические файлы для веб-сервера (смотрите разделы *Запуск сервера* и *Обслуживание статических файлов*):

```
weblate collectstatic
```

6. Сожмите файлы JavaScript и CSS (необязательно, смотрите раздел *Сжатие клиентских ресурсов*):

```
weblate compress
```

7. Запустите рабочие процессы Celery. Для целей разработки это делать не обязательно, но настоятельно рекомендуется. Дополнительную информацию смотрите в разделе *Фоновые задачи с использованием Celery*:

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

8. Запустите сервер разработки (шаги для запуска рабочего сервера смотрите в разделе *Запуск сервера*):

```
weblate runserver
```

После установки

Поздравляем, теперь ваш сервер Weblate запущен и вы можете начать его использовать.

- Теперь вы можете получить доступ к Weblate по адресу `http://localhost:8000/`.
- Войдите в систему с учётными данными администратора, полученными во время установки, или зарегистрируйтесь в качестве нового пользователя.
- Теперь при помощи команды **`weblate`** и когда активно виртуальное окружение Weblate вы можете запускать его команды, смотрите раздел *Команды управления*.
- Остановить тестовый сервер можно комбинацией `Ctrl+C`.
- Просмотрите список вероятных проблем с вашей установкой или по адресу `/manage/performance/` в URL, или с помощью команды **`weblate check --deploy`**; см. также *Рабочая среда*.

Добавление перевода

1. Откройте интерфейс администратора (`http://localhost:8000/create/project/`) и создайте проект, который вы хотите перевести. Подробнее смотрите в разделе *Настройки проекта*.

   Всё, что вам нужно здесь указать — это название проекта и его веб-сайт.

2. Создайте компонент, который является реальным объектом для перевода — он указывает на репозиторий системы контроля версий и задаёт файлы для перевода. Подробнее смотрите в разделе *Настройки компонента*.

   Здесь важны следующие поля: название компонента, адрес репозитория системы контроля версий и маска для поиска переводимых файлов. Weblate поддерживает широкий спектр форматов, включая PO-файлы gettext, строки ресурсов Android'а, строки свойств iOS, файлы свойств Java или Qt Linguist, подробнее смотрите в разделе *Поддерживаемые форматы файлов*.

3. После того, как вышеперечисленные действия будут завершены (это может быть довольно длительный процесс в зависимости от размера вашего репозитория системы контроля версий и количества сообщений для перевода), вы сможете начать перевод.

Установка из исходников

1. Пожалуйста, сперва проследуйте инструкциям по установке для вашей системы:

   - *Установка на Debian и Ubuntu*
   - *Установка на SUSE и openSUSE*
   - *Установка на RedHat, Fedora и CentOS*

2. Получите из Git'а последние исходные коды Weblate'а (или скачайте архив и распакуйте его):

   ```
   git clone https://github.com/WeblateOrg/weblate.git weblate-src
   ```

   В качестве альтернативы вы можете использовать архивы выпусков. Вы можете скачать их с нашего сайта <https://weblate.org/>. Эти файлы подписаны криптографической подписью, пожалуйста, смотрите раздел *Проверка подписей выпусков*.

3. Установите текущий код Weblate'а в виртуальное окружение:

   ```
   . ~/weblate-env/bin/activate
   pip install -e weblate-src
   ```

4. Скопируйте файл `weblate/settings_example.py` в `weblate/settings.py`.

5. По своему усмотрению подкорректируйте значения в новом файле `settings.py`. Для тестирования вы можете использовать поставляемый пример, но для рабочей среды вы захотите внести в него изменения, смотрите раздел *Изменение конфигурации под свои нужды*.
6. Создайте используемую Weblate'ом базу данных, подробности смотрите в разделе *Настройка базы данных для Weblate*.
7. Создайте Django'вские таблицы, соберите статические файлы и заполните исходные данные (смотрите разделы *Наполнение базы данных* и *Обслуживание статических файлов*):

```
weblate migrate
weblate collectstatic
weblate compress
weblate compilemessages
```

Примечание: Этот шаг следует повторять при обновлении репозитория.

Установка на OpenShift

С помощью OpenShift-шаблона Weblate вы можете обзавестись своим личным экземпляром Weblate за считаные секунды. Все зависимости Weblate уже включены в образ. В качестве базы данных по умолчанию настроена PostgreSQL, а также используются устойчивые заявки на тома (persistent volume claim).

Вы можете найти этот шаблон здесь: <https://github.com/WeblateOrg/openshift/>.

Установка

В следующих примерах предполагается, что у вас есть рабочее окружение OpenShift v3.x, в котором установлена клиентская утилита `oc`. Для получения конкретных инструкций ознакомьтесь с документацией OpenShift.

Веб-консоль

Скопируйте содержимое из файла `template.yml` и импортируйте их в свой проект, затем создайте приложение с помощью кнопки `Create` (Создать) в веб-консоли OpenShift. Веб-консоль предложит вам изменить значения параметров, используемых в шаблоне.

Командная строка

Чтобы загрузить шаблон Weblate в свою текущую библиотеку проектов, передайте файл `template.yml` следующей команде:

```
$ oc create -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
   -n <PROJECT>
```

Теперь шаблон можно выбрать в веб-консоли или через интерфейс командной строки.

Параметры

Список параметров, которые вы можете переопределить приведён в соответствующей секции шаблона. Вы можете посмотреть их из командной строки (указав необходимый файл) следующим образом:

```
$ 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
```

Развёртывание

Вы также можете использовать интерфейс командной строки, чтобы обработать шаблоны и использовать получившуюся конфигурацию, чтобы создать все объекты немедленно.

```
$ 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
```

После удачной миграции и развёртывания Weblate должен быть доступен по указанному адресу в параметре `WEBLATE_SITE_DOMAIN`.

После настройки контейнера вы можете войти в систему под пользователем *admin* с паролем, указанным в `WEBLATE_ADMIN_PASSWORD`, или случайным паролем, сгенерированным при первом запуске, если он не был установлен явно.

Для сброса пароля пользователя *admin* перезапустите контейнер с переменной *WEBLATE_ADMIN_PASSWORD*, установленной в новое значение пароля в соответствующем хранилище секретов (`Secrete`).

Удаление

```
$ 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
```

Конфигурация

При обработке шаблона будет создан соответствующий `ConfigMap`, который можно использовать для настройки образа Weblate. Этот `ConfigMap` напрямую монтируется в качестве переменных среды. Он вызывает новую развёртку, каждый раз, когда изменяется. Информацию по другим параметрам настройки и полный список переменных среды смотрите в *описании настройки Docker*.

Установка на Kubernetes

Примечание: Для наполнения данного раздела руководства большим количеством деталей разыскиваются участники, имеющие опыт работы с Kubernetes.

С помощью схемы Kubernetes Helm вы можете обзавестись своим личным экземпляром Weblate за считаные секунды. Все зависимости Weblate уже включены в образ. В качестве базы данных по умолчанию настроена PostgreSQL, а также используются устойчивые заявки на тома (persistent volume claim).

Вы можете найти схему на <https://github.com/WeblateOrg/helm/> , а просмотреть её можно на <https://artifacthub.io/packages/helm/weblate/weblate>.

Установка

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

В зависимости от ваших настроек и опыта выберите подходящий для вас способ установки:

- *Установка с помощью Docker'а*, рекомендуется для установки на рабочих серверах.
- Установка в виртуальное окружение, рекомендуется для рабочей среды:
  - *Установка на Debian и Ubuntu*
  - *Установка на SUSE и openSUSE*
  - *Установка на RedHat, Fedora и CentOS*
  - *Установка на macOS*
- *Установка из исходников*, рекомендуется для разработки.
- *Установка на OpenShift*
- *Установка на Kubernetes*

2.1.2 Требования к программному обеспечению

Операционная система

Weblate работает под Linux, FreeBSD и MacOS. Скорее всего он будет работать и под другими Unix-подобными системами.

Работа Weblate под Windows не поддерживается. Тем не менее, работать под ней он всё ещё может и патчи под неё с радостью принимаются.

Другие сервисы

Для своей работы Weblate использует другие сервисы. Вам понадобятся, по крайней мере, следующие запущенные сервисы:

- Сервер базы данных PostgreSQL, смотрите раздел *Настройка базы данных для Weblate*.
- Сервер Redis для кэша и очереди задач, смотрите раздел *Фоновые задачи с использованием Celery*.
- SMTP-сервер для исходящей почты, смотрите раздел *Настройка исходящей почты*.

Зависимости Python

Weblate написан на Python'е и поддерживает Python 3.6 или более новую версию. Вы можете установить зависимости с помощью команды pip или из пакетов вашего дистрибутива, полный список зависимостей находится в файле `requirements.txt`.

Основные зависимости:

Django https://www.djangoproject.com/

Celery https://docs.celeryproject.org/

Translate Toolkit https://toolkit.translatehouse.org/

translation-finder https://github.com/WeblateOrg/translation-finder

Python Social Auth https://python-social-auth.readthedocs.io/

Фреймворк Django REST https://www.django-rest-framework.org/

Необязательные зависимости

Для работы некоторых функций Weblate необходимы следующие модули. Полный их список можно найти в файле `requirements-optional.txt`.

`Mercurial` (необязательная зависимость для поддержки репозиториев Mercurial) https://www.mercurial-scm.org/

`phply` (необязательная зависимость для поддержки PHP) https://github.com/viraptor/phply

`tesserocr` (необязательная зависимость для поддержки распознавания текста со снимков экрана) https://github.com/sirfz/tesserocr

`akismet` (необязательная зависимость для защиты предложений от спама) https://github.com/ubernostrum/akismet

`ruamel.yaml` (необязательная зависимость для поддержки *файлов YAML*) https://pypi.org/project/ruamel.yaml/

`Zeep` (необязательная зависимость для поддержки *терминологической службы Майкрософт*) https://docs.python-zeep.org/

`aeidon` (необязательная зависимость для поддержки *файлов субтитров*) https://pypi.org/project/aeidon/

Зависимости серверной части базы данных

Weblate поддерживает PostgreSQL, MySQL и MariaDB, подробнее смотрите в разделе *Настройка базы данных для Weblate* и документации по реализации поддержки соответствующей базы данных.

Другие системные требования

В системе должны быть установлены следующие зависимости:

`Git` https://git-scm.com/

Pango, Cairo и связанные заголовочные файлы и данные интроспекции gir https://cairographics.org/, https://pango.gnome.org/, смотрите раздел *Pango и Cairo*

`git-review` (необязательная зависимость для поддержки Gerrit) https://pypi.org/project/git-review/

`git-svn` (необязательная зависимость для поддержки Subversion) https://git-scm.com/docs/git-svn

`tesseract` и его данные (необязательная зависимость для поддержки распознавания текста со снимков экрана) https://github.com/tesseract-ocr/tesseract

`licensee` (необязательная зависимость для определения лицензии при создании компонента) https://github.com/licensee/licensee

Зависимости времени компиляции

Для сборки некоторых *зависимостей Python* может потребоваться установка также и их зависимостей. Это зависит от того, как именно вы их устанавливаете, так что за справкой по установке обратитесь к документации соответствующих пакетов. Вручную устанавливать зависимости зависимостей вам не понадобятся, если вы, например, устанавливаете их через `pip` и используете предварительно-собранные «колёса» (`Wheels`) или устанавливаете их через пакетный менеджер вашего дистрибутива.

Pango и Cairo

Изменено в версии 3.7.

Weblate использует Pango и Cairo для отрисовки растровых виджетов (смотрите раздел promotion) и проверки отрисовки текста (смотрите раздел *Управление шрифтами*). Для правильной установки привязок Python сначала нужно установить системные библиотеки — вам нужны и Cairo, и Pango, которым, в свою очередь, нужна GLib. Все они должны быть установлены с файлами для разработки и данными интроспекции GObject.

2.1.3 Проверка подписей выпусков

Выпуски Weblate подписываются криптографической подписью выпускающего разработчика. В настоящее время им является Михал Чигарж. Отпечаток его PGP-ключа:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

а дополнительную идентификационную информацию вы можете получить по адресу <https://keybase.io/nijel>.

Вы должны удостовериться, что подпись соответствует скачанному вами архиву. Таким образом, вы сможете быть уверены, что используете тот же код, который был выпущен. Также вы должны проверить дату подписания, чтобы удостовериться, что вы скачали последнюю версию.

Каждый архив сопровождается файлами `.asc`, содержащими его PGP-подпись. После того, как вы разместите оба файла в одном каталоге, вы сможете проверить подпись:

```
$ 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
```

Как видите, GPG жалуется на то, что он не знает открытого ключа. На данный момент вы должны сделать одно из следующего:

- Использовать *wkd* для скачивания ключа:

```
$ gpg --auto-key-locate wkd --locate-keys michal@cihar.com
pub   rsa4096 2009-06-17 [SC]
      63CB1DF1EF12CF2AC0EE5A329C27B31342B7511D
uid           [ultimate] Michal Čihař <michal@cihar.com>
uid           [ultimate] Michal Čihař <nijel@debian.org>
uid           [ultimate] [jpeg image of size 8848]
uid           [ultimate] Michal Čihař (Braiins) <michal.cihar@braiins.cz>
sub   rsa4096 2009-06-17 [E]
sub   rsa4096 2015-09-09 [S]
```

- Скачать связку ключей с сервера Михаила, а затем импортировать её командой:

```
$ gpg --import wmxth3chu9jfxdxywj1skpmhsj311mzm
```

- Скачать и импортировать ключ с одного из серверов ключей:

```
$ 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
```

Это немного улучшит ситуацию — на данный момент вы можете проверить правильность подписи с данного ключа, но всё равно не можете доверять используемому в ключе имени:

```
$ 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
```

Проблема в том, что выдать ключ с таким именем может кто угодно. Вам нужно убедиться, что ключ действительно принадлежит указанному лицу. Руководство GNU по обеспечению конфиденциальности раскрывает эту тему в главе Проверка достоверности ключей. Самый надёжный метод — лично встретиться с разработчиком и обменяться отпечатками ключей, однако вы также можете положиться на сеть доверия. Таким образом, вы можете транзитивно доверять ключу через подписи других людей, которые встречались с разработчиком лично.

Когда достоверность ключа будет установлена, предупреждение больше появляться не будет:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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]
```

В случае недействительной подписи (архив был изменён), вы получите явную ошибку, независимо от того, доверяете ли вы ключу или нет:

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

2.1.4 Права доступа к файлам

Процесс Weblate должен мочь читать и писать в каталог, в котором он хранит свои данные — `DATA_DIR`. Все файлы в этом каталоге должны принадлежать и иметь разрешение на запись для пользователей, от имени которых работают все процессы Weblate (в частности WSGI и Celery, см. *Запуск сервера* и *Фоновые задачи с использованием Celery*).

В конфигурации по умолчанию они расположены в том же дереве, что и исходный код Weblate, однако вы можете предпочесть переместить их в более подходящее место, например, в `/var/lib/weblate`.

Weblate попытается автоматически создать эти каталоги, но у него ничего не получится, если у него не будет соответствующих прав доступа.

Также вам следует быть особенно внимательными, когда запускаете *команды управления*, поскольку они должны выполняться под тем же пользователем, под которым запущен сам Weblate. В противном случае права доступа на некоторые файлы могут быть выставлены неверно.

В контейнере Docker все файлы в томе `/app/data` должны принадлежать пользователю `weblate` внутри самого контейнера (UID 1000).

См.также:

Обслуживание статических файлов

2.1.5 Настройка базы данных для Weblate

Рекомендуется запускать Weblate с сервером баз данных PostgreSQL.

См.также:

Использование мощного движка базы данных, Базы данных, *Переход с других баз данных на PostgreSQL*

PostgreSQL

PostgreSQL, как правило, является лучшим выбором для сайтов, написанных на Django. Это эталонная база данных, используемая для реализации слоя баз данных Django.

Примечание: Weblate использует расширение для триграмм, которое в некоторых случаях должно быть установлено отдельно. Ищите пакет `postgresql-contrib` или аналогичный.

См.также:

PostgreSQL notes

Создание базы данных в PostgreSQL

Обычно рекомендуется запускать Weblate в отдельной базе данных и под отдельной учётной записью:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser --superuser --pwprompt weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -O weblate weblate
```

Подсказка: Если вы не хотите делать пользователя Weblate суперпользователем в PostgreSQL, вы можете пропустить этот шаг. В этом случае вам придётся выполнить некоторые шаги по миграции вручную, так как суперпользователь PostgreSQL запускает в схеме Weblate следующую команду:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
```

Настройка Weblate для использования PostgreSQL

Фрагмент настроек `settings.py` для PostgreSQL:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    }
}
```

При миграции базы данных выполняется ALTER ROLE той роли, которую используется Weblate. В большинстве случаев имя роли совпадает с именем пользователя. В более же сложных случаях они могут отличаться и при миграции вы можете получить ошибку о том, что такая роль не существует (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). Известно, что такое случается при использовании базы данных Azure совместно с PostgreSQL, хотя это может происходить и при других обстоятельствах. Установите параметр `ALTER_ROLE`, чтобы задать ту роль, которую Weblate должен изменить во время миграции базы данных.

MySQL и MariaDB

Подсказка: Некоторые функции Weblate лучше работают с *PostgreSQL*. Среди них: поиск и память переводов, обе из которых используют функции полнотекстового поиска базы данных, а их реализация в PostgreSQL превосходит все остальные.

Weblate также можно использовать с MySQL или MariaDB, ознакомитесь с предостережениями по их использованию вместе с Django в разделах Замечания по MySQL и Замечания по MariaDB его документации. Из-за ограничений рекомендуется использовать :ref:`postgresql`при новой установке.

Weblate требует MySQL версии 5.7.8 или новее или MariaDB 10.2.7 или новее.

Для Weblate рекомендуется следующая конфигурация:

- Используйте кодовую таблицу `utf8mb4` для представления символов с более высоких плоскостей Юникода (например, эмодзи).
- Настройте сервер на использование `innodb_large_prefix` для разрешения более длинных индексов для текстовых полей.
- Установите уровень изоляции в `READ COMMITTED`.
- Режим SQL должен быть установлен в `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.

Ниже приведён пример типового файла настроек `/etc/my.cnf.d/server.cnf` для сервера с 8ГБ ОЗУ. Таких настроек должно хватить для большинства установок. У MySQL и MariaDB есть и другие параметры, которые могут повысить производительность сервера, но обычно прибегать к ним нет смысла, если только вы не планируете, что достаточно большое количество пользователей будут работать с вашему сервером одновременно. В таком случае смотрите подробности по конфигурации сервера для таких задач в документации своей СУБД.

Дабы при установке Weblate не возникло проблем, абсолютно критично, чтобы параметр `innodb_file_per_table` был включён; а также после его изменения нужно не забыть перезапустить MySQL/MariaDB.

```
[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
```

Подсказка: Если вы получаете ошибку `#1071 - Specified key was too long; max key length is 767 bytes` (указанный ключ слишком длинный; максимальная длина ключа - 767 байт), то убедитесь, что в конфигурации вашего сервера включены параметры `innodb`, описанные выше.

Подсказка: В случае, если вы получаете ошибку `#2006 - MySQL server has gone away` (сервер MySQL более недоступен), то может помочь настройка параметра `CONN_MAX_AGE`.

Настройка Weblate для работы совместно с MySQL/MariaDB

Фрагмент файла настроек `settings.py` для MySQL и 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": {},
    }
}
```

Также прежде чем начать установку вам нужно будет создать пользователя базы данных с именем `weblate`. Это можно сделать с помощью следующей команды:

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

2.1.6 Другие настройки

Настройка исходящей почты

Weblate отправляет электронные письма по разным поводам — для активации учётной записи и по различным уведомлениям, настроенным пользователями. Для этого ему необходим доступ к SMTP-серверу.

Настройка почтового сервера производится с помощью следующих параметров: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_TLS`, `EMAIL_USE_SSL`, `EMAIL_HOST_USER` и `EMAIL_PORT`. Их имена говорят сами за себя, а более подробную информацию вы можете найти в документации Django.

Подсказка: Если вы получаете ошибку о том , что аутентификация не поддерживается (например, `SMTP AUTH extension not supported by server`, «расширение SMTP AUTH не поддерживается сервером»), то скорей всего это вызвано тем, что используется небезопасное соединение и из-за этого сервер отказывается производить аутентификацию. В таком случае попробуйте включить `EMAIL_USE_TLS`.

См.также:

Не доходит электронная почта от Weblate, *Настройка исходящего адреса электронной почты внутри контейнера Docker*

Работа за обратным прокси

Некоторые функции Weblate полагаются на возможность получения IP-адреса клиента. Среди этих функций *Ограничение частоты запросов*, *Защита от спама* или *Журнал аудита*.

В стандартной конфигурации Weblate достаёт IP-адрес из переменной `REMOTE_ADDR`, которая устанавливается обработчиком WSGI.

Если же вы используете обратный прокси, то это поле, скорее всего, будет содержать его адрес. Вам нужно настроить Weblate на доверие дополнительным HTTP-заголовкам и доставать IP-адрес из них. Эту настройку нельзя включить по умолчанию, поскольку она разрешит подмену IP-адреса для установок, не использующих обратный прокси. Для самых обычных установок будет достаточно включения параметра `IP_BEHIND_REVERSE_PROXY`, но вам также может понадобиться подправить параметры `IP_PROXY_HEADER` и `IP_PROXY_OFFSET`.

См.также:

Защита от спама, *Ограничение частоты запросов*, *Журнал аудита*, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`, `SECURE_PROXY_SSL_HEADER`

HTTP-прокси

Weblate запускает команды системы контроля версий, а те берут настройки прокси из переменных окружения. Для их настройки рекомендуемый подход — определить необходимые параметры прокси в файле `settings.py`:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

См.также:

Переменные окружения для прокси

2.1.7 Изменение конфигурации под свои нужды

См.также:

Пример файла настроек

Скопируйте `weblate/settings_example.py` в `weblate/settings.py` и подправьте его в соответствии со своими настройками. Скорее всего, вы захотите настроить следующие параметры: `ADMINS`

> Список администраторов сайтов для получения уведомлений, когда что-то идёт не так, например, для получения уведомлений о неудачных слияниях, или об ошибках Django.
>
> **См.также:**
>
> `ADMINS`

`ALLOWED_HOSTS`

> Вы должны установить эту настройку в список хостов, которые должен обслуживать ваш сайт. Например:
>
> ```
> ALLOWED_HOSTS = ["demo.weblate.org"]
> ```
>
> Или же вы можете указать символ шаблона:
>
> ```
> ALLOWED_HOSTS = ["*"]
> ```

См.также:

ALLOWED_HOSTS, *WEBLATE_ALLOWED_HOSTS*, *Настройка разрешённых хостов*

SESSION_ENGINE

Настраивает то, как будут храниться ваши сессии. В случае, если вы оставили механизм хранения по умолчанию — в базе данных — вы должны настроить расписание для команды **weblate clearsessions**, удаляющей из базы данных данные об устаревших сессиях.

Если вы используете в качестве кэша Redis (смотрите раздел *Включение кэширования*), рекомендуется использовать его также и для сессий:

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

См.также:

Настройка движка сессий, SESSION_ENGINE

DATABASES

Подключение к серверу базы данных, подробнее об этой настройке читайте в документации Django.

См.также:

Настройка базы данных для Weblate, DATABASES, Базы данных

DEBUG

Отключите эту настройку для любого рабочего сервера. При включённом режиме отладки Django в случае возникновения ошибки будет показывать трассировки стека пользователям, при его же отключении ошибки будут отправляться на адреса электронной почты из массива ADMINS (смотрите выше).

Кроме того, отладочный режим замедляет работу Weblate, поскольку в этом случае Django сохраняет внутри себя гораздо больше информации.

См.также:

DEBUG

DEFAULT_FROM_EMAIL

Адрес электронной почты отправителя для исходящих сообщений электронной почты, например, для писем регистрации.

См.также:

DEFAULT_FROM_EMAIL

SECRET_KEY

Ключ, используемый Django для подписывания некоторой информации, сохраняемой в куках, подробнее смотрите в разделе *Секретный ключ Django*.

См.также:

SECRET_KEY

SERVER_EMAIL

Адрес электронной почты, используемый в качестве адреса отправителя для отправки сообщений администратору, например, уведомлений о неудачных слияниях.

См.также:

SERVER_EMAIL

2.1.8 Наполнение базы данных

После того, как ваша конфигурация будет готова, вы можете запустить `weblate migrate` для создания структуры базы данных. Теперь вы можете создавать проекты перевода через интерфейс администратора.

В случае, если вы хотите запустить установку не в интерактивном режиме, вы можете использовать `weblate migrate --noinput`, а затем создать пользователя-администратора при помощи команды *`createadmin`*.

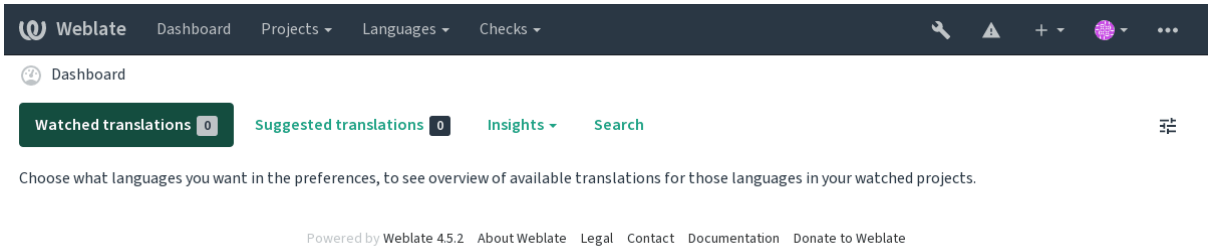
После того, как вы закончите, вам также стоит проверить *Отчёт о производительности* в интерфейсе администратора. Информация оттуда может дать вам подсказки по исправлению потенциально не оптимальной конфигурации вашего сайта.

См.также:

Конфигурация, *Список привилегий*

2.1.9 Рабочая среда

Для установки в рабочую среду необходимо провести корректировки, описанные в следующих разделах. Наиболее критичные настройки вызовут предупреждение, которое обозначается восклицательным знаком в верхней панели, если вы вошли в систему под суперпользователем:



Также рекомендуется изучить все проверки, произведённые Django (хотя, возможно, исправлять их все нет необходимости):

```
weblate check --deploy
```

Вы можете просмотреть этот же список из *интерфейса управления*.

См.также:

Контрольный список развертывания

Отключение отладочного режима

Отключите отладочный режим Django (*`DEBUG`*), установив:

```
DEBUG = False
```

При включённом отладочном режиме Django сохраняет все выполненные запросы и при возникновении ошибок показывает пользователям обратную трассировку стека, что в рабочей среде не желательно.

См.также:

Изменение конфигурации под свои нужды

Правильная настройка администраторов

Укажите в настройке *ADMINS* правильные адреса администраторов, чтобы определить, кто будет получать электронную почту в случае, если на сервере что-то пойдёт не так, например:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

См.также:

Изменение конфигурации под свои нужды

Установка правильного домена сайта

Поправьте имя сайта и домен в интерфейсе администратора, иначе ссылки в RSS или регистрационные письма не будут работать. Эта настройка осуществляется с помощью параметра *SITE_DOMAIN*, который должен содержать доменное имя сайта.

Изменено в версии 4.2: До выхода версии 4.2 вместо него использовался фреймворк «сайтов» Django, смотрите раздел Фреймворк «сайтов».

См.также:

Настройка разрешённых хостов, *Правильная настройка HTTPS* *SITE_DOMAIN*, *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

Правильная настройка HTTPS

Настоятельно рекомендуется запускать Weblate с использованием шифрующего протокола HTTPS. После его включения в настройках необходимо установить параметр *ENABLE_HTTPS*:

```
ENABLE_HTTPS = True
```

Подсказка: Возможно, вы также захотите настроить HSTS, для получения более подробной информации смотрите документ SSL/HTTPS.

См.также:

ENABLE_HTTPS, *Настройка разрешённых хостов*, *Установка правильного домена сайта*

Установка правильного значения 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
```

Если для него установлено ненулевое целое значение, то `django.middleware.security.SecurityMiddleware` устанавливает заголовок HTTP Strict Transport Security у всех ответов, которые его ещё не имеют.

Предупреждение: Неправильная настройка этого параметра может привести к необратимому (на некоторое время) выходу из строя вашего сайта. Поэтому сперва ознакомьтесь с документацией по HTTP Strict Transport Security.

Использование мощного движка базы данных

Пожалуйста, используйте для рабочего окружения PostgreSQL, для получения более подробной информации смотрите раздел *Настройка базы данных для Weblate*.

См.также:

Настройка базы данных для Weblate, *Переход с других баз данных на PostgreSQL*, *Изменение конфигурации под свои нужды*, Базы данных

Включение кэширования

Если возможно, используйте Redis из Django, настроив переменную `CACHES` в файле настроек, например:

```
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",
        },
    }
}
```

Подсказка: Если вы измените настройки кеширования Redis, то вам также понадобится изменить их и в Celery, см. *Фоновые задачи с использованием Celery*.

См.также:

Кеширование аватаров, Фреймворк кэширования Django

Кеширование аватаров

Помимо кэширования силами Django, Weblate выполняет кэширование аватаров. Для этой цели рекомендуется воспользоваться отдельным файловым кэшем:

```
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,
        },
    },
}
```

См.также:

ENABLE_AVATARS, *AVATAR_URL_PREFIX*, *Аватары*, *Включение кэширования*, Фреймворк кэширования Django

Настройка отправки электронной почты

Weblate'у по разным случаям необходимо отправлять электронные письма и эти письма должны иметь правильный адрес отправителя, поэтому настройте параметры *SERVER_EMAIL* и *DEFAULT_FROM_EMAIL* так, чтобы они соответствовали вашему окружению, например:

```
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"
```

Примечание: Для отключения отправки писем Weblate'ом установите переменную EMAIL_BACKEND в значение `django.core.mail.backends.dummy.EmailBackend`.

При этом доставка сообщений электронной почты будет отключена *целиком*, включая сообщения для регистрации или сброса пароля.

См.также:

Изменение конфигурации под свои нужды, *Настройка исходящей почты*, EMAIL_BACKEND, DEFAULT_FROM_EMAIL, SERVER_EMAIL

Настройка разрешённых хостов

Django требует, чтобы в параметре *ALLOWED_HOSTS* содержался список доменных имён, которые разрешено обслуживать вашему сайту, если его оставить пустым, любые запросы будут заблокированы.

В случае, если он не настроен на соответствие вашему HTTP-серверу, вы будете получать ошибки вида `Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS.` (`Неверный заголовок HTTP_HOST: '1.1.1.1'. Возможно, вам нужно добавить '1.1.1.1' в ALLOWED_HOSTS.`)

Подсказка: Внутри контейнера Docker он доступен через переменную окружения *WEBLATE_ALLOWED_HOSTS*.

См.также:

ALLOWED_HOSTS, *WEBLATE_ALLOWED_HOSTS*, *Установка правильного домена сайта*

Секретный ключ Django

Параметр *SECRET_KEY* используется Django для подписания файлов кук и вы обязательно должны сгенерировать своё собственное значение, а не использовать приведённое в примере настройки.

Сгенерировать новый ключ вы можете с помощью скрипта `weblate/examples/generate-secret-key`, поставляемого вместе с Weblate.

См.также:

SECRET_KEY

Домашний каталог

Изменено в версии 2.1: Больше он не требуется, теперь Weblate хранит все свои данные в каталоге *DATA_DIR*.

Домашний каталог для пользователя, запустившего Weblate, должен существовать и быть доступен для записи из под этого пользователя. Особенно он необходим, если вы хотите использовать SSH для доступа к частным репозиториям, но доступ к этому каталогу может понадобиться и Git'у (это зависит от используемой вами версии Git'a).

Изменить используемый Weblate каталог вы можете в файле `settings.py`, например, установив его в каталог `configuration` внутри дерева Weblate:

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

Примечание: В Linux и других UNIX-подобных системах путь к домашнему каталогу пользователя определяется в файле `/etc/passwd`. Многие дистрибутивы по умолчанию имеют не записываемый каталог для пользователей, используемый для раздачи веб-содержимого (вроде `apache`, `www-data` или ``wwwrun`), так что вам либо придётся запускать Weblate под другим пользователем, либо изменить эту настройку.

См.также:

Доступ к репозиториям

Загрузка шаблонов

Рекомендуется использовать кэширующий загрузчик шаблонов для Django. Он кэширует обработанные шаблоны, избавляя фреймворк от необходимости разбирать их каждый запрос. Настроить его вы можете с помощью следующего фрагмента кода (самое важное здесь — это настройка `loaders`):

```
TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [
            os.path.join(BASE_DIR, "templates"),
        ],
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
            "loaders": [
                (
                    "django.template.loaders.cached.Loader",
                    [
                        "django.template.loaders.filesystem.Loader",
                        "django.template.loaders.app_directories.Loader",
                    ],
                ),
            ],
        },
    },
]
```

См.также:

`django.template.loaders.cached.Loader`

Выполнение задач технического обслуживания

Для оптимальной производительности неплохо бы в фоновом режиме выполнять некоторые задачи по техническому обслуживанию. Теперь это делается автоматически с помощью *фоновых задач Celery*, которые включают в себя:

- Проверку работоспособности состояния конфигурации (ежечасно).
- Фиксацию отложенных изменений (ежечасно), смотрите раздел *Отложенные коммиты* и команду `commit_pending`.
- Обновление предупреждений компонента (ежедневно).
- Обновление удалённых веток (по ночам), смотрите параметр `AUTO_UPDATE`.
- Резервное копирование памяти переводов в JSON (ежедневно), смотрите команду `dump_memory`.
- Задачи по обслуживанию полнотекстового индекса и базы данных (ежедневные и еженедельные задачи), смотрите команду `cleanuptrans`.

Изменено в версии 3.2: Начиная с версии 3.2, по умолчанию эти задачи выполняются через Celery и Weblate уже поставляется с соответствующей конфигурацией, смотрите раздел *Фоновые задачи с использованием Celery*.

Системные локали и кодировки

Системные локали должны быть настроены на поддерживающие UTF-8. В большинстве дистрибутивов Linux это умолчательная их настройка. В случае, если в вашей системе это не так, измените пожалуйста локали на вариант с поддержкой UTF-8.

Например, путём редактирования файла `/etc/default/locale` и прописывания в нём команды `LANG="C.UTF-8"`.

В некоторых случаях отдельные сервисы имеют отдельные настройки для локалей. Например, при использовании Apache вы можете установить локаль в файле `/etc/apache2/envvars`:

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

Использование пользовательского центра сертификации

Во время HTTP-запросов Weblate проверяет SSL-сертификаты. Если вы используете пользовательский центр сертификации, который цепочке сертификатов по умолчанию не является доверенным, вам придётся добавить его сертификат в качестве доверенного.

Предпочтительный подход — сделать это на системном уровне, подробнее смотрите документацию по дистрибутиву (например, в debian это можно сделать, разместив сертификат центра сертификации в каталоге `/usr/local/share/ca-certificates/` и запустив команду **`update-ca-certificates`**).

Как только это будет сделано, системные инструменты станут доверять сертификату, среди этих инструментов будет и Git.

Для кода Python вам будет необходимо настроить запросы на использование системной цепочки сертификатов вместо использования цепочки, поставляемой с Python'ом. Этого можно добиться, поместив в `settings.py` следующий фрагмент кода (путь указан для Debian):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Сжатие клиентских ресурсов

Weblate поставляется с кучей файлов JavaScript и CSS. Из соображений производительности хорошей практикой будет сжимать их перед отправкой клиенту. В конфигурации по умолчанию это делается на лету ценой небольших накладных расходов. В больших же установках рекомендуется включить режим предварительного сжатия. Это можно сделать в настройках, а повторное сжатие файлов должно запускаться при каждом обновлении Weblate.

Такое переключение режима выполняется простым включением параметра `django.conf.settings.COMPRESS_OFFLINE` и заданием соответствующих настроек в параметре `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` (последний уже включён в пример файла настроек):

```
COMPRESS_OFFLINE = True
```

При каждом развёртывании вам необходимо сжимать файлы, чтобы они соответствовали текущей версии:

```
weblate compress
```

Подсказка: В официальном образе Docker эта функция уже включена.

См.также:

Распространённые сценарии развёртывания, *Обслуживание статических файлов*

2.1.10 Запуск сервера

Подсказка: In case you are not experienced with services described below, you might want to try *Установка с помощью Docker'а*.

Чтобы запустить Weblate, вам понадобится несколько сервисов, рекомендуемая конфигурация состоит из:

- Сервера базы данных (смотрите раздел *Настройка базы данных для Weblate*)
- Сервера кэша (смотрите раздел *Включение кэширования*)
- Веб-сервера на клиентской части для раздачи статических файлов и SSL-терминации (смотрите раздел *Обслуживание статических файлов*)
- Сервера WSGI для генерации динамического содержимого (смотрите раздел *Примеры файлов настроек NGINX и uWSGI*)
- Celery для выполнения фоновых задач (смотрите раздел *Фоновые задачи с использованием Celery*)

Примечание: Между некоторыми сервисами существуют зависимости, например, кэш и база данных должны быть запущены до запуска процессов Celery или uwsgi.

В большинстве случаев вы будете запускать все сервисы на одном (виртуальном) сервере, но в случае, если ваша установка испытывает тяжёлые нагрузки, вы можете их разделить. Единственным ограничением является то, что серверы Celery и Wsgi должны иметь доступ к каталогу `DATA_DIR`.

Примечание: Процесс WSGI должен выполняться под тем же пользователем, под которым работает Celery, иначе файлы в каталоге *DATA_DIR* будут сохраняться с разными владельцами, что приведёт к проблемам во время выполнения.

См. также *Права доступа к файлам* и *Фоновые задачи с использованием Celery*.

Запуск веб-сервера

Запуск Weblate ничем не отличается от запуска любой другой программы на базе Django. Как правило, Django выполняется как uWSGI или fcgi (смотрите ниже примеры для различных веб-серверов).

В целях тестирования можно использовать встроенный в Django веб-сервер:

```
weblate runserver
```

Предупреждение: НЕ ИСПОЛЬЗУЙТЕ ЭТОТ СЕРВЕР В РАБОЧЕЙ СРЕДЕ. Он не проходил аудиты безопасности или тесты производительности. Также смотрите документацию Django по команде `runserver`.

Подсказка: Встроенный сервер Django обслуживает статические файлы только с включённым параметром *DEBUG*, поскольку он предназначен только для разработки. Для использования в рабочей среде смотрите установки wsgi в разделах *Примеры файлов настроек NGINX и uWSGI*, *Пример файла настроек Apache*, *Пример файла настроек Apache и Gunicorn* и *Обслуживание статических файлов*.

Обслуживание статических файлов

Изменено в версии 2.4: До версии 2.4 Weblate некорректно использовал фреймворк статических файлов Django, поэтому его настройка была более сложной.

Django необходимо собрать свои статические файлы в одном каталоге. Для этого необходимо выполнить команду `weblate collectstatic --noinput`. Она скопирует статические файлы в каталог, указанный в параметре `STATIC_ROOT` (по умолчанию это каталог `static` внутри каталога *DATA_DIR*).

Статические файлы рекомендуется раздавать непосредственно с вашего веб-сервера, это следует делать для следующих путей:

/static/ Обслуживает статические файлы для Weblate и административного интерфейса (из каталога, определённого параметром `STATIC_ROOT`).

/media/ Используется для загрузки пользовательских мультимедийных данных (например, снимков экрана).

/favicon.ico Должен быть переписан, чтобы переписать правило для раздачи `/static/favicon.ico`.

См.также:

Примеры файлов настроек NGINX и uWSGI, *Пример файла настроек Apache*, *Пример файла настроек Apache и Gunicorn*, *Сжатие клиентских ресурсов*, Deploying Django, Deploying static files

Политика безопасности содержимого

В конфигурации Weblate по умолчанию уже включено использование промежуточного ПО `weblate.middleware.SecurityMiddleware`, которое устанавливает связанные с безопасностью заголовки HTTP, такие как `Content-Security-Policy` или `X-XSS-Protection`. По умолчанию они настроены для работы с Weblate и его конфигурацией, но для этого может потребоваться настройка вашего окружения.

См.также:

CSP_SCRIPT_SRC, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

Примеры файлов настроек NGINX и uWSGI

Для запуска рабочего веб-сервера используйте обёртку wsgi, устанавливаемую вместе с Weblate (в виртуальном окружении она устанавливается в `~/weblate-env/lib/python3.7/site-packages/weblate/wsgi.py`). Не забудьте также поместить ваше виртуальное окружение в пути поиска Python'а (например, используя установку `virtualenv = /home/user/weblate-env` в uWSGI).

Со следующими файлами настроек Weblate будет запускаться как uWSGI-приложение под веб-сервером NGINX.

Файл настроек NGINX (этот же пример есть в `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;
        # uwsgi_pass 127.0.0.1:8080;
    }
}
```

Файл настроек uWSGI (этот же пример есть в `weblate/examples/weblate.uwsgi.ini`):

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

# Add path to Weblate checkout if you did not install
# Weblate by pip
# python-path   = /path/to/weblate

# In case you're using virtualenv uncomment this:
virtualenv = /home/weblate/weblate-env

# Needed for OAuth/OpenID
buffer-size   = 8192

# Reload when consuming too much of memory
reload-on-rss = 250

# Increase number of workers for heavily loaded sites
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
```

См.также:

Как совместно использовать Django с uWSGI

Пример файла настроек Apache

Когда Weblate используется с WSGI, то рекомендуется использовать модуль мультипроцессовой обработки (MPM) prefork.

Со следующим файлом настроек Weblate будет запускаться как WSGI-приложение, вам нужно включить модуль `mod_wsgi` (этот же пример есть в `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
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

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

    <Directory /home/weblate/weblate-env/lib/python3.7/site-packages/weblate/>
        <Files wsgi.py>
        Require all granted
        </Files>
    </Directory>

</VirtualHost>
```

Примечание: Weblate требует Python 3, поэтому, пожалуйста, убедитесь, что вы используете вариант modwsgi, собранный с третьим Python'ом. Обычно он доступен в виде отдельного пакета, например `libapache2-mod-wsgi-py3`.

См.также:

Системные локали и кодировки, Как совместно использовать Django с Apache и mod_wsgi

Пример файла настроек Apache и Gunicorn

Со следующим файлом настроек Weblate будет запускаться в Gunicorn и Apache 2.4 (этот же пример есть в `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>
```

См.также:

Как совместно использовать Django с Gunicorn

Запуск Weblate по определённому пути в URL

Добавлено в версии 1.3.

Когда Weblate используется с WSGI, то рекомендуется использовать модуль мультипроцессовой обработки (MPM) prefork.

Пример файла настроек Apache для предоставления доступу к Weblate по пути `/weblate`. Снова используя модуль `mod_wsgi` (этот же пример есть в `weblate/examples/apache-path.conf`):

```
#
# VirtualHost for Weblate, running under /weblate path
```

(continues on next page)

(продолжение с предыдущей страницы)

```
#
# 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
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

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

    <Directory /home/weblate/weblate-env/lib/python3.7/site-packages/weblate/>
        <Files wsgi.py>
        Require all granted
        </Files>
    </Directory>

</VirtualHost>
```

Кроме того, вам нужно будет подправить `weblate/settings.py`:

```
URL_PREFIX = "/weblate"
```

2.1.11 Фоновые задачи с использованием Celery

Добавлено в версии 3.2.

Weblate использует Celery для выполнения задач в фоне. Типовая конфигурация, основанная на Redis, выглядит как-то так:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

См.также:

Настройка Redis в качестве брокера в Celery

При разработке Weblate или модулей для него вы можете использовать «нетерпеливую» конфигурацию, тогда все задачи будут выполнятся непосредственно при их вызове, хотя это и ударит по производительности:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

Также вы должны запустить рабочего Celery для обработки задач и запуска запланированных задач, это можно сделать непосредственно в командной строке (что в основном полезно при отладке или разработке):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Примечание: Процесс Celery должен выполняться под тем же пользователем, под которым работает процесс WSGI, иначе файлы в каталоге `DATA_DIR` будут сохраняться с разными владельцами, что приведёт к проблемам во время выполнения.

См. также *Права доступа к файлам* и *Запуск сервера*.

Запуск Celery в качестве системного сервиса

Скорее всего вы захотите запустить Celery в качестве демона, и как выполнить ваше желание описывается в разделе Демонизация документации Celery. Для наиболее распространённой установки Linux с использованием systemd вы можете воспользоваться примерами перечисленных ниже файлов, поставляемых в каталоге `examples`.

Модуль systemd, размещаемый по пути `/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
```

Файл настроек переменных окружения должен быть размещён в `/etc/default/celery-weblate`:

```
# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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 concurency 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"

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

Для ротации журналов Celery файл настроек **logrotate** должен быть размещён в `/etc/logrotate.d/celery`:

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

Периодические задачи при помощи Celery beat

Weblate поставляется со встроенной конфигурацией планировщика задач. Вы, однако, можете определить в `settings.py` дополнительные задачи, для примера смотрите раздел *Отложенные коммиты*.

Предполагается, что задачи будут выполняться демоном Celery beat. В случае, если он не работает должным образом, он может быть не запущен или его база данных была повреждена. В этом случае проверьте журналы запуска Celery, чтобы выяснить первопричину проблемы.

Мониторинг состояния Celery

Вы можете использовать команду `celery_queues` для просмотра текущей длины очереди задач Celery. В случае, если очередь станет слишком длинной, вы также увидите ошибку конфигурации в интерфейсе администратора.

> **Предупреждение:** Ошибки Celery по умолчанию записываются только в журнал Celery и пользователю не видны. В случае, если вы хотите видеть сводку по таким ошибкам, рекомендуется настроить *сбор отчётов об ошибках*.

См.также:

Конфигурация и настройки по умолчанию, Руководство по обработчикам, Демонизация, Руководство по мониторингу и управлению, *celery_queues*

2.1.12 Мониторинг Weblate

Weblate предоставляет URL-адрес `/healthz/`, который можно использоваться для простой проверки его работоспособности, например, при использовании Kubernetes.

2.1.13 Сбор отчётов об ошибках

Weblate, как и любое другое программное обеспечение, может дать сбой. Для сбора полезной информации о сбоях мы рекомендуем использовать услуги сторонних сервисов. Особенно они полезны в случае сбоя в задачах Celery, которые в противном случае просто записали бы ошибку в журнал, и вы не получили о ней никакого уведомления. Weblate поддерживает следующие сервисы:

Sentry

В Weblate встроена поддержка Sentry. Для его использования достаточно установить параметр *SENTRY_DSN* в `settings.py`:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

Weblate имеет встроенную поддержку Rollbar. Для его использования достаточно следовать инструкциям из документа`Уведомления Rollbar для Python'а <https://docs.rollbar.com/docs/python/>`_.

Вкратце, вам необходимо дописать в `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",
}
```

Всё остальное интегрируется автоматически, теперь у вас будут собираться ошибки как со стороны сервера, так и со стороны клиента.

2.1.14 Перенос Weblate на другой сервер

Перенос Weblate на другой сервер должен быть достаточно простой операцией, однако он хранит данные в нескольких местах, переносить которые следует с некоторой осторожностью. Лучше всего для переноса на время остановить Weblate.

Перенос базы данных

В зависимости от серверной части вашей базы данных, у вас может быть несколько вариантов переноса базы данных. Самый простой из них — это сделать дамп базы на старом сервере и импортировать его на новый. В качестве альтернативы вы можете использовать репликацию, если ваша база данных её поддерживает.

Наилучшим подходом будет использование родных инструментов для работы с базами данных, так как они обычно наиболее эффективны (например, **mysqldump** или **pg_dump**). Если вы выполнить перенос между разными базами данных, то единственным вариантом может оказаться использование команд управления Django для создания дампа и импорта базы данных:

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

Перенос репозиториев системы контроля версий

Также необходимо перенести репозитории системы контроля версий, хранящиеся в каталоге `DATA_DIR`. Вы можете просто скопировать их, либо воспользоваться командой **rsync** для более эффективного выполнения переноса.

Прочие заметки

Не забудьте переместить и другие сервисы, которые Weblate мог использовать, например, Redis, задачи Cron или модули пользовательской аутентификации.

2.2 Развёртывание Weblate

Weblate легко устанавливается в вашем облаке. Пожалуйста, изучите подробное руководство по вашей платформе:

- *Установка с помощью Docker'а*
- *Установка на OpenShift*
- *Установка на Kubernetes*

2.2.1 Сторонние методы развёртывания Weblate

Примечание: Команда и разработчики Weblate не имеют отношения к разработке и поддержке следующих методов развёртывания. Некоторые детали процесса установки могут отличаться от того, что описано в данном документе.

Стек Bitnami Weblate

Bitnami предоставляет стек Weblate для многих платформ по адресу <https://bitnami.com/stack/weblate>. Во время установки настройки будут скорректированы, дополнительную документацию смотрите в <https://bitnami.com/stack/weblate/README.txt>.

Пакет Cloudron Weblate

Cloudron — это платформа для собственного хостинга web-приложений. Weblate установленный через Cloudron будет автоматически обновляться. Этот пакет поддерживается командой Cloudron в их репозитории пакетов Weblate.

Cloudron install

Weblate в YunoHost

Проект самостоятельного хостинга YunoHost предоставляет пакет для Weblate. После установки YunoHost вы сможете установить Weblate как любое другое приложение. Он предоставит вам полностью рабочий стек с резервным копированием и восстановлением, но вам всё равно может понадобиться отредактировать файл настроек для определённых целей использования.

Вы можете воспользоваться интерфейсом администрирования или этой кнопкой (она приведёт вас на ваш сервер):

Install with YUNOHOST

Также можно использовать интерфейс командной строки:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Обновление Weblate

2.3.1 Обновления образа Docker'а

В официальный образ Docker'а (смотрите раздел *Установка с помощью Docker'а*) уже интегрированы все шаги для обновления. Вручную делать ничего не надо, за исключением извлечения последней версии.

2.3.2 Общие инструкции по обновлению

Перед обновлением, пожалуйста, проверьте текущие *зависимости программного обеспечения*, поскольку они могли измениться. После того, как все зависимости будут установлены или обновлены, пожалуйста, приведите свой файл настроек `settings.py` в соответствие с изменениями в конфигурации (правильные значения смотрите в `settings_example.py`).

Перед обновлением всегда проверяйте *инструкции для конкретных версий*. Если вы пропустили какие-то версии, пожалуйста, при обновлении проследуйте инструкциям для всех версий, которые вы пропустили. Иногда, чтобы обеспечить плавный переход, лучше сперва обновиться до какой-нибудь промежуточной версии. Обновление через несколько версий должно работать, но оно протестировано не так хорошо, как обновление на одну версию.

Примечание: Перед обновлением рекомендуется выполнить полное резервное копирование базы данных, чтобы её можно было откатить в случае неудачи обновления, смотрите раздел *Резервное копирование и перенос Weblate*.

1. Остановите процессы wsgi и Celery. Обновление может выполнять несовместимые изменения в базе данных, поэтому во время обновления всегда безопаснее избегать выполнения старых процессов.
2. Обновите код Weblate.

   При установке через pip это можно сделать следующим образом:

   ```
   pip install -U Weblate
   ```

   При использовании кода из Git'а вам нужно будет извлечь новый исходный код и обновить вашу установку:

   ```
   cd weblate-src
   git pull
   # Update Weblate inside your virtualenv
   . ~/weblate-env/bin/pip install -e .
   # Install dependencies directly when not using virtualenv
   pip install --upgrade -r requirements.txt
   ```

3. Обновите файл настроек, пример смотрите в `settings_example.py` или обратитесь к разделу *Инструкции для конкретных версий* для получения информации о необходимых шагах.
4. Обновите структуру базы данных:

   ```
   weblate migrate --noinput
   ```

5. Соберите обновлённые статические файлы (смотрите разделы *Запуск сервера* и *Обслуживание статических файлов*):

   ```
   weblate collectstatic --noinput
   ```

6. Сожмите файлы JavaScript и CSS (необязательно, смотрите раздел *Сжатие клиентских ресурсов*):

   ```
   weblate compress
   ```

7. Если вы используете версию из Git'а, вам также при каждом обновлении следует заново сгенерировать файлы локалей. Вы можете сделать это, выполнив следующую команду:

   ```
   weblate compilemessages
   ```

8. Убедитесь, что ваша установка в порядке (также смотрите раздел *Рабочая среда*):

   ```
   weblate check --deploy
   ```

9. Перезапустите рабочий процесс celery (смотрите раздел *Фоновые задачи с использованием Celery*).

2.3.3 Инструкции для конкретных версий

Обновление с ветки 2.x

При обновлении с выпусков ветки 2.x всегда сначала выполняйте обновление до версии 3.0.1, а затем продолжайте обновление в ветке 3.x. Обновления, пропустившие этот этап, не поддерживаются и будут работать неправильно.

См.также:

Обновление с версии 2.20 до версии 3.0 в документации Weblate 3.0

Обновление с ветки 3.x

При обновлении с выпусков ветки 3.x всегда сначала выполняйте обновление до версии 4.0.4 или 4.1.1, а затем продолжайте обновление в ветке 4.x. Обновления, пропустившие этот этап, не поддерживаются и будут работать неправильно.

См.также:

Обновление с версии 3.11 до версии 4.0 в документации Weblate 4.0

Обновление с версии 4.0 до версии 4.1

Для выполнения обновления, пожалуйста, следуйте *общим инструкциям по обновлению*.

Заметные изменения конфигурации или зависимостей:

- В файл `settings_example.py` внесено несколько изменений, наиболее заметные из которых касаются промежуточного программного обеспечения, пожалуйста, соответствующим образом подправьте свой файл настроек.
- Появилась поддержка нескольких новых форматов файлов, возможно, вы захотите включить их обработку в случае, если вы изменяли значение параметра *`WEBLATE_FORMATS`*.
- Появились новые проверки качества, возможно, вы захотите включить их, если вы изменяли параметр *`CHECK_LIST`*.
- Параметр `DEFAULT_THROTTLE_CLASSES` был изменён, чтобы разрешать сообщать об ограничении частоты запросов к API.
- Появились новые зависимости и обновились старые.
- Произошло изменение параметра `INSTALLED_APPS`.
- Машинный перевод *DeepL* теперь по умолчанию использует API версии 2, вам может понадобиться подправить параметр *`MT_DEEPL_API_VERSION`* на тот случай, если ваша текущая подписка DeepL не поддерживает эту версию.

См.также:

Общие инструкции по обновлению

Обновление с версии 4.1 до версии 4.2

Для выполнения обновления, пожалуйста, следуйте *общим инструкциям по обновлению*.

Заметные изменения конфигурации или зависимостей:

- Обновление с выпусков ветки 3.x больше не поддерживается, пожалуйста, сначала обновитесь до версии 4.0 или 4.1.
- Появились новые зависимости и обновились старые.
- В файл `settings_example.py` внесено несколько изменений, наиболее заметные из которых касаются нового промежуточного программного обеспечения и изменения порядка приложений.
- Ключи для форматов на основе JSON больше не включают ведущую точку. Строки корректируются во время миграции базы данных, но может понадобиться откорректировать внешние компоненты, если вы полагались на значения ключей после экспорта или в API.
- Настройки Celery были изменены: очередь `memory` более не используется. Пожалуйста, подправьте свои скрипты запуска и значение параметра `CELERY_TASK_ROUTES`.
- Домен Weblate'а теперь настраивается через параметры, смотрите описание параметра *`SITE_DOMAIN`* (или переменной окружения *`WEBLATE_SITE_DOMAIN`*). Перед запуском Weblate их необходимо настроить.
- Имена пользователей и адреса электронной почты в базе данных теперь уникальны независимо от регистра. Ранее из-за ошибки и особенностей PostgreSQL имена и адреса с разным регистром символов считались одинаковыми.

См.также:

Общие инструкции по обновлению

Обновление с версии 4.2 до версии 4.3

Для выполнения обновления, пожалуйста, следуйте *общим инструкциям по обновлению*.

Заметные изменения конфигурации или зависимостей:

- В проверках качества произошли некоторые изменения, возможно, вы захотите включить их, если вы изменяли параметр *`CHECK_LIST`*.
- Атрибут «исходный язык» перемещён из проекта в компонент, что также затронуло API. Вам нужно будет обновить *клиента Weblate*, если вы используете оный.
- Миграция базы данных на версию 4.3 также может занять значительное количество времени (в зависимость от количества строк, которые вы переводите). Стоит ожидать, что миграция будет занимать примерно один час на каждые 100 000 исходных строк.
- Произошло изменение параметра `INSTALLED_APPS`.
- Появился новый параметр *`SESSION_COOKIE_AGE_AUTHENTICATED`*, который дополняет `SESSION_COOKIE_AGE`.
- Если вы пользовались командами **`hub`** или **`lab`** для интеграции с GitHub или GitLab, то вам нужно будет настроить их заново, см. *`GITHUB_CREDENTIALS`* и *`GITLAB_CREDENTIALS`*.

  Изменено в версии 4.3.1: The Celery configuration was changed to add `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.

  Изменено в версии 4.3.2: The `post_update` method of addons now takes extra `skip_push` parameter.

См.также:

Общие инструкции по обновлению

Обновление с версии 4.3 до версии 4.4

Для выполнения обновления, пожалуйста, следуйте *общим инструкциям по обновлению*.

Заметные изменения конфигурации или зависимостей:

- Произошло изменение параметра `INSTALLED_APPS`, к нему надо добавить `weblate.configuration`.
- Теперь требуется Django 3.1.
- Если вы используете MySQL или MariaDB, то требуемая минимальная версия увеличилась, см. *mysql*.

  Изменено в версии 4.4.1: *Одноязычный 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 recommeded to force loading of affected component using `loadpo`. Increased minimal required version of translate-toolkit to address several file format issues.

См.также:

Общие инструкции по обновлению

Обновление с версии 4.4 до версии 4.5

Для выполнения обновления, пожалуйста, следуйте *общим инструкциям по обновлению*.

Заметные изменения конфигурации или зависимостей:

- Если у вас есть большие глоссарии, миграция может занять значительное время.
- Глоссарии теперь хранятся как обычные компоненты.
- API глоссария удалён, для доступа к глоссариям используйте стандартный API перевода.
- Параметр `INSTALLED_APPS` измененён: к нему надо добавить `weblate.metrics`.

  Изменено в версии 4.5.1: There is a new dependency on the *pyahocorasick* module.

См.также:

Общие инструкции по обновлению

2.3.4 Обновление с Python'а 2 на Python 3

Weblate больше не поддерживает Python старше версии 3.5. В случае, если вы всё ещё работаете на более старой версии, пожалуйста, сначала выполните миграцию на Python 3 на существующей версии и обновите её только после перехода. Смотрите раздел Обновление с Python'а 2 на Python 3 в документации к Weblate 3.11.1.

2.3.5 Переход с других баз данных на PostgreSQL

Если вы используете Weblate базой данных, отличной от PostgreSQL, вам следует перейти на PostgreSQL, поскольку в версии 4.0 это будет единственная поддерживаемая база данных. Следующие шаги помогут вам осуществить перенос данных между базами данных. Пожалуйста, не забудьте перед переездом остановить веб-сервер и сервер Celery, в противном случае у вас могут появиться несогласованные данные.

Создание базы данных в PostgreSQL

Обычно рекомендуется запускать Weblate в отдельной базе данных и под отдельной учётной записью:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser -D -P weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -O weblate weblate
```

Переход с использованием JSON-дампов Django

Простейшим способом перехода является использование JSON-дампов Django. Они хорошо работают для небольших установок. На больших сайтах вы вместо него можете использовать утилиту pgloader, смотрите раздел *Переход на PostgreSQL с помощью pgloader*.

1. Добавьте PostgreSQL в файл `settings.py` в качестве дополнительного подключения к базе данных:

```
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": "",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    },
}
```

2. Запустите миграции и удалите все данные, вставленные в таблицы:

```
weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --database=postgresql
```

3. Сделайте дамп прежней базы данных и импортируйте его в PostgreSQL

```
weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql
```

4. Подправьте параметр `DATABASES`, задав в нём использование по умолчанию только базы данных PostgreSQL и удалив старое соединение.

Теперь Weblate должен быть готов к запуску из базы данных PostgreSQL.

Переход на PostgreSQL с помощью pgloader

pgloader — это универсальный инструмент миграции для переноса данных на PostgreSQL. Вы можете использовать его для переноса базы данных Weblate.

1. Подправьте свой файл `settings.py`, задав в нём использование базы данных PostgreSQL.
2. Перенесите схему в базу данных PostgreSQL:

```
weblate migrate
weblate sqlflush | weblate dbshell
```

3. Запустите pgloader для переноса данных. Для переноса базы данных можно использовать следующий скрипт, но вы, возможно, захотите побольше узнать о pgloader'е, чтобы понять, что он делает, и подстроить его для соответствия вашим настройкам:

```
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 Переход с Pootle

Поскольку изначально Weblate был написан как замена для Pootle, он поддерживает перенос учётных записей пользователей из Pootle. Вы можете создать дамп пользователей Pootle, а затем импортировать его, используя команду Django `importusers`.

2.4 Резервное копирование и перенос Weblate

2.4.1 Автоматическое резервное копирование с помощью BorgBackup

Добавлено в версии 3.9.

В Weblate встроена поддержка создания сервисных резервных копий с помощью BorgBackup. Borg создаёт компактные зашифрованные резервные копии, которые можно безопасно хранить в облаке. Управлять резервными копиями можно через интерфейс управления на вкладке *Резервные копии*.

Изменено в версии 4.4.1: Теперь обе поддерживаемые базы данных: и PostgreSQL, и MySQL/MariaDB, включаются в автоматические резервные копии.

Резервные копии, которые делает Borg, являются инкрементальными, и Weblate настроен на хранение следующих ревизий резервных копий:

- Ежедневных, за последние 14 дней
- Еженедельных, за последние 8 недель
- Ежемесячных, за последние 6 месяцев

Weblate Dashboard Projects Languages Checks

Manage / Backups

Backup process triggered

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

Backup service: /tmp/tmpo07jlq8uweblate

Backup service credentials — March 26, 2021

Backup repository /tmp/tmpo07jlq8uweblate

Passphrase DVZ(Zi7MqJ0^#XXhclkiKdL$481qU7GM&rw0QipH96JoY7di7z

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

SSH key Download private key

The private key is needed to access the remote backup repository.

Deleted the oldest backups — March 26, 2021

Backup performed — March 26, 2021

Repository initialization — March 26, 2021

Turn off | Perform backup | Delete

Activate support package

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate | Purchase support package

Add backup service

Backup repository URL

Use /path/to/repo for local backups or user@host:/path/to/repo for remote SSH backups.

Add

Powered by Weblate 4.5.2 About Weblate Legal Contact Documentation Donate to Weblate

Ключ шифрования Borg

BorgBackup создаёт зашифрованные резервные копии и восстановить из без кодовой фразы у вас не получится. Парольная фраза генерируется при добавлении новой службы резервного копирования. Вы должны скопировать её и сохранить в безопасном месте.

Если же вы используете *хранилище резервных копий на Weblate.org*, то сохраните также резервную копию вашего закрытого SSH-ключа — он понадобится для доступа к вашим резервным копиям.

См.также:

borg init

2.4.2 Предоставляемое Weblate'ом хранилище резервных копий

Самый простой способ резервного копирования вашего экземпляра Weblate — приобрести сервис резервного копирования на weblate.org. Его можно будет активировать следующим образом:

1. Покупка *сервиса резервного копирования* на https://weblate.org/support/#backup.
2. Ввод полученного ключа в интерфейсе управления, смотрите раздел *Интегрированная поддержка*.
3. Weblate подключается к облачному сервису и получает информацию для доступа к резервным копиям.
4. Включение новой конфигурации резервного копирования на вкладке *Резервные копии*.
5. Создайте резервную копию своих учётных данных Borg, чтобы иметь возможность восстанавливать резервные копии; смотрите раздел «*Ключ шифрования Borg*».

Подсказка: Шаг ручного включения предусмотрен для вашей безопасности. Без вашего согласия никакие данные в хранилище резервных копий, полученное в процессе регистрации, не отправляются.

2.4.3 Использование собственного хранилища резервных копий

Также вы можете использовать своё собственное хранилище для резервного копирования. Загрузка резервных копий на удаленный сервер может осуществляться через SSH, на целевом сервере должен быть установлен BorgBackup.

См.также:

Раздел Общие сведения в документации Borg

Локальная файловая система

Для локальной резервной копии рекомендуется указывать абсолютный путь, например */путь/к/резервной_копии*. Каталог должен быть доступен на запись для пользователя, под которым запущен Weblate (смотрите раздел *Права доступа к файлам*). В случае, если каталог не существует, Weblate попытается его создать, но для этого понадобятся соответствующие права доступа.

Подсказка: Если Weblate запущен в Docker'е, убедитесь, что местоположение резервных копий в контейнере Weblate отображается на какой-либо том. В противном случае при перезагрузке контейнера Docker все резервные копии будут потеряны.

Одним из вариантов является размещение резервных копий в существующем томе. Подойдёт, к примеру, `/app/data/borgbackup`. Это один из томов уже существующих в контейнере.

Также вы можете добавить в файл Docker Compose новый контейнер для резервных копий и использовать, например, путь `/borgbackup`:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

Владельцем каталога, в котором будут хранится резервные копии, должен быть пользователь с UID 1000, Иначе Weblate не сможет сохранять в него оные.

Удалённые резервные копии

Для создания удалённых резервных копий вам также понадобится установить BorgBackup и на второй сервер, который будет доступен по SSH. Удостоверьтесь также в том, что он принимает клиентский SSH-ключ Weblate, т.е. тот, который он использует для доступа к другим серверам.

Подсказка: *Предоставляемое Weblate'ом хранилище резервных копий* предоставляет вам автоматическое удалённое резервное копирование.

См.также:

SSH-ключ Weblate

2.4.4 Восстановление из резервной копии Borg

1. Восстановите доступ к своему репозиторию резервных копий и подготовьте свою парольную фразу для резервного копирования.
2. Получите список всех резервных копий на сервере с помощью команды `borg list РЕПОЗИТОРИЙ`.
3. Восстановите нужную резервную копию в текущий каталог с помощью команды `borg extract РЕПОЗИТОРИЙ::АРХИВ`.
4. Восстановите базу данных из SQL-дампа, размещённого в каталоге `backup` каталога данных Weblate (смотрите раздел *Дампы данных для резервного копирования*).
5. Скопируйте файл настроек Weblate (`backups/settings.py`, смотрите раздел *Дампы данных для резервного копирования*) в нужное место, смотрите раздел *Изменение конфигурации под свои нужды*.
6. Скопируйте весь восстановленный каталог данных в место, заданное параметром `DATA_DIR`.

Сеанс Borg может выглядеть как-то так:

```
$ borg list /tmp/xxx
Enter passphrase for key /tmp/xxx:
2019-09-26T14:56:08                  Thu, 2019-09-26 14:56:08␣
↪[de0e0f13643635d5090e9896bdaceb92a023050749ad3f3350e788f1a65576a5]
$ borg extract /tmp/xxx::2019-09-26T14:56:08
Enter passphrase for key /tmp/xxx:
```

См.также:

borg list, borg extract

2.4.5 Ручное резервное копирование

В зависимости от того, что именно вы хотите сохранить, создайте резервную копию тех типов данных, которые Weblate сохраняет в каждом соответствующем месте.

Подсказка: Если вы выполняете резервное копирование вручную, вы можете подавить предупреждение Weblate об отсутствии резервного копирования, добавив проверку `weblate.I028` к параметру `SILENCED_SYSTEM_CHECKS` в `settings.py` или к переменной окружения *`WEBLATE_SILENCED_SYSTEM_CHECKS`* Docker'a.

```
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
```

База данных

Фактическое расположение хранилища зависит от настроек вашей базы данных.

Подсказка: База данных является самым важным хранилищем. Настройте регулярное резервное копирование своей БД. Без неё все переводы попросту пропадут.

Встроенные средства базы данных для резервного копирования

Рекомендуемый подход заключается в создании дампа базы данных с помощью родных для конкретной СУБД инструментов, таких как **`pg_dump`** или **`mysqldump`**. Обычно они работают лучше, чем резервные копии Django, и восстанавливают полные таблицы со всеми данными.

Эту резервную копию вы можете восстановить и на болле новом выпуске Weblate, при запуске через `migrate` он выполнит все необходимые миграции. За более подробной информацией о том, как выполнять обновление между версиями обратитесь к разделу «*Обновление Weblate*».

Резервное копирование базы данных средствами Django

В качестве альтернативы вы можете создать резервную копию базы данных при помощи Django'вской команды `dumpdata`. Резервное копирование этим способом не зависит от базы данных и может быть использовано в случае, если вы захотите сменить СУБД.

Перед восстановлением вам следует убедиться, что у вас запущена в точности та же самая версия Weblate, с которой была сделана резервная копия. Это необходимо, так как структура базы данных меняется от выпуска к выпуску, и если вы не выполните это условие, то, в конечном итоге, повредите данные тем или иным образом. После установке этой версии запустите все миграции базы данных с помощью команды `migrate`.

После этого некоторые записи уже будут созданы в базе данных, и они также попадут в резервную копию базы данных. Такие записи рекомендуется удалять вручную в оболочке управления (смотрите раздел *Вызов команд управления*):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Файлы

Если у вас достаточно места для резервного копирования, просто создайте резервную копию всего каталога *DATA_DIR*. Это беспроигрышный вариант, даже если он включает в себя некоторые файлы, которые вам не нужны. В следующих разделах подробно описано, для каких файлов нужно создавать резервные копии, а какие можно и пропустить.

Дампы данных для резервного копирования

Хранятся в *DATA_DIR* `/backups`.

Weblate сбрасывает сюда различные данные, и для создания более полных резервных копий вы можете включить эти файлы. Файлы обновляются ежедневно (требуется отдельный запущенный сервер Celery beat, смотрите раздел *Фоновые задачи с использованием Celery*). В настоящее время здесь хранятся:

- Настройки Weblate в виде файла `settings.py` (также здесь под именем `settings-expanded.py` лежит обработанная версия этого файла).
- Резервная копия базы данных PostgreSQL в виде файла `database.sql`.

Резервные копии базы данных по умолчанию сохраняются в виде простого текста, но их также можно сжать или полностью выключить, устанавив параметр *DATABASE_BACKUP*.

Репозитории системы контроля версий

Хранятся в *DATA_DIR* `/vcs`.

Репозитории системы контроля версий содержат копии ваших вышестоящих репозиториев с изменениями из Weblate'a. Если вы включили для всех ваших компонентов перевода отправку изменений в вышестоящий репозиторий при коммите (*Отправлять при коммите*), в этот репозиторий включаются все изменения из Weblate'a. Так что на стороне Weblate'a вам уже не нужно создавать резервные копии репозиториев. Они могут быть повторно склонированы из вышестоящих репозиториев без потери данных.

Ключи SSH и GPG

Хранятся в *DATA_DIR* `/ssh` и *DATA_DIR* `/home`.

Если вы используете сгенерированные Weblate'ом ключи SSH или GPG, вы должны создать резервные копии этих каталогов. Иначе вы потеряете закрытые ключи и вам придётся заново сгенерировать новые.

Загруженные пользователем файлы

Хранятся в *DATA_DIR* `/media`.

Вы должны сделать резервные копии всех загруженных пользователями файлов (например, *снимков экрана*).

Задачи Celery

Очередь задач Celery может содержать некоторую информацию, но обычно её не требуется сохранять в резервной копии. В худшем случае вы потеряете обновления, которые ещё не были занесены в память переводов. Поскольку при восстановлении в любом случае рекомендуется выполнять обновления полнотекстового индекса или репозитория, нет ничего страшного в том, что они потеряются.

См.также:

Фоновые задачи с использованием Celery

Командная строка для ручного резервного копирования

Используя задание в cron'е, вы можете задать команду bash'а, которая будет выполняться каждый день, например:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz␣
↪backups vcs ssh home media fonts secret
```

Строка между кавычками после *XZ_OPT* позволяет вам выбрать параметры для запуска xz, например, объём памяти, используемый для сжатия; смотрите руководство на странице https://linux.die.net/man/1/xz

Вы можете настроить список папок и файлов в соответствии со своими потребностями. Например, чтобы избежать сохранения памяти переводов (в папке резервных копий), вы можете использовать следующую команду:

```
$ 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 Восстановление ручных резервных копий

1. Восстановите все данные, сохранённые вами в резервной копии.
2. Обновите все репозитории с помощью команды `updategit`.

   ```
   weblate updategit --all
   ```

2.4.7 Перенос установки Weblate

Переместите свою установку на другую систему, следуя приведённым выше инструкциям по резервному копированию и восстановлению из резервной копии.

См.также:

Обновление с Python'а 2 на Python 3, *Переход с других баз данных на PostgreSQL*

2.5 Аутентификация

2.5.1 Регистрация пользователя

В установке по умолчанию Weblate использует python-social-auth, форму на сайте для обработки регистраций новых пользователей. После подтверждения своей электронной почты новый пользователь может вносить свой вклад в переводы или пройти аутентификацию, воспользовавшись одним из сторонних сервисов.

Также вы можете отключить регистрацию новых пользователей с помощью установки параметра `REGISTRATION_OPEN`.

Попытки аутентификации учитываются при применении *ограничений на частоту запросов*.

2.5.2 Способы аутентификации

Для аутентификации используется встроенное решение Django, включающее в себя для этого аутентификацию от различных социальных сетей. Использование этого решения означает, что вы сможете импортировать себе базу данных пользователей других проектов, написанных на Django (смотрите раздел *Переход с Pootle*).

Кроме того, аутентификация в Django может быть настроена и с использованием других способов.

См.также:

В разделе *Параметры аутентификации* описано, как настроить аутентификацию в официальном образе Docker'a.

2.5.3 Социальная аутентификация

Благодаря пакету Python Social Auth, Weblate поддерживает аутентификацию с использованием множества сторонних сервисов, таких как GitLab, Ubuntu, Fedora и другие.

Для получения общих инструкций по настройке ознакомьтесь с их документацией в разделе Фреймворк Django.

Примечание: По умолчанию Weblate полагается на сторонние сервисы аутентификации для предоставления подтверждённого адреса электронной почты. Если некоторые сервисы, которые вы хотите использовать, это не поддерживают, включите принудительную проверку адреса электронной почты на стороне Weblate, настроив их параметр FORCE_EMAIL_VALIDATION. Например:

```
SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True
```

См.также:

Конвейер

Включить конкретный способ аутентификации довольно просто: нужно всего-навсего добавить его имя в параметр `AUTHENTICATION_BACKENDS`, а также, по необходимости, задать необходимые ему ключи доступа. Обратите внимание, что некоторые способы по умолчанию не предоставляют адрес электронной почты пользователя, в этом случае вы должны запросить его явно, иначе Weblate не сможет правильно указать авторство выполненных пользователями переводов.

См.также:

Бэкэнд пакета Python Social Auth

Аутентификация OpenID

Для сервисов, основанных на OpenID, обычно это просто вопрос её включения. Следующий блок кода включает аутентификацию OpenID для OpenSUSE, Fedora и 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",
)
```

См.также:

OpenID

Аутентификация GitHub

Вам нужно будет зарегистрировать приложение OAuth в GitHub'е, а затем передать Weblate'у все его секреты:

```
# 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"]
```

На GitHub нужно задать URL-адрес обратного вызова, указывающий на `https://example.com/accounts/complete/github/`.

Примечание: Weblate предоставляет URL-адрес обратного вызова, вызываемый во время аутентификации, в который включён настроенный домен. В случае, если вы получаете ошибки о несовпадении URL, возможно, вы захотите это исправить, смотрите раздел *Установка правильного домена сайта*.

См.также:

GitHub

Аутентификация Bitbucket

Вам нужно будет зарегистрировать приложение в Bitbucket'е, а затем рассказать Weblate'у все его секреты:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.bitbucket.BitbucketOAuth",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_BITBUCKET_KEY = "Bitbucket Client ID"
SOCIAL_AUTH_BITBUCKET_SECRET = "Bitbucket Client Secret"
SOCIAL_AUTH_BITBUCKET_VERIFIED_EMAILS_ONLY = True
```

Примечание: Weblate предоставляет URL-адрес обратного вызова, вызываемый во время аутентификации, в который включён настроенный домен. В случае, если вы получаете ошибки о несовпадении URL, возможно, вы захотите это исправить, смотрите раздел *Установка правильного домена сайта*.

См.также:

Bitbucket

Google OAuth 2

Для использования Google OAuth 2 необходимо зарегистрировать приложение на <https://console.developers.google.com/> и включить Google+ API.

URL-адрес перенаправления — `https://СЕРВЕР WEBLATE/accounts/complete/google-oauth2/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"
```

Примечание: Weblate предоставляет URL-адрес обратного вызова, вызываемый во время аутентификации, в который включён настроенный домен. В случае, если вы получаете ошибки о несовпадении URL, возможно, вы захотите это исправить, смотрите раздел *Установка правильного домена сайта*.

См.также:

Google

Facebook OAuth 2

Как обычно с сервисами OAuth 2, вам необходимо зарегистрировать ваше приложение в Facebook. После этого вы сможете настроить Weblate на его использование:

URL-адрес перенаправления — `https://СЕРВЕР WEBLATE/accounts/complete/facebook/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
```

Примечание: Weblate предоставляет URL-адрес обратного вызова, вызываемый во время аутентификации, в который включён настроенный домен. В случае, если вы получаете ошибки о несовпадении URL, возможно, вы захотите это исправить, смотрите раздел *Установка правильного домена сайта*.

См.также:

Facebook

GitLab OAuth 2

Для использования GitLab OAuth 2 необходимо зарегистрировать приложение на <https://gitlab.com/profile/applications>.

URL-адрес перенаправления — `https://СЕРВЕР WEBLATE/accounts/complete/gitlab/` и убедитесь, что вы отметили область видимости *read_user*.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.gitlab.GitLabOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITLAB_KEY = "Application ID"
SOCIAL_AUTH_GITLAB_SECRET = "Secret"
SOCIAL_AUTH_GITLAB_SCOPE = ["read_user"]

# If you are using your own GitLab
# SOCIAL_AUTH_GITLAB_API_URL = 'https://gitlab.example.com/'
```

Примечание: Weblate предоставляет URL-адрес обратного вызова, вызываемый во время аутентификации, в который включён настроенный домен. В случае, если вы получаете ошибки о несовпадении URL, возможно, вы захотите это исправить, смотрите раздел *Установка правильного домена сайта*.

См.также:

GitLab

Microsoft Azure Active Directory

Weblate можно настроить на использование общих или конкретных арендаторов для аутентификации.

URL-адрес перенаправления — `https://СЕРВЕР WEBLATE/accounts/complete/azuread-oauth2/` для общей и `https://СЕРВЕР WEBLATE/accounts/complete/azuread-tenant-oauth2/` для конкретной для арендатора аутентификации.

```
# 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",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET = ""
# Tenant ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID = ""
```

Примечание: Weblate предоставляет URL-адрес обратного вызова, вызываемый во время аутентификации, в который включён настроенный домен. В случае, если вы получаете ошибки о несовпадении URL, возможно, вы захотите это исправить, смотрите раздел *Установка правильного домена сайта*.

См.также:

Microsoft Azure Active Directory

Slack

Для использования Slack OAuth 2 необходимо зарегистрировать приложение на <https://api.slack.com/apps>.

URL-адрес перенаправления — `https://СЕРВЕР WEBLATE/accounts/complete/slack/`.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.slack.SlackOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SLACK_KEY = ""
SOCIAL_AUTH_SLACK_SECRET = ""
```

Примечание: Weblate предоставляет URL-адрес обратного вызова, вызываемый во время аутентификации, в который включён настроенный домен. В случае, если вы получаете ошибки о несовпадении URL, возможно, вы захотите это исправить, смотрите раздел *Установка правильного домена сайта*.

См.также:

Slack

Отключение аутентификации по паролю

Аутентификацию по электронной почте и паролю можно отключить, удалив `social_core.backends.email.EmailAuth` из параметра `AUTHENTICATION_BACKENDS`. Никогда не удаляйте оттуда `weblate.accounts.auth.WeblateUserBackend`, он необходим для работы основных функций Weblate.

Совет: Вы всё ещё можете использовать аутентификацию по паролю для интерфейса администратора и для пользователей, которых вы создаёте там вручную. Просто зайдите на `/admin/`.

Например, аутентификация с использованием только провайдера openSUSE Open ID может быть достигнута при помощи следующего кода:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Аутентификация по паролю

По умолчанию `settings.py` поставляется с разумным набором проверок в параметре `AUTH_PASSWORD_VALIDATORS`:

- Пароли не могут быть слишком похожи на другие ваши личные данные.
- Пароли должны содержать минимум 10 символов.
- Пароли не могут быть широко распространёнными паролями.
- Пароли не могут быть полностью цифровыми.
- Пароли не могут состоять из одного символа или только из пробелов.
- Пароли не могут совпадать с предыдущими вашими паролями.

Вы можете настроить этот параметр в соответствии с вашей политикой паролей.

Дополнительно можно установить пакет django-zxcvbn-password, который вычисляет вполне реалистичные оценки сложности пароля и позволяет отклонять пароли с оценкой ниже определённого порога.

2.5.5 Аутентификация через SAML

Добавлено в версии 4.1.1.

Пожалуйста, следуйте инструкциям по настройке Python Social Auth. Значимые отличия:

- Weblate поддерживает один IDP, который в `SOCIAL_AUTH_SAML_ENABLED_IDPS` должен называться `weblate`.
- URL-адрес XML с метаданными SAML — `/accounts/metadata/saml/`.
- Следующие параметры заполняются автоматически: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT` и `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Пример настроек:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.saml.SAMLAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbnl+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
```

См.также:

Настройка SAML в Docker'е, SAML

2.5.6 Аутентификация через LDAP

Аутентификацию через LDAP лучше всего реализовать с помощью пакета *django-auth-ldap*. Установить его вы можете как обычно:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap
```

Предупреждение: При использовании django-auth-ldap версии старше 1.3.0, *автоматические прикрепления группы* для новых созданных пользователей будут работать неправильно.

Примечание: У Python'ьего модуля LDAP версии 3.1.0 есть некоторые несовместимости, которые могут помешать вам использовать эту версию. Если вы получили ошибку AttributeError: „module“ object has no attribute „_trace_level“, то может помочь понижение версии python-ldap до 3.0.0.

После установки пакета вы сможете подключить его к системе аутентификации Django:

```
# Add LDAP backed, keep Django one if you want to be able to sign in
# even without LDAP for admin account
AUTHENTICATION_BACKENDS = (
    "django_auth_ldap.backend.LDAPBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# LDAP server address
AUTH_LDAP_SERVER_URI = "ldaps://ldap.example.net"

# DN to use for authentication
AUTH_LDAP_USER_DN_TEMPLATE = "cn=%(user)s,o=Example"
# Depending on your LDAP server, you might use a different DN
# like:
# AUTH_LDAP_USER_DN_TEMPLATE = 'ou=users,dc=example,dc=com'

# List of attributes to import from LDAP upon sign in
# Weblate stores full name of the user in the full_name attribute
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "name",
    # Use the following if your LDAP server does not have full name
    # Weblate will merge them later
    # 'first_name': 'givenName',
    # 'last_name': 'sn',
    # Email is required for Weblate (used in VCS commits)
    "email": "mail",
}

# Hide the registration form
REGISTRATION_OPEN = False
```

Примечание: Вы должны удалить значение `'social_core.backends.email.EmailAuth'` из параметра `AUTHENTICATION_BACKENDS`, в противном случае пользователи смогут установить в Weblate

свой пароль и пройти аутентификацию, используя его. Оставление `'weblate.accounts.auth.WeblateUserBackend'` необходимо для того, чтобы работали разрешения и анонимные пользователи. Также он позволит вам войти в систему, используя учётную запись локального администратора, если вы его создали (например, с помощью команды `createadmin`).

Использование пароля привязки

Если вы не можете использовать для аутентификации прямую привязку, вам нужно будет использовать поиск, и предоставить для поиска привязываемого пользователя. Например:

```
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)"
)
```

Интеграция с 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
```

См.также:

Аутентификация в Django через LDAP, Аутентификация

2.5.7 Аутентификация через CAS

Аутентификация через CAS может быть выполнена с помощью такого пакета, как *django-cas-ng*.

Первый шаг — раскрытие поля электронной почты пользователя через CAS. Раскрытие нужно настроить на самом сервере CAS, и оно требует, чтобы у вас был запущен CAS как минимум версии v2, поскольку CAS версии v1 вообще не поддерживает атрибуты.

Второй шаг — обновление Weblate'а для использования вашего сервера CAS и атрибутов.

Для установки *django-cas-ng* выполните команду:

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

После установки пакета вы сможете подключить его к системе аутентификации Django, изменив файл `settings.py`:

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

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

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

Наконец, для связывания поля электронной почты с объектом пользователя может быть использован сигнал. Для этого необходимо импортировать сигнал из пакета *django-cas-ng* и подключить к этому сигналу свой код. Выполнение этого действия в файле настроек может вызвать проблемы, поэтому его код предлагается поместить:

- В метод `django.apps.AppConfig.ready()` конфигурации вашего приложения
- В проектовый файл `urls.py` (если у вас нет моделей)

```
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()
```

См.также:

Django CAS NG

2.5.8 Настройка сторонней аутентификации Django

Как правило, с Weblate должен заработать любой плагин аутентификации Django. Просто следуйте инструкциям для установки плагина и не забудьте оставить в способах аутентификации пользователя Weblate.

См.также:

Аутентификация через LDAP, *Аутентификация через CAS*

Обычно установка заключается в том, чтобы добавить способ аутентификации в список `AUTHENTICATION_BACKENDS` и установить соответствующее приложение аутентификации (если таковое имеется) и добавить его в список `INSTALLED_APPS`:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

2.6 Управление доступом

Weblate поставляется с тонко настраиваемой системой привилегий для назначения пользователям прав как в рамках всего экземпляра приложения, так и в рамках отдельных его частей.

Изменено в версии 3.0: Before Weblate 3.0, the privilege system was based on Django privilege system only, but is specifically built for Weblate now. If using anything older, please consult the documentation for the specific version you are using.

2.6.1 Simple access control

If you are not administrating the whole Weblate installation and just have access to manage certain projects (like on Hosted Weblate), your access control management options are limited to following settings. If you don't need any complex setup, those are sufficient for you.

Управление доступом к проекту

Примечание: Эта функция недоступна на тарифном плане для свободного ПО на облачный хостинг Weblate.

You can limit user's access to individual projects by selecting a different *Access control* setting. Available options are:

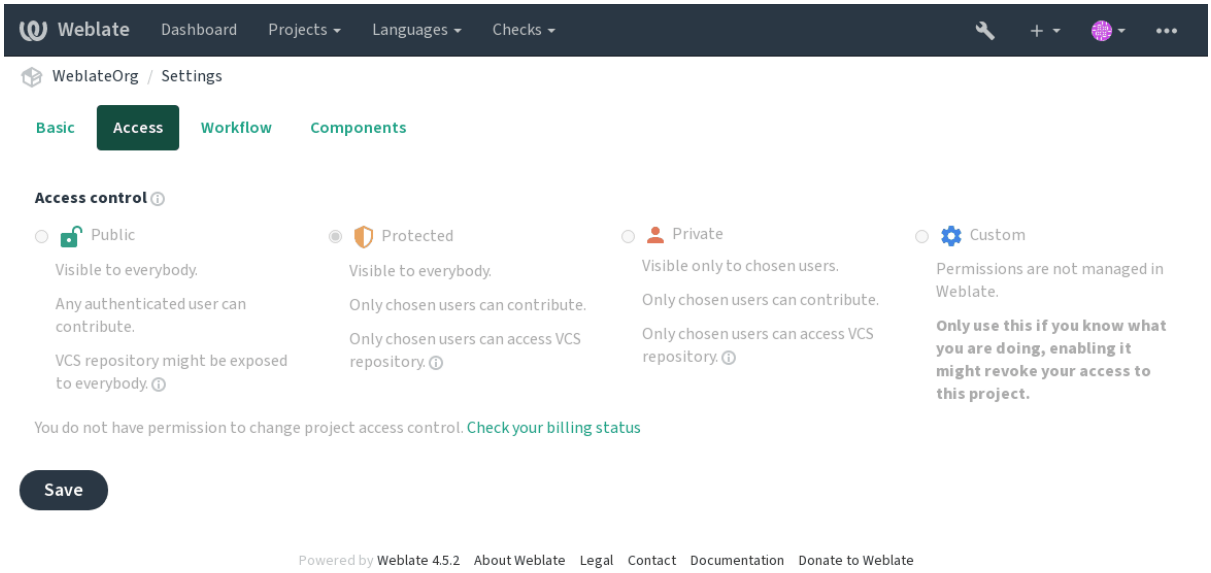
Публичный Publicly visible, translatable for all logged-in users.

Защищённый Publicly visible, but translatable only for selected users.

Частный Visible and translatable only for selected users.

Настраиваемый *User management* features will be disabled; by default all users are forbidden to performed any actions on the project. You will have to set up all the permissions using *Настраиваемый контроль доступа*.

Access control can be changed in the *Access* tab of the configuration (*Manage* ↓ *Settings*) of each respective project.



The default value can be changed by `DEFAULT_ACCESS_CONTROL`.

Примечание: Even for *Private* projects, some info about your project will be exposed: statistics and language summary for the whole instance will include counts for all projects despite the access control setting. Your project name and other information can't be revealed through this.

Примечание: The actual set of permissions available for users by default in *Public*, *Protected*, and *Private* projects can be redefined by Weblate instance administrator using *custom settings*.

Предупреждение: By turning on *Custom* access control, Weblate will remove all *special groups* it has created for a selected project. If you are doing this without admin permission for the whole Weblate instance, you will instantly lose your access to manage the project.

См.также:

Управление доступом

По-проектная настройка доступа

Users with the *Manage project access* privilege (see *Список привилегий*) can manage users in projects with non-*Custom* access control. They can assign users to one of the following groups.

For *Public*, *Protected* and *Private* projects:

Администрирование Includes all permissions available for the project.

Review (only if *review workflow* is turned on) Может одобрять переводы во время рецензирования.

For *Protected* and *Private* projects only:

Перевести Can translate the project and upload translations made offline.

Источники Can edit source strings (if allowed in the *project settings*) and source string info.

Языки Может управлять языками перевода (добавлять или удалять переводы).

Глоссарий Can manage glossary (add or remove entries, also upload).

Накопитель Может управлять памятью переводов.

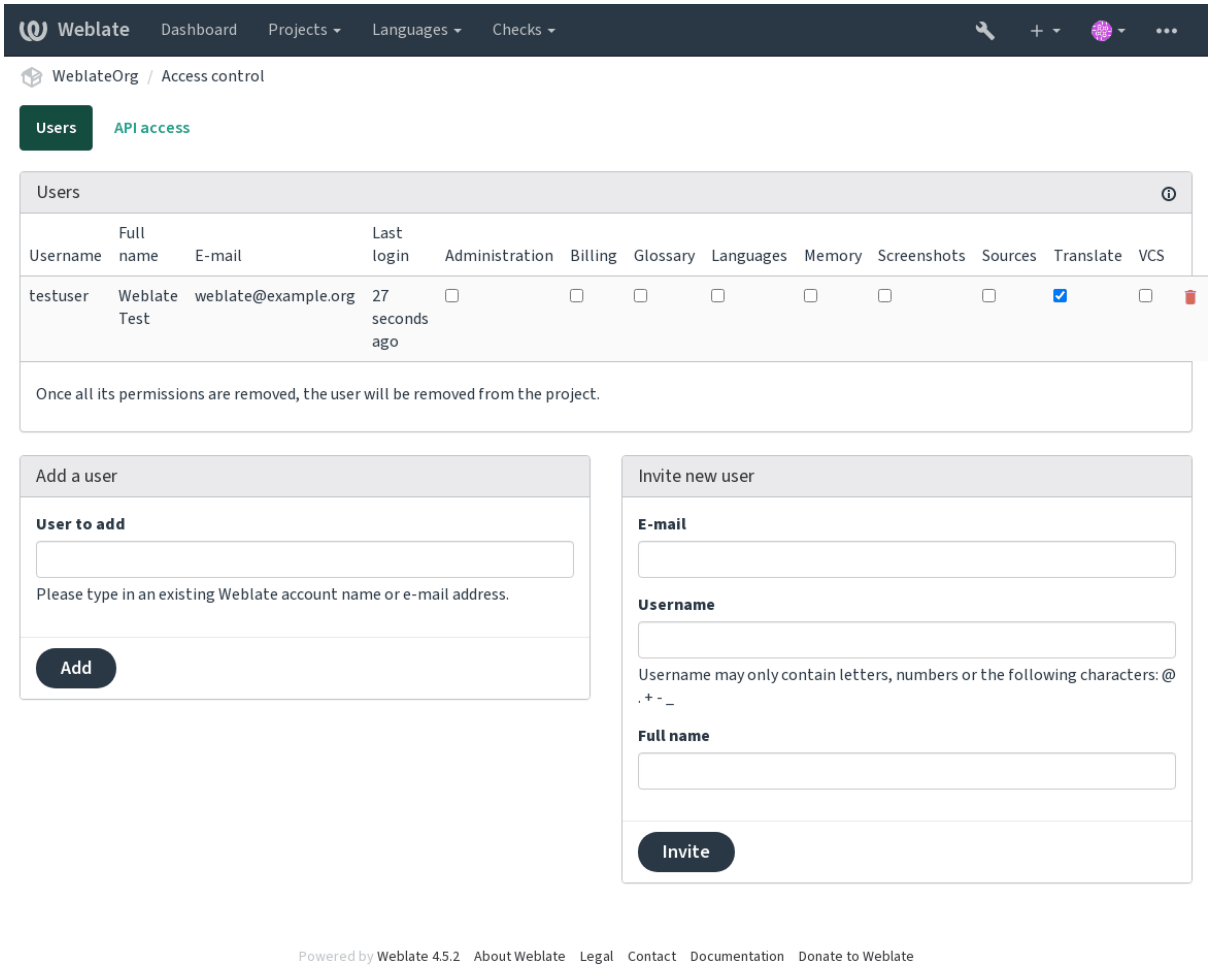
Снимки экрана Может управлять снимками экрана (добавлять или удалять их, а также связывать их с исходными строками).

СКВ (VCS) Может управлять системой контроля версий и имеет доступ к экспортированному репозиторию.

Тарифный план Can access billing info and settings (see *Тарифный план*).

Unfortunately, it's not possible to change this predefined set of groups for now. Also this way it's not possible to give just some additional permissions to all users.

Примечание: For non-*Custom* access control an instance of each group described above is actually defined for each project. The actual name of those groups will be `Project@Group`, also displayed in the Django admin interface this way. Although they can't be edited from Weblate user-interface.



These features are available on the *Access control* page, which can be accessed from the project's menu *Manage* ↓ *Users*.

New user invitation

Also, besides adding an existing user to the project, it is possible to invite new ones. Any new user will be created immediately, but the account will remain inactive until signing in with a link in the invitation sent via an e-mail. It is not required to have any site-wide privileges in order to do so, access management permission on the project's scope (e.g. a membership in the *Administration* group) would be sufficient.

Подсказка: If the invited user missed the validity of the invitation, they can set their password using invited e-mail address in the password reset form as the account is created already.

Добавлено в версии 3.11: It is possible to resend the e-mail for user invitations (invalidating any previously sent invitation).

The same kind of invitations are available site-wide from the *management interface* on the *Users* tab.

Per-project permission management

You can set your projects to *Protected* or *Private*, and *manage users* per-project in the Weblate user interface.

By default this prevents Weblate from granting access provided by *Users* and *Viewers* *default groups* due to these groups' own configuration. This doesn't prevent you from granting permissions to those projects site-wide by altering default groups, creating a new one, or creating additional custom settings for individual component as described in *Настраиваемый контроль доступа* below.

One of the main benefits of managing permissions through the Weblate user interface is that you can delegate it to other users without giving them the superuser privilege. In order to do so, add them to the *Administration* group of the project.

2.6.2 Настраиваемый контроль доступа

Примечание: Эта функция недоступна на тарифном плане для свободного ПО на облачный хостинг Weblate.

The permission system is based on groups and roles, where roles define a set of permissions, and groups link them to users and translations, see *Users, roles, groups, and permissions* for more details.

The most powerful features of the Weblate's access control system for now are available only through the *Django admin interface*. You can use it to manage permissions of any project. You don't necessarily have to switch it to *Custom access control* to utilize it. However you must have superuser privileges in order to use it.

If you are not interested in details of implementation, and just want to create a simple-enough configuration based on the defaults, or don't have a site-wide access to the whole Weblate installation (like on Hosted Weblate), please refer to the *Simple access control* section.

Распространённые варианты конфигурации

This section contains an overview of some common configurations you may be interested in.

Site-wide permission management

To manage permissions for a whole instance at once, add users to appropriate *default groups*:

- *Users* (this is done by default by the *automatic group assignment*).
- *Reviewers* (if you are using *review workflow* with dedicated reviewers).
- *Managers* (if you want to delegate most of the management operations to somebody else).

You should keep all projects configured as *Public* (see *Управление доступом к проекту*), otherwise the site-wide permissions provided by membership in the *Users* and *Reviewers* groups won't have any effect.

You may also grant some additional permissions of your choice to the default groups. For example, you may want to give a permission to manage screenshots to all the *Users*.

You can define some new custom groups as well. If you want to keep managing your permissions site-wide for these groups, choose an appropriate value for the *Project selection* (e.g. *All projects* or *All public projects*).

Особые разрешения для определённых языков, компонентов или проектов

You can create your own dedicated groups to manage permissions for distinct objects such as languages, components, and projects. Although these groups can only grant additional privileges, you can't revoke any permission granted by site-wide or per-project groups by adding another custom group.

Пример:

If you want (for whatever reason) to allow translation to a specific language (lets say *Czech*) only to a closed set of reliable translators while keeping translations to other languages public, you will have to:

1. Remove the permission to translate *Czech* from all the users. In the default configuration this can be done by altering the *Users* *default group*.

Таблица 1: Group *Users*

Выбор языка	*As defined*
Языки	All but *Czech*

2. Add a dedicated group for *Czech* translators.

Таблица 2: Group *Czech translators*

Роли	*Power users*
Выбор проекта	*All public projects*
Выбор языка	*As defined*
Языки	*Czech*

3. Add users you wish to give the permissions to into this group.

As you can see, permissions management this way is powerful, but can be quite a tedious job. You can't delegate it to another user, unless granting superuser permissions.

Users, roles, groups, and permissions

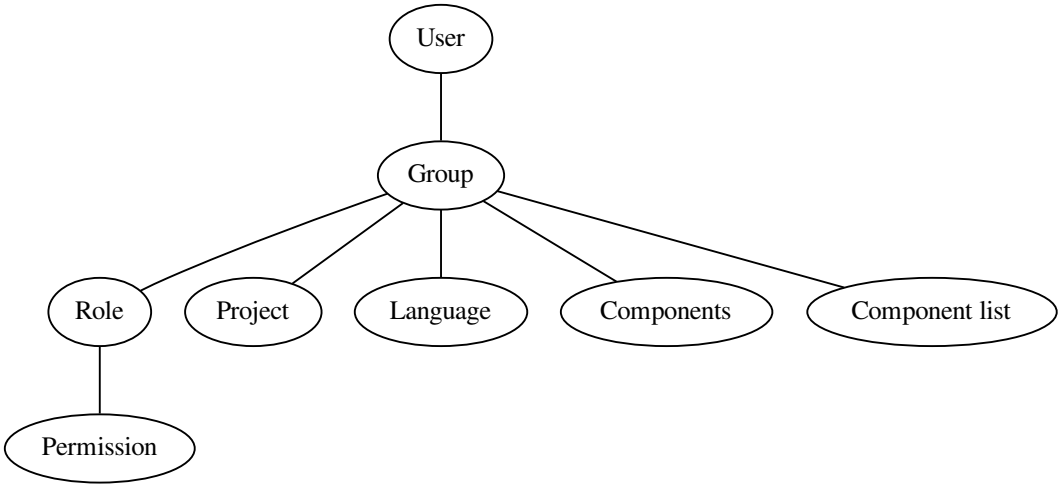
Модели аутентификации состоят из нескольких объектов:

Разрешение Individual permission defined by Weblate. Permissions cannot be assigned to users. This can only be done through assignment of roles.

Роль A role defines a set of permissions. This allows reuse of these sets in several places, making the administration easier.

Пользователь User can belong to several groups.

Группа Group connect roles, users, and authentication objects (projects, languages, and component lists).



Примечание: A group can have no roles assigned to it, in that case access to browse the project by anyone is assumed (see below).

Access for browse to a project

A user has to be a member of a group linked to the project, or any component inside that project. Having membership is enough, no specific permissions are needed to browse the project (this is used in the default *Viewers* group, see *Список групп*).

Access for browse to a component

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 *Ограниченный доступ* turned on, access to the component requires explicit permissions for the component (or a component list the component is in).

Scope of groups

The scope of the permission assigned by the roles in the groups are applied by the following rules:

- 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 *Selected projects* set to a value like *All public*, 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.

Подсказка: Use *Language selection* or *Project selection* to automate inclusion of all languages or projects.

Пример:

Let's say there is a project `foo` with the components: `foo/bar` and `foo/baz` and the following group:

Таблица 3: Group *Spanish Admin-Reviewers*

Роли	*Review Strings*, *Manage repository*
Компоненты	foo/bar
Языки	*Spanish*

Members of that group will have following permissions (assuming the default role settings):

- General (browsing) access to the whole project `foo` including both components in it: `foo/bar` and `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.

Автоматические прикрепления группы

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`.

Примечание: 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).

Примечание: 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*.

Группы и роли по умолчанию

After installation, a default set of groups is created (see *Список групп*).

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.

Список привилегий

Тарифный план (смотрите раздел *Тарифный план*) Просмотр платёжной информации [*Администрирование*, *Биллинг*]

Изменения Скачивание изменений [*Администрирование*]

Комментарии Публикация комментариев [*Администрирование*, *Правка источника*, *Опытный пользователь*, *Рецензирование строк*, *Перевод*]

Удаление комментариев [*Администрирование*]

Компонент Правка параметров компонентов [*Администрирование*]

Блокировка компонента, предотвращает его перевод [*Администрирование*]

Глоссарий Добавление записей в глоссарий [*Администрирование*, *Управление глоссарием*, *Опытный пользователь*]

Правка записей глоссария [*Администрирование*, *Управление глоссарием*, *Опытный пользователь*]

Удаление записей из глоссария [*Администрирование*, *Управление глоссарием*, *Опытный пользователь*]

Загрузка записей в глоссарий [*Администрирование*, *Управление глоссарием*, *Опытный пользователь*]

Автоматические предложения Использование автоматических предложений [*Администрирование*, *Правка источника*, *Опытный пользователь*, *Рецензирование строк*, *Перевод*]

Память переводов Правка памяти переводов [*Администрирование*, *Управление памятью переводов*]

Удаление памяти переводов [*Администрирование*, *Управление памятью переводов*]

Проекты Правка параметров проекта [*Администрирование*]

Управление доступом к проекту [*Администрирование*]

Отчёты Скачивание отчётов [*Администрирование*]

Снимки экрана Добавление снимков экрана [*Администрирование*, *Управление снимками экрана*]

Правка снимков экрана [*Администрирование*, *Управление снимками экрана*]

Удаление снимков экрана [*Администрирование*, *Управление снимками экрана*]

Исходные строки Правка дополнительных свойств строки [*Администрирование*, *Правка источника*]

Строки Добавление новой строки [*Администрирование*]

Удаление строки [*Администрирование*]

Игнорирование неудачных проверок [*Администрирование*, *Правка источника*, *Опытный пользователь*, *Рецензирование строк*, *Перевод*]

Правка строк [*Администрирование*, *Правка источника*, *Опытный пользователь*, *Рецензирование строк*, *Перевод*]

Рецензирование строк [*Администрирование*, *Рецензирование строк*]

Правка строк, когда предложения являются обязательными [*Администрирование*, *Рецензирование строк*]

Правка исходных строк [*Администрирование*, *Правка источника*, *Опытный пользователь*]

Предложения Принятие предложения [*Администрирование*, *Правка источника*, *Опытный пользователь*, *Рецензирование строк*, *Перевод*]

Добавление предложения [*Администрирование*, *Правка источника*, *Добавление предложения*, *Опытный пользователь*, *Рецензирование строк*, *Перевод*]

Удаление предложения [*Администрирование*, *Опытный пользователь*]

Голосование за предложения [*Администрирование*, *Правка источника*, *Опытный пользователь*, *Рецензирование строк*, *Перевод*]

Переводы Добавление языка для перевода [*Администрирование*, *Опытный пользователь*, *Управление языками*]

Производство автоматического перевода [*Администрирование*, *Управление языками*]

Удаление существующего перевода [*Администрирование*, *Управление языками*]

Добавление нескольких языков для перевода [*Администрирование*, *Управление языками*]

Загрузки Задание автора загружаемого перевода [*Администрирование*]

Перезапись существующих строк при загрузке новых [*Администрирование*, *Правка источника*, *Опытный пользователь*, *Рецензирование строк*, *Перевод*]

Загрузка переводов [*Администрирование*, *Правка источника*, *Опытный пользователь*, *Рецензирование строк*, *Перевод*]

СКВ (VCS) Доступ к внутреннему репозиторию [*Администрирование*, *Доступ к репозиторию*, *Опытный пользователь*, *Управление репозиторием*]

Коммит изменений во внутренний репозиторий [*Администрирование*, *Управление репозиторием*]

Отправка изменений из внутреннего репозитория [*Администрирование*, *Управление репозиторием*]

Сброс изменений во внутреннем репозитории [*Администрирование*, *Управление репозиторием*]

Просмотр расположения вышестоящего репозитория [*Администрирование*, *Доступ к репозиторию*, *Опытный пользователь*, *Управление репозиторием*]

Обновление внутреннего репозитория [*Администрирование*, *Управление репозиторием*]

Привилегии для всего сайта Использование интерфейса управления

Добавление новых проектов

Добавление определений языка

Управление определениями языков

Управление группами

Управление пользователями

Управление ролями

Управление объявлениями

Управление памятью переводов

Управление списками компонентов

Примечание: Привилегии для всего сайта не предоставляются никаким умолчательным ролям. Они мощны и достаточно близки к статусу суперпользователя. Большинство из них влияют на все проекты вашей установки Weblate.

Список групп

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.

Гости Определяет разрешения для неаутентифицированных пользователей.

Эта группа содержит только анонимных пользователей (смотрите параметр `ANONYMOUS_USER_NAME`).

Вы можете удалить роли из этой группы, чтобы ограничить разрешения для неаутентифицированных пользователей.

Роли по умолчанию: *Добавление предложений*, *Доступ к репозиторию*

Просматривающие This role ensures visibility of public projects for all users. By default, all users are members of this group.

By default, *automatic group assignment* makes all new accounts members of this group when they join.

Роли по умолчанию: нет

Пользователи Стандартная группа для всех пользователей.

By default, *automatic group assignment* makes all new accounts members of this group when they join.

Роли по умолчанию: *Опытный пользователь*

Рецензенты Группа для рецензентов (смотрите раздел *Рабочие процессы перевода*).

Роли по умолчанию: *Рецензирование строк*

Управляющие Группа для администраторов.

Роли по умолчанию: *Администрирование*

Предупреждение: 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 Additional access restrictions

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 logging-in to access all the site pages. This is basically the way to lock your Weblate installation.

Подсказка: You can use built-in *invitations* to add new users.

2.7 Проекты перевода

2.7.1 Организация переводов

Weblate организует переводимое содержимое проектов/компонентов из системы контроля версий в древовидную структуру.

- Объектом нижнего уровня является *проект*, который должен содержать все связанные переводы (например, перевод приложения различных версий и/или сопроводительную документацию).
- На уровне выше располагается *компонент*, который фактически и является компонентом для перевода, для которого вы определяете используемый репозиторий системы контроля версий и маску файлов для перевода.
- Ещё выше *компонента* находятся отдельные переводы, которые Weblate обрабатывает автоматически, как только файлы перевода (соответствующие *маске файлов*, определённой в *конфигурации компонента*), появляются в репозитории системы контроля версий.

Weblate поддерживает широкий набор форматов перевода (как двуязычных, так и одноязычных), поддерживаемых набором инструментов Translate Toolkit, смотрите раздел *Поддерживаемые форматы файлов*.

Примечание: Вы можете совместно использовать склонированные репозитории системы контроля версий, используя *внутренние URL-адреса Weblate*. Использование этой возможности настоятельно рекомендуется при наличии множества компонентов, совместно использующих один и тот же репозиторий. Она повышает производительность и уменьшает необходимое дисковое пространство.

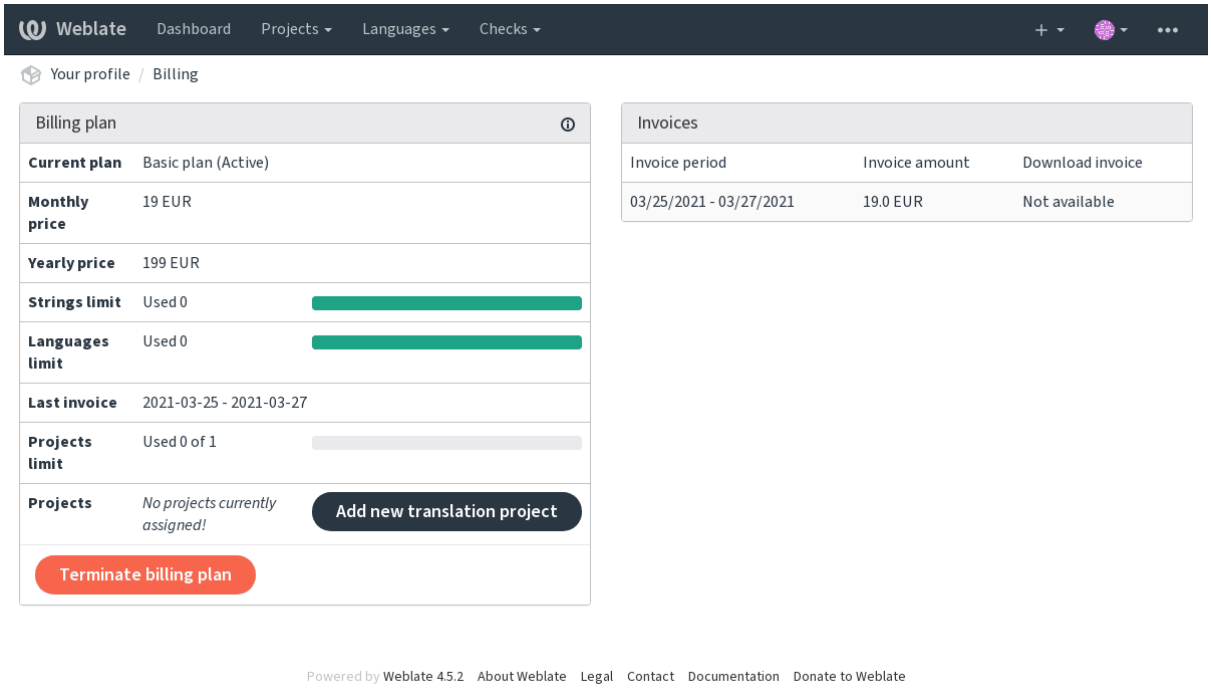
2.7.2 Добавление проектов и компонентов перевода

Изменено в версии 3.2: В Weblate'е появился интерфейс для добавления проектов и компонентов, так что вам больше не нужно использовать для этого *административный интерфейс Django*.

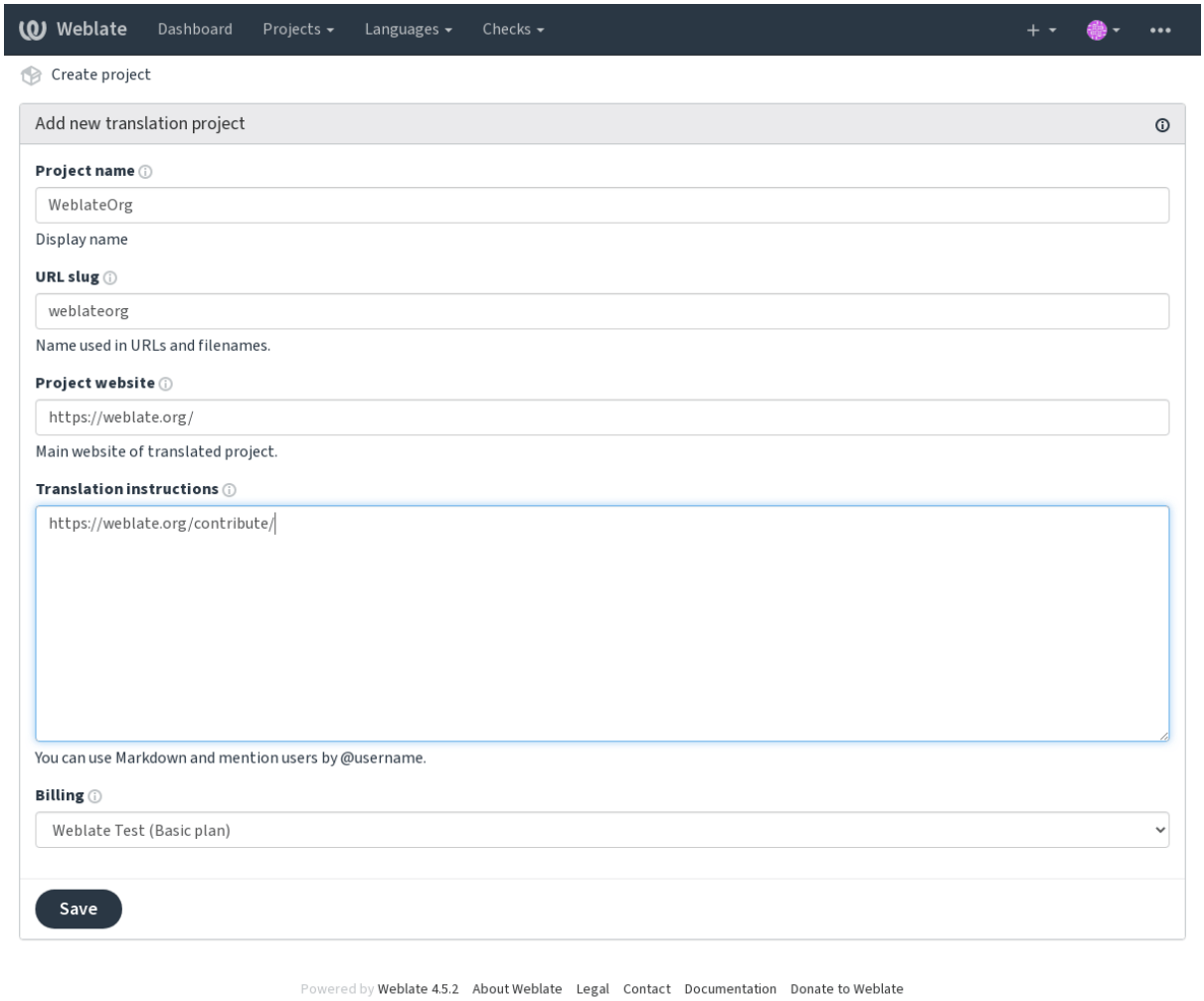
Изменено в версии 3.4: Теперь процесс добавления компонентов состоит из нескольких шагов, с автоматическим обнаружением большинства параметров.

В зависимости от ваших разрешений вы можете создавать новые проекты и компоненты перевода. Для пользователей с разрешением *Добавление новых проектов* их создание всегда разрешено, а если ваш экземпляр использует тарифные планы (как, например, https://hosted.weblate.org/, смотрите раздел *Тарифный план*), вы также можете создавать их на основе разрешений из вашего плана, установленных учётной записью пользователя, управляющего тарификацией.

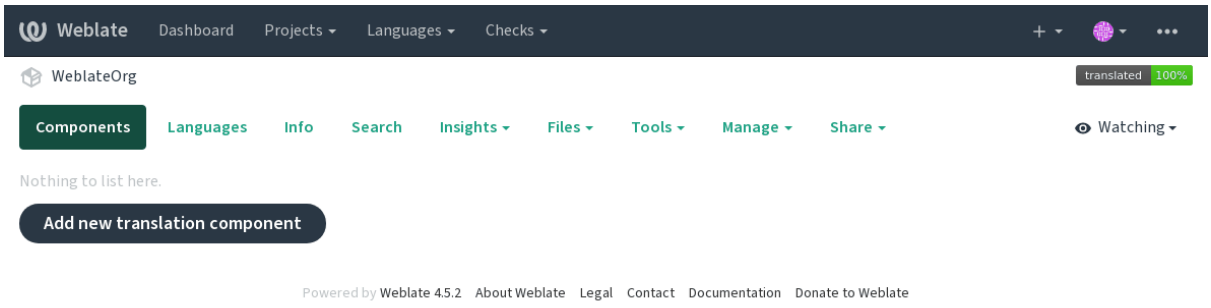
Вы можете просмотреть свой текущий тарифный план на отдельной странице:



Создание проекта можно инициировать оттуда или из меню на панели навигации. Чтобы завершить добавление проекта перевода, заполните основную информацию о нём:



После создания проекта вы попадаете непосредственно на страницу проекта:



Создание нового компонента перевода можно инициировать одним щелчком мыши. Процесс создания компонента многоступенчатый, большинство параметров перевода он определяет автоматически. Существует несколько подходов к созданию компонента:

Из системы контроля версий Создать компонент из удалённого репозитория системы контроля версий.

Из уже существующего компонента Создать дополнительный компонент на основе уже существующего, только с другими файлами.

Дополнительная ветка Создать дополнительный компонент на основе уже существующего, только для другой ветки.

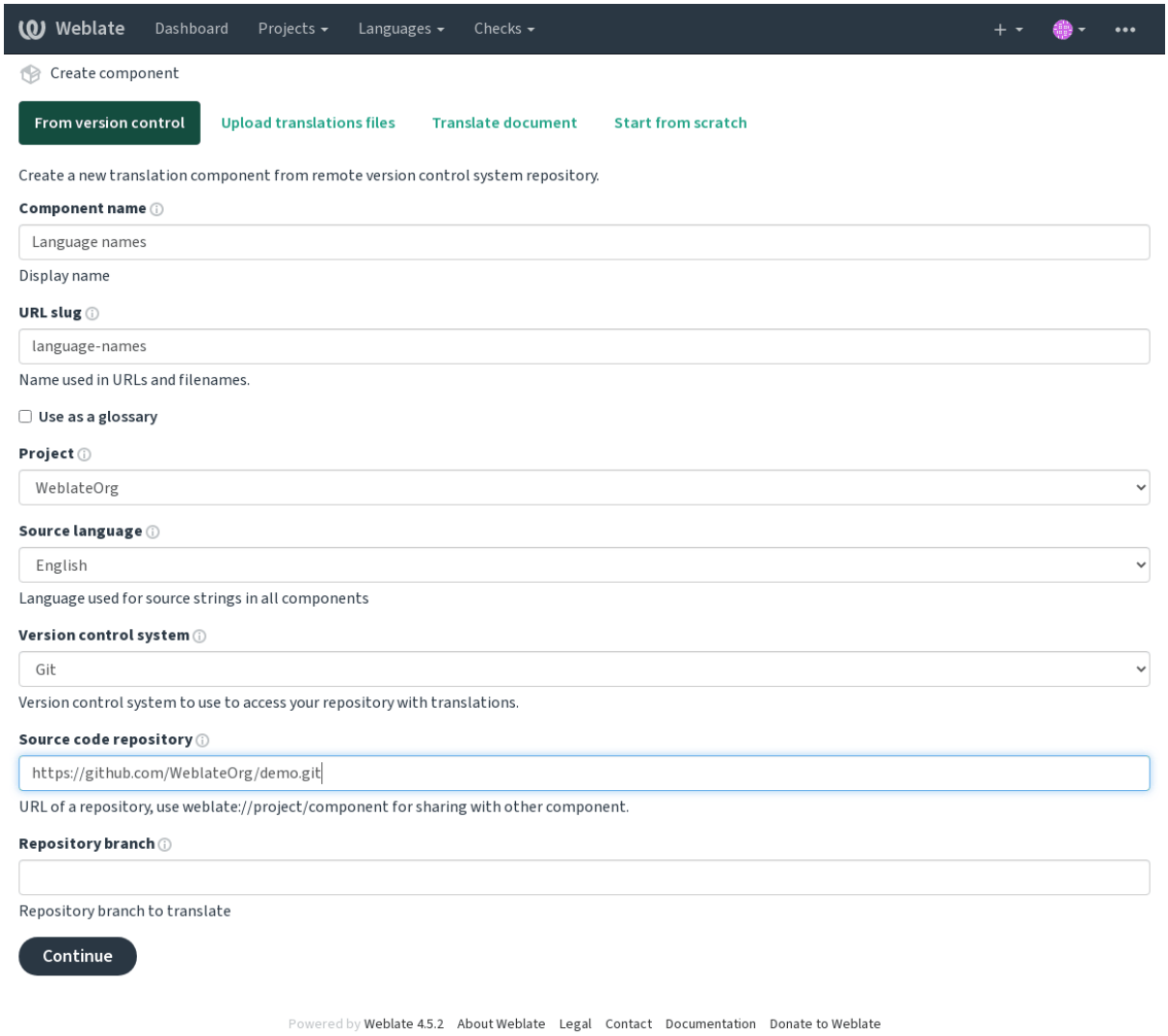
Загрузить файлы перевода Загрузить файлы перевода в Weblate в случае, если у вас нет системы контроля версий или вы не хотите интегрировать её с Weblate. Позже вы сможете обновить содержимое через веб-интерфейс или *REST API Weblate'a*.

Перевести документ Загрузить одиночный документ и перевести его.

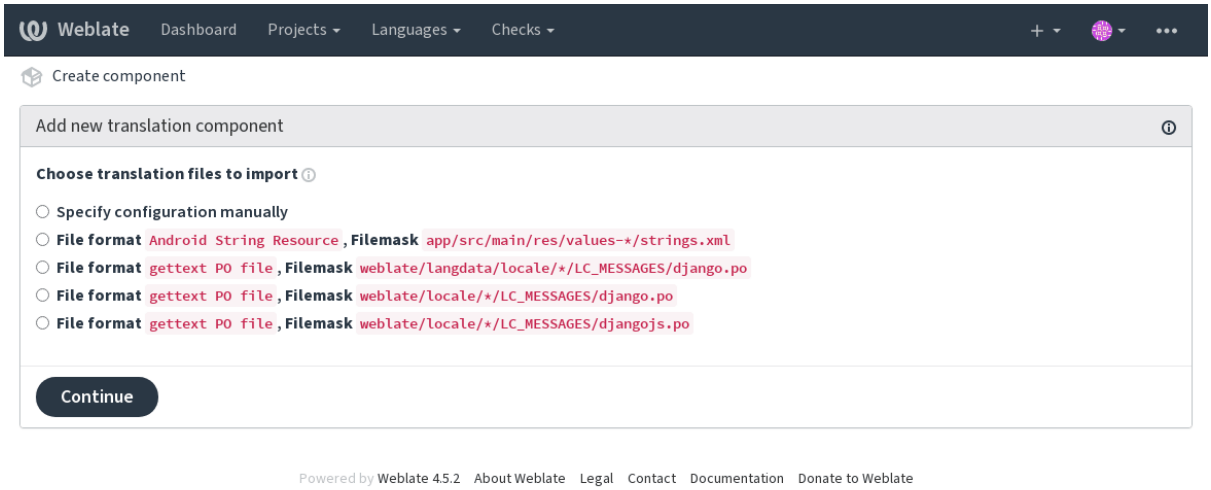
Начать с нуля Создать пустой проект перевода и добавить строки вручную.

Имея существующие компоненты перевода, вы также можете легко добавлять новые для дополнительных файлов или веток из того же репозитория.

Сначала вам нужно указать наименование и расположение репозитория:



На следующей странице вам будет представлен список обнаруженных переводимых ресурсов:



На последнем этапе просмотрите информацию о компоненте перевода и укажите дополнительные сведения:

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.

File format

gettext PO file

Filemask

weblate/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

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 filemask.

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 4.5.2 About Weblate Legal Contact Documentation Donate to Weblate

См.также:

Интерфейс администратора Django, *Настройки проекта*, *Настройки компонента*

2.7.3 Настройки проекта

Создайте проект перевода, а затем добавьте в него новый компонент, в котором вы и будете непосредственно делать перевод. Проект здесь похож на полку, на которую укладываются реальные переводы. Все компоненты одного проекта разделяют между собой общие предложения и глоссарий; переводы также автоматически распространяются во все компоненты одного проекта (если только оно не отключено в настройках компонентов), смотрите раздел *Память переводов*.

См.также:

/devel/integration

Это основные атрибуты для настройки, они предоставляют переводчикам информацию о проекте:

Название проекта

Подробное наименование проекта, используемое для его отображения.

URL-плашка

Название проекта, подходящее для указания его в URL-адресе.

Веб-сайт проекта

URL-адрес, по которому переводчики могут найти дополнительную информацию о проекте.

This is a required parameter unless turned off by `WEBSITE_REQUIRED`.

Инструкции для переводчиков

URL-адрес на сайт, содержащий более подробные инструкции для переводчиков.

Задать заголовок «Language-Team»

Должен ли Weblate управлять заголовком `Language-Team` (сейчас это единственная особая возможность для *GNU gettext*).

Использовать совместную память переводов

Использовать ли совместно используемую память переводов, подробнее смотрите в разделе *Совместная память переводов*.

Default value is determined by `DEFAULT_SHARED_TM`.

Вносить вклад в совместную память переводов

Помещать ли переводы в совместно используемую память переводов, подробнее смотрите в разделе *Совместная память переводов*.

Default value is determined by `DEFAULT_SHARED_TM`.

Управление доступом

Настройка контроля доступа для каждого проекта, подробнее смотрите в разделе *Управление доступом к проекту*.

Значение по умолчанию можно изменить установкой параметра `DEFAULT_ACCESS_CONTROL`.

Включить рецензии

Включение процесса рецензирования переводов, смотрите раздел *Выделенные рецензенты*.

Включить рецензии исходных строк

Включение процесса рецензирования исходных строк, смотрите раздел *Рецензирование исходных строк*.

См.также:

report-source, *Комментарии*

Включить обработчики

Должны ли для этого репозитория использоваться неаутентифицированные *обработчики уведомлений*.

См.также:

Файл промежуточного языка, *Контроль качества исходных строк*, *Двуязычные и одноязычные форматы*, *Определение языков*

Псевдонимы языка

Определяет отображение кодов языков при импорте переводов в Weblate. Используйте этот параметр, если коды языков в ваших репозиториях не согласуются друг с другом, а вы хотите, чтобы в Weblate они были представлены единообразно; или если вы хотите использовать нестандартные имена файлов перевода.

Типовым случаем использования псевдонимов, например, является отображение американского английского языка на английский: `en_US:en`

Несколько отображений должны разделяться запятой: `en_GB:en,en_US:en`

Использование нестандартного кода: `ru_FOO:ru`

Подсказка: Коды языков отображаются при успешном сопоставлении с именами файлов перевода, а оно чувствительно к регистру, поэтому убедитесь, что вы используете коды исходного языка в той же форме, в которой они используются в именах файлов.

См.также:

Разбор кодов языков

2.7.4 Настройки компонента

Компонент — это группировка каких-то сущностей для перевода. Вы вводите расположение репозитория системы контроля версий и маску файлов, определяющую, какие файлы вы хотите переводить, а Weblate автоматически извлекает код из этого репозитория и находит все соответствующие файлы для перевода.

См.также:

/devel/integration

Некоторые примеры типовых конфигураций можно найти в разделе *Поддерживаемые форматы файлов*.

Примечание: Рекомендуется держать размер компонентов перевода в разумных пределах — разбивайте перевод по любому критерию, который в вашем случае имеет смысл (по отдельным приложениям или дополнениям, по главам книг или по веб-сайтам).

Weblate легко справляется с переводами с 10000 строк, но с такими объёмами труднее разделять работу между несколькими переводчиками и координировать их действия.

Если определение языка для перевода отсутствует, создаётся пустое определение с именем вида «ru_RU (generated)». Вы должны скорректировать определение и сообщить об этом авторам Weblate, чтобы недостающие языки могли быть включены в следующий выпуск.

Компонент содержит все важные параметры для работы с системой контроля версий, а также для получения из неё переводов:

Название компонента

Подробное наименование компонента, используемое для его отображения.

Плашка компонента

Название компонента, подходящее для указания его в URL-адресе.

Проект компонента

Настройки проекта, которому принадлежит компонент.

Система контроля версий

Используемая система контроля версий, подробнее смотрите в разделе *Интеграция с системой контроля версий*.

См.также:

Отправка изменений из Weblate'а

Репозиторий исходного кода

Репозиторий системы контроля версий, из которого извлекаются изменения.

См.также:

Для получения более подробной информации о том, как указывать URL-адреса, смотрите раздел *Доступ к репозиториям*.

Подсказка: Это может быть либо реальный URL-адрес системы контроля версий, либо адрес вида `weblate://проект/компонент`, указывающий на то, что репозиторий должен совместно использоваться с другим компонентом. Для получения более подробной информации обратитесь к разделу *Внутренние URL-адреса Weblate*.

URL для отправки в репозиторий

URL-адрес репозитория, используемый для отправки изменений. Этот параметр используется только для систем контроля версий *Git* и *Mercurial* и если он пуст, поддержка отправки в них отключается.

См.также:

Для получения более подробной информации о том, как указывать URL-адрес репозитория, смотрите раздел *Доступ к репозиториям*, а для получения более подробной информации об отправке изменений из Weblate'а обратитесь к разделу *Отправка изменений из Weblate'а*.

Просмотрщик репозитория

URL-адрес браузера репозитория, используемый для отображения исходных файлов (местоположения используемых сообщений). Если пустой, такие ссылки генерироваться не будут. Вы можете использовать *шаблонную разметку*.

Например, для ссылок на GitHub, используйте что-то наподобие: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

В случае, если ваши пути указываются относительно другого каталога, вы можете удалить ведущий каталог фильтром `parentdir` (смотрите раздел *Шаблонная разметка*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

URL экспортированного репозитория

URL-адрес, куда экспортируются изменения, сделанные в Weblate. Он имеет значение, если не используется *непрерывная локализация* или в том случае, когда есть необходимость ручного слияния изменений. Для автоматизации этого процесса для Git-репозиториев вы можете использовать *экспортер Git*.

Ветка репозитория

Какую ветку извлекать из системы контроля версий и где искать переводы.

Ветка для отправки

Ветка, в которую отправлять изменения, оставьте пустой для использования параметра *Ветка репозитория*.

Примечание: В настоящее время параметр поддерживается только для Git'a, GitLab'a и GitHub'a, для интеграций с другими системами контроля версий он игнорируется.

См.также:

Отправка изменений из Weblate'a

Маска файла

Маска файлов для перевода, включая путь. Она должна включать в себя один символ `*`, который заменяется на код языка (для получения информации о том, как он обрабатывается, смотрите раздел *Определение языков*). В случае, если в вашем репозитории содержится более одного файла перевода (например, несколько доменов gettext), вам необходимо создать компонент для каждого из них.

Например, `po/*.po` или `locale/*/LC_MESSAGES/django.po`.

В случае, если имя вашего файла содержит такие специальные символы, как `[` или `]`, они должны быть экранированы как `[[]` и `[]]`.

См.также:

Двуязычные и одноязычные форматы, *Что означает сообщение «Есть больше файлов для одного языка (en)»?*

Одноязычный базовый файл

Базовый файл, содержащий определения строк для *одноязычных компонентов*.

См.также:

Двуязычные и одноязычные форматы, *Что означает сообщение «Есть больше файлов для одного языка (en)»?*

Правка базового файла

Разрешать ли редактирование базового файла для *одноязычных компонентов*.

Файл промежуточного языка

Файл промежуточного языка для *одноязычных компонентов*. В большинстве случаев это файл перевода предоставляется разработчиками и именно из него берутся настоящие исходные строки для перевода на другие языки.

Если он задан, то исходные строки будут взяты из этого файла, а переводы на все остальные языки — из *одноязычных базовых файлов*. В случае, если строка не «переведена» на исходный язык, перевод на другие языки запрещается. Этот запрет можно использовать для *контроля качества исходных строк*.

См.также:

Контроль качества исходных строк, *Двуязычные и одноязычные форматы*, *Что означает сообщение «Есть больше файлов для одного языка (en)»?*

Шаблон для новых переводов

Базовый файл, используемый для генерирования новых переводов, например, `.pot` файл для gettext.

Подсказка: Для многих одноязычных форматов Weblate по умолчанию начинает перевод с пустого файла. Используйте этот параметр, если вы хотите, чтобы при создании нового перевода присутствовали все строки с пустым значением.

См.также:

adding-translation, *Добавление новых переводов*, *Добавление нового перевода*, *Двуязычные и одноязычные форматы*, *Что означает сообщение «Есть больше файлов для одного языка (en)»?*

Формат файла

Формат файла перевода, смотрите также *поддерживаемые форматы файлов*.

Адрес для сообщения об ошибке в исходной строке

Адрес электронной почты, используемый для отправки сообщений об ошибках в исходных строках. На этот адрес также будут отсылаться уведомления о любых сделанных в Weblate комментариях к исходной строке.

Разрешить распространение перевода

Вы можете отключить распространение переводов на этот компонент из других компонентов этого же проекта. Это сильно зависит от того, что вы переводите, иногда желательно использовать разные переводы для одинаковых строк.

Обычно для одноязычных переводов рекомендуется отключать этот параметр, если только вы не используете во всем проекте одни и те же идентификаторы для одинаковых строк.

Значение по умолчанию может быть изменено параметром `DEFAULT_TRANSLATION_PROPAGATION`.

Включить предложения

Принимать ли предложения по переводу этого компонента.

Голосование по предложению

Включает голосование за предложения, смотрите раздел *Голосование по предложению*.

Автоматическое принятие предложений

Автоматически принимать предложения, за которые проголосовали, смотрите раздел *Голосование по предложению*.

Флаги перевода

Настройка проверок качества и другого поведения Weblate'а, смотрите раздел *Настройка поведения с помощью флагов*.

Принудительные проверки

Список проверок, которые нельзя проигнорировать, смотрите раздел *Принудительные проверки*.

Примечание: Добавление проверки в качестве принудительной не включает её автоматически; вам всё равно придётся включить её с помощью *настроек поведения* во *флагах перевода* или *дополнительной информации об исходных строках*.

Лицензия перевода

Лицензия на перевод (не обязательно должна совпадать с лицензией на исходный код).

Соглашение участника

Пользовательское соглашение, которое должно быть одобрено, прежде чем пользователь сможет переводить этот компонент.

Добавление нового перевода

Как обрабатывать запросы на создание новых языков. Доступные варианты:

Связаться с сопровождающими Пользователь может выбрать желаемый язык, а сопровождающие проекта получат об этом уведомление. Добавление (или не добавление) языка в репозиторий зависит от них.

Указать URL на инструкции по переводу Пользователю предоставляется ссылка на страницу, которая описывает процесс начала новых переводов. Используйте этот вариант в случае, если вам требуется более формальный процесс (например, формирование команды людей перед началом фактического перевода).

Создать новый язык Пользователь может выбрать язык и Weblate автоматически создаст для него файл, так что можно сразу начинать переводить.

Запретить добавление новых переводов У пользователя не будет возможности начать новый перевод.

Подсказка: The project admins can add new translations even if it is disabled here when it is possible (either *Шаблон для новых переводов* or the file format supports starting from an empty file).

См.также:

adding-translation, *Добавление новых переводов*

Управление строками

Добавлено в версии 4.5.

Настраивает, смогут ли пользователи добавлять новые строки и удалять существующие из Weblate. В зависимости от вашего рабочего процесса (как именно должны добавляться новые строки) вы можете включить этот параметр или оставить отключённым.

В двуязычных форматах строки обычно извлекаются из исходного кода (например, с помощью **xgettext**) и добавление новых строк через Weblate должно быть отключено (они всё равно удалятся, когда вы в следующий раз будете обновлять свои файлы перевода). В Weblate вы можете управлять строками для каждого перевода отдельно, это не будет принудительно поддерживать набор исходных строк для всех языков одинаковым.

Для одноязычных форматов управление строками происходит только на исходном языке, в переводах они добавляются и удаляются автоматически. В файлах перевода строки появляются только после их фактического перевода.

См.также:

Двуязычные и одноязычные форматы, adding-new-strings, `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Стиль кода языка

Настройте код языка, используемый для генерации имени файла для переводов, созданных Weblate.

См.также:

Добавление новых переводов, *Код языка*, *Разбор кодов языков*

Стиль слияния

Вы можете настроить обработку обновлений из вышестоящего репозитория. Для некоторых систем контроля версий этот параметр может не поддерживаться. За более подробной информацией обратитесь к разделу *Слияние или перебазирование*.

Значение по умолчанию можно изменить установкой параметра `DEFAULT_MERGE_STYLE`.

Сообщения коммита, добавления, удаления, слияния и сообщения надстроек

Сообщение, используемое при коммите перевода, смотрите раздел *Шаблонная разметка*.

Значения по умолчанию можно изменить установкой параметров `DEFAULT_ADD_MESSAGE`, `DEFAULT_ADDON_MESSAGE`, `DEFAULT_COMMIT_MESSAGE`, `DEFAULT_DELETE_MESSAGE` и `DEFAULT_MERGE_MESSAGE`.

Отправлять при коммите

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 *Отложенные коммиты*). To actually enable pushing *Repository push URL* has to be configured as well.

Возраст изменений для коммита

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.

Значение по умолчанию можно изменить установкой параметра `COMMIT_PENDING_HOURS`.

Подсказка: There are other situations where pending changes might be committed, see *Отложенные коммиты*.

Блокировать при ошибке

Locks the component (and linked components, see *Внутренние URL-адреса 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.

Компонент будет автоматически разблокирован, как только в репозитории не останется ошибок.

Исходный язык

Язык, используемый для исходных строк. Изменяйте этот параметр, если вы переводите не с английского.

Подсказка: 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.

Для одноязычных переводов в данном случае можно использовать промежуточный перевод, смотрите раздел *Файл промежуточного языка*.

Языковой фильтр

Regular expression used to filter the translation when scanning for filemask. It can be used to limit the list of languages managed by Weblate.

Примечание: Вам необходимо перечислить коды языков в том же виде, в котором они встречаются в именах файлов.

Несколько примеров фильтров:

Описание фильтра	Регулярное выражение
Только выбранные языки	`^(cs|de|es)$`
Исключить языки	`^(?!(it|fr)$).+$`
Выбрать только двубуквенные языковые коды	`^..$`
Исключить неязыковые файлы	`^(?!(blank)$).+$`
Включить все файлы (по умолчанию)	`^[^.]+$`

Регулярное выражение вариантов

Регулярное выражение, используемое для определения вариантов строки, смотрите раздел variants.

Примечание: Большинство полей могут быть отредактированы владельцами или управляющими проекта через интерфейс Weblate'а.

См.также:

Поддерживает ли Weblate другие системы контроля версий, кроме Git и Mercurial?, alerts

Приоритет

Переводчикам вначале предлагаются компоненты с большим приоритетом.

Ограниченный доступ

По умолчанию компонент виден любому, кто имеет доступ к проекту, даже если этот человек не может выполнить в компоненте никаких изменений. Это облегчает поддержание согласованности перевода в рамках проекта.

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`.

Подсказка: This applies to project admins as well — please make sure you will not loose access to the component after toggling the status.

Разделяемые компоненты

You can choose additional projects where the component will be visible. Useful for shared libraries which you use in several projects.

Примечание: 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.

Использовать в качестве глоссария

Добавлено в версии 4.5.

Позволяет использовать данный компонент в качестве глоссария. Вы можете настроить, как именно он будет отображаться, с помощью параметра «*Цвет глоссария*».

Глоссарий будет доступен во всех проектах, в которые он добавлен в качестве *разделяемого компонента*.

Для словарей также рекомендуется включить *управление строками*, чтобы в в них можно было добавлять новые слова.

См.также:

Глоссарий

Цвет глоссария

Цвет отображения для названия глоссария, используемый в списке терминов, найденных в переводе.

2.7.5 Шаблонная разметка

В нескольких местах, где необходима отрисовка текста, Weblate использует простой язык разметки. Он основан на языке шаблонов Django, так что он может быть достаточно мощным.

В настоящее время он используется:

- Для форматирования сообщений коммитов, смотрите раздел *Настройки компонента*
- **В нескольких надстройках**
  - *Обнаружение компонентов*
  - *Генератор статистики*
  - *Выполнение скриптов из надстройки*

В шаблонах компонентов доступны следующие переменные:

`{{ language_code }}` Код языка

`{{ language_name }}` Название языка

`{{ component_name }}` Название компонента

`{{ component_slug }}` Плашка компонента

`{{ project_name }}` Название проекта

`{{ project_slug }}` Плашка проекта

`{{ url }}` URL-адрес перевода

`{{ filename }}` Имя файла перевода

`{{ stats }}` Статистика перевода, у неё есть дополнительные атрибуты, примеры ниже.

`{{ stats.all }}` Общее количество строк

`{{ stats.fuzzy }}` Количество строк, требующих рецензирования

`{{ stats.fuzzy_percent }}` Процент строк, требующих рецензирования

`{{ stats.translated }}` Количество переведённых строк

`{{ stats.translated_percent }}` Процент переведённых строк

`{{ stats.allchecks }}` Количество строк с неудачными проверками

`{{ stats.allchecks_percent }}` Процент строк с неудачными проверками

`{{ author }}` Автор текущего коммита, доступна только в области действия коммита.

`{{ addon_name }}` Имя исполняемой в данный момент надстройки, доступна только в сообщении о коммите надстройки.

Следующие переменные доступны в шаблонах браузера репозитория или редактора:

`{{branch}}` текущая ветка

`{{line}}` строка в файле

`{{filename}}` имя файла, также с помощью фильтра `parentdir` вы можете удалить ведущие части, например `{{filename|parentdir}}`

Вы можете объединять их с фильтрами:

```
{{ component|title }}
```

Можете использовать условия:

```
{% if stats.translated_percent > 80 %}Well translated!{% endif %}
```

Для замены символов доступен дополнительный тег:

```
{% replace component "-" " " %}
```

Вы можете объединить его с фильтрами:

```
{% replace component|capfirst "-" " " %}
```

Также есть дополнительный фильтр для работы с именами файлов:

```
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 }}
```

…и другие возможности шаблонов Django.

2.7.6 Скорость импорта

Извлечение из репозитория системы контроля версий и импорт переводов в Weblate может быть длительным процессом, в зависимости от размера ваших переводов. Вот несколько советов:

Оптимизация настроек

Конфигурация по умолчанию полезна для тестирования и отладки Weblate, в то же время для установки в рабочую среду в неё необходимо внести некоторые коррективы. Многие из параметров довольно сильно влияют на производительность. Для получения более подробной информации обратитесь к разделу *Рабочая среда*, особенно обратите внимание на следующие темы:

- Настройка Celery для выполнения фоновых задач (смотрите раздел *Фоновые задачи с использованием Celery*)
- *Включение кэширования*
- *Использование мощного движка базы данных*
- *Отключение отладочного режима*

Проверка ограничения по ресурсам

Если вы импортируете огромные переводы или репозитории, вы можете столкнуться с ограничениями по ресурсам вашего сервера.

- Проверьте объём свободной памяти, наличие кэшированных операционной системой файлов перевода значительно повысит производительность.
- Если нужно обработать множество строк, дисковые операции могут стать узким местом — диск насилуется как Weblate'ом, так и базой данных.
- Улучшить производительность фоновых задач могут помочь дополнительные ядра процессора (смотрите раздел *Фоновые задачи с использованием Celery*).

Отключение ненужных проверок

Некоторые проверки качества могут быть довольно дорогими, и если в них нет необходимости, при импорте можно сэкономить некоторое время, если их пропустить. Информацию по их настройке смотрите в описании параметра `CHECK_LIST`.

2.7.7 Автоматическое создание компонентов

Если в вашем проекте есть куча файлов перевода (например, различные домены gettext или части разных Android приложений), вы можете захотеть импортировать их автоматически. Этого можно сделать либо из командной строки с помощью команд `import_project` или `import_json`, либо с помощью установки надстройки *Обнаружение компонентов*.

Для использования надстройки вам сначала нужно создать компонент для одного файла перевода (выберите тот, который с наименьшей вероятностью в будущем будет переименован или удалён), и установить надстройку в этот компонент.

При использовании команд управления вам необходимо создать проект, который будет содержать все компоненты, а затем выполнить команду `import_project` или `import_json`.

См.также:

Команды управления, *Обнаружение компонентов*

2.8 Определение языков

Для правильного представления различных переводов необходима информация о названии языка, направлении текста, определения форм множественного числа и код языка.

2.8.1 Разбор кодов языков

При разборе переводов Weblate пытается сопоставить код языка (обычно по ISO 639-1) с любым существующим объектом языка.

Дальнейшую настройку этого сопоставления на уровне проекта можно осуществить с помощью *псевдонимов языков*.

Если точное совпадение не может быть найдено, будет сделана попытка подогнать его под наиболее подходящий существующий язык. Следующие попытки будут предприняты:

- Регистронезависимый поиск.
- Нормализация подчёркиваний и дефисов.
- Поиск по встроенным псевдонимам языка.
- Поиск по названию языка.
- Игнорирование кода страны по умолчанию для данного языка — выбором `ru` вместо `ru_RU`.

Если и это также не получится, будет создано новое определение языка с использованием настроек по умолчанию (направление текста слева направо, одна форма множественного числа). Такой автоматически созданный язык с кодом `xx_XX` будет иметь имя *xx_XX (generated)*. Возможно, позже вы захотите изменить его в интерфейсе администратора (смотрите раздел *Изменение определений языка*) и сообщить об этом в систему отслеживания ошибок (смотрите раздел *Как внести свой вклад в Weblate*), чтобы соответствующее определение было добавлено в последующих выпусках Weblate.

Подсказка: В случае, если вы видите в качестве языка что-то нежелательное, вам может понадобиться настроить *языковой фильтр* на игнорирование такого файла при разборе переводов.

См.также:

Код языка, *Добавление новых переводов*

2.8.2 Изменение определений языка

Вы можете изменить определения языка в интерфейсе по языкам (по адресу `/languages/`).

Во время редактирования убедитесь, что все поля заполнены правильно (особенно формы множественного числа и направление текста), иначе переводчики не смогут правильно редактировать эти переводы.

2.8.3 Встроенные определения языков

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 *Общие инструкции по обновлению*) the database of languages is updated to include all language definitions shipped in Weblate.

Это можно отключить, отключив параметр *`UPDATE_LANGUAGES`*. Вы также можете принудительно обновить базу данных, чтобы она соответствовала встроенной БД Weblate с помощью *`setuplang`*.

См.также:

The language definitions are in the weblate-language-data repository.

2.8.4 Неоднозначные коды языков и макроязыки

В большинстве случаев использовать код макроязыка для перевода не является хорошей идеей. Типичный проблемный случай — это курдский язык, разные варианты которого могут записываться либо латиницей, либо арабским письмом. Чтобы добиться корректного поведения в Weblate рекомендуется использовать только индивидуальные коды языков, избегая макроязыки.

См.также:

Что такое макроязык (англ.), Список макроязыков

2.8.5 Определение языков

Каждый язык состоит из следующих полей:

Код языка

Код, идентифицирующий язык. Weblate предпочитает двухбуквенные коды, как они определены в ISO 639-1, но для языков, для которых нет двухбуквенных кодов, использует коды ISO 639-2 или ISO 639-3. Также он может поддерживать расширенные коды, определенные в BCP 47.

См.также:

Разбор кодов языков, *Добавление новых переводов*

Название языка

Видимое название языка. Названия языков, включённых в Weblate, также локализуются в зависимости от языка пользовательского интерфейса.

Направление текста

Определяет, идёт ли запись в языке справа налево или слева направо. Для большинства языков это свойство корректно определяется автоматически.

Количество форм множественного числа

Количество используемых в языке форм множественного числа.

Формула множественного числа

Совместимая с Gettext формула множественного числа, используемая для определения того, какая форма множественного числа используется для данного количества.

См.также:

Формы множественного числа, Утилиты GNU gettext: формы множественного числа, Языковые правила для форм множественного числа от Консорциума Юникода

2.8.6 Добавление новых переводов

Изменено в версии 2.18: В версиях до 2.18 поведение добавления новых переводов зависело от формата файла.

Weblate может автоматически начинать новый перевод для всех форматов файлов.

Некоторые форматы ожидают, что они будут начинаться с пустого файла и в него будут включены только переведённые строки (например, *Строковые ресурсы Android*), в то время как другие ожидают, что в файле будут присутствовать все ключи (например, *GNU gettext*). В некоторых ситуациях это на самом деле зависит не от формата, а от фреймворка, который вы используете для работы с файлами перевода (например, с *файлами JSON*).

При указании в *конфигурации компонента* файла *шаблона нового перевода* Weblate будет использовать его для начинания новых переводов. Все существующие переводы при этом будут из файла удалены.

Если *шаблон для новых переводов* не задан и формат файла поддерживает его, то создаётся пустой файл, в который будут добавляться новые строки по мере их перевода.

Параметр «*Стиль кода языка*» позволяет вам настраивать код языка, используемый в именах сгенерированных файлов:

Умолчание, основанное на формате файла Зависит от формата файла, для большинства форматов используется стиль POSIX.

Стиль POSIX, использующий подчёркивание как разделитель Обычно используется gettext'ом и связанными с ним инструментами, выдаёт коды языков вида `pt_BR`.

Стиль POSIX, использующий подчёркивание как разделитель и включающий код страны Код языка в стиле POSIX, включающий код страны, даже если в этом нет необходимости (например, `ru_RU`).

Стиль BCP, использующий в качестве разделителя дефис Обычно используется на веб-платформах, выдаёт коды языков вида `pt-BR`.

Стиль BCP, использующий в качестве разделителя дефис и включающий код страны Код языка в стиле BCP, включающий код страны, даже если в этом нет необходимости (например, ``ru-RU`).

Стиль Android Используется только в приложениях для Android, выдаёт коды языков вида `pt-rBR`.

Стиль Java Используется Java — по сути это тот же BCP, но с устаревшими кодами для китайского языка.

В дополнение к этому, любое отображение, определённое как *псевдоним языка*, применяется в обратном направлении.

Примечание: Weblate распознаёт любой из этих стилей при разборе файлов перевода, вышеуказанные настройки влияют только на то, в каком стиле создавать новые файлы.

См.также:

Код языка, *Разбор кодов языков*

2.9 Непрерывная локализация

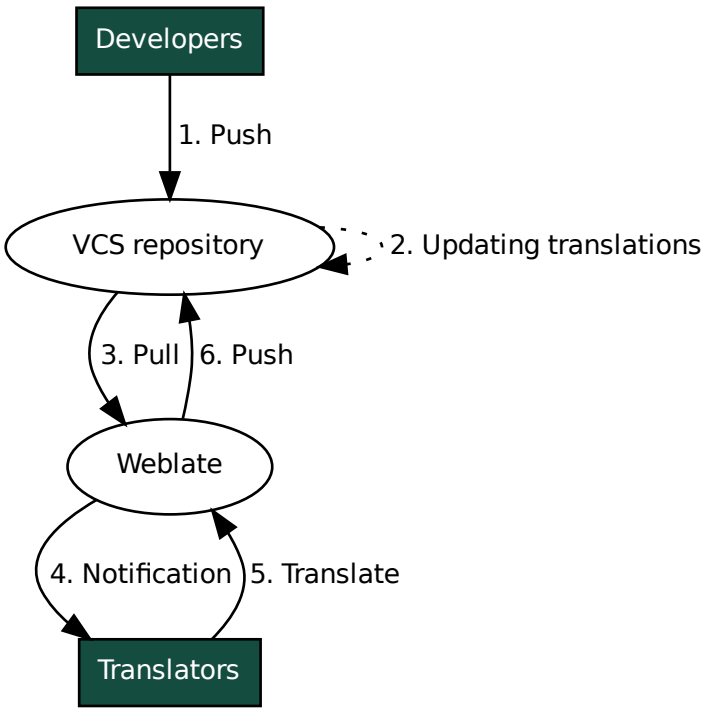
Weblate строит такую инфраструктуру, чтобы ваш перевод непрерывно следовал за разработкой. Таким образом, переводчики могут работать над переводами всё время разработки, вместо того, чтобы работать с огромным количеством нового текста непосредственно перед выпуском.

См.также:

В разделе «/devel/integration» на базовом уровне описывается, как интегрировать ваш процесс разработки с Weblate.

Процесс следующий:

1. Разработчики вносят изменения и отправляют их в репозиторий системы контроля версий.
2. По необходимости обновляются файлы перевода (это зависит от формата файла, смотрите вопрос *Почему Weblate до сих пор показывает старые строки перевода, хотя я уже обновил шаблон?*).
3. Weblate извлекает изменения из репозитория системы контроля версий, смотрите раздел *Обновление репозиториев*.
4. Как только Weblate обнаруживает изменения в переводах, он уведомляет об этом переводчиков в соответствии с настройками их подписок.
5. Переводчики присылают переводы через веб-интерфейс Weblate, или загружают файлы, изменённые ими в автономном режиме.
6. После завершения работы переводчиков Weblate коммитит изменения в локальный репозиторий (смотрите раздел *Отложенные коммиты*) и отправляет их обратно, если у него есть на это разрешение (смотрите раздел *Отправка изменений из Weblate'а*).



2.9.1 Обновление репозиториев

Вы должны каким-то образом настроить обновления репозиториев из их источников.

- Use *Обработчики уведомлений* to integrate with most of common code hosting services:
  - *Автоматическое получение изменений из GitHub'а*
  - *Автоматическое получение изменений из GitLab'а*
  - *Автоматическое получение изменений из Bitbucket'а*
  - *Автоматическое получение изменений из Pagure*
  - *Автоматическое получение изменений из Azure Repos*
- Вручную запускайте обновление либо в разделе управления репозиторием, либо с помощью *REST API Weblate'а* или *клиента Weblate*
- Включите параметр `AUTO_UPDATE` для автоматического обновления всех компонентов на вашем экземпляре Weblate'а
- Выполните команду `updategit` (с выбранным проектом проекта или с ключом *–all* для обновления всех проектов)

Всякий раз, когда Weblate обновляет репозиторий, будут срабатывать надстройки «после обновления», смотрите раздел *Надстройки*.

Предотвращение конфликтов слияния

Конфликты слияния из Weblate возникают, когда один и тот же файл изменяется как в Weblate'е, так и за его пределами. Существует два подхода к решению этой проблемы — избегать редактирования файлов вне Weblate'а или интегрировать Weblate в процесс обновления таким образом, чтобы он фиксировал сделанные изменения перед обновлением файлов вне Weblate'а.

Первый подход прост с одноязычными файлами — вы можете добавлять новые строки, не выходя из Weblate'а, и возложить на него любое редактирование файлов. Для двуязычных файлов, как правило, существует некий процесс извлечения сообщений из исходного кода для генерации файлов перевода. В некоторых случаях он может быть разбит на две части — одна, извлекающая, часть генерирует шаблон (например, файл POT gettext'а генерируется с помощью программы **xgettext**), а дальнейший процесс объединяет его с фактическими переводами (файлы PO gettext'а, обновляемые с помощью программы **msgmerge**). Второй шаг вы можете выполнить в Weblate'е, и он удостоверится, что до выполнения этой операции все отложенные изменения были зафиксированы.

Второй подход может быть достигнут путём использования *REST API Weblate'а*, заставляющего Weblate отправить на все отложенные изменения и заблокировать перевод, пока вы на своей стороне проводите изменения.

Скрипт для выполнения обновлений может выглядеть следующим образом:

```
# 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
```

Если у вас есть несколько компонентов, совместно использующих один и тот же репозиторий, вам необходимо заблокировать их все по отдельности:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Примечание: В примере используется *клиент Weblate*, который, чтобы иметь возможность управлять Weblate'ом удалённо, нуждается в некоторой настройке (в ключах API). Также вы можете решить задачу при помощи любого HTTP-клиента вместо wlc, к примеру, curl, смотрите раздел *REST API Weblate'а*.

См.также:

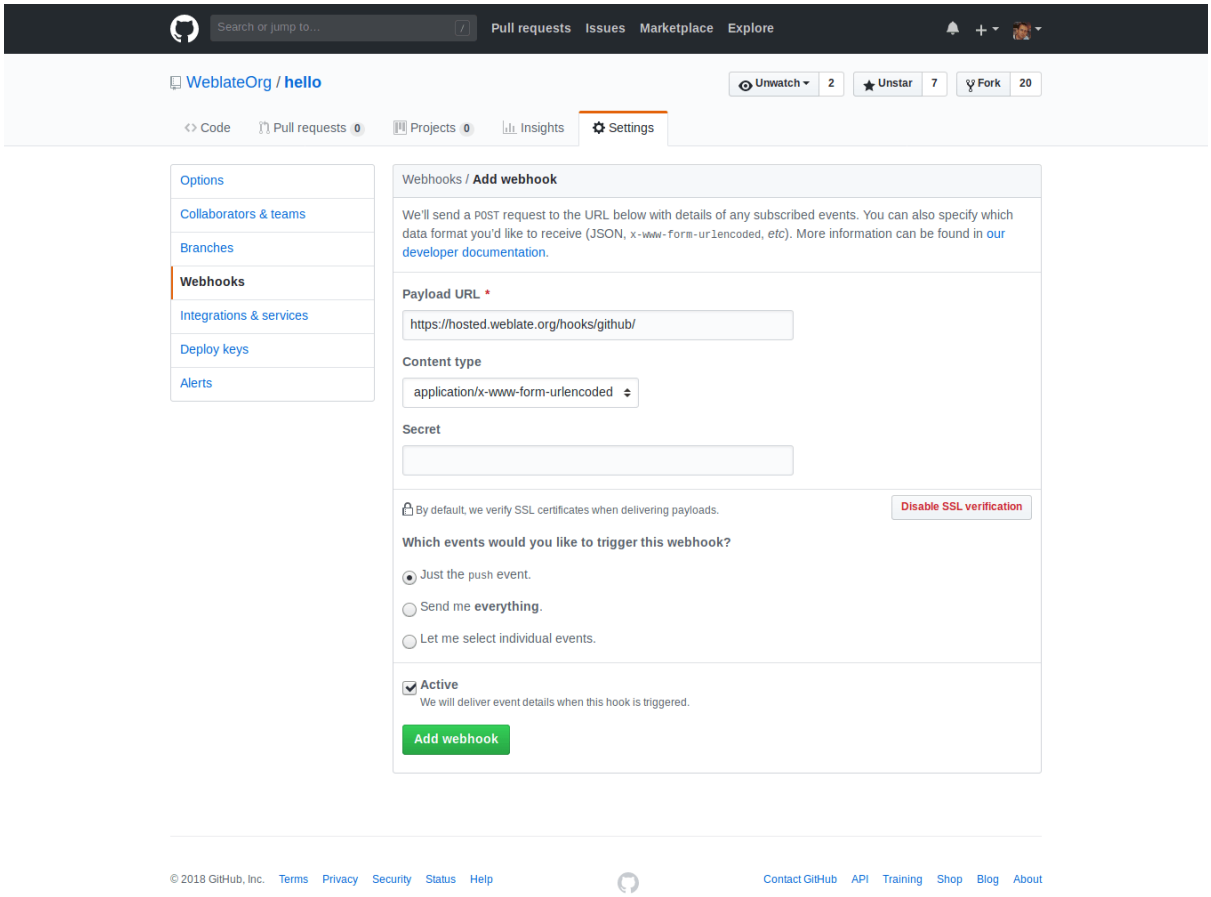
Клиент Weblate

Автоматическое получение изменений из GitHub'а

Weblate поставляется со встроенной поддержкой GitHub'а.

Если вы используете Hosted Weblate, рекомендуемый подход заключается в установке приложения Weblate, таким образом вы получите правильную настройку без необходимости в дополнительной настройке. Также оно может быть использовано для отправки изменений обратно в GitHub.

Для получения уведомлений о каждой отправке в репозиторий GitHub, добавьте в настройки репозитория (*Webhooks*) веб-обработчик Weblate, как показано на изображении ниже:



Для заполнения адреса полезной нагрузки (поля *Payload URL*), добавьте к URL-адресу вашего Weblate'а суффикс `/hooks/github/`, например, для сервиса Hosted Weblate, это этот адрес будет таким: `https://hosted.weblate.org/hooks/github/`.

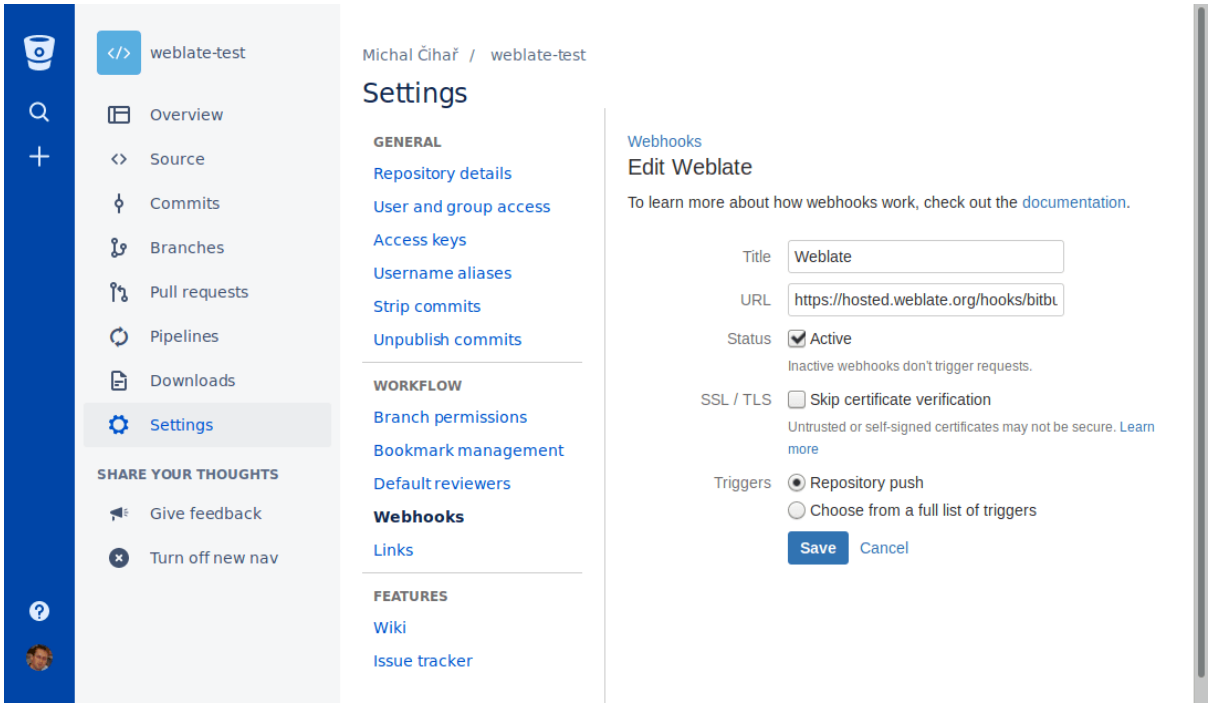
Остальные поля вы можете оставить с настройками по умолчанию (Weblate умеет обрабатывать оба типа содержимого и потребляет только событие *push*).

См.также:

`POST /hooks/github/`, *Доступ к репозиториям из Hosted Weblate*

Автоматическое получение изменений из Bitbucket'a

Weblate поддерживает веб-обработчики Bitbucket, добавьте веб-обработчик, который срабатывает при отправке изменений в репозиторий, указав в качестве *URL* адрес вашей установки Weblate с суффиксом `/hooks/bitbucket/` (например, `https://hosted.weblate.org/hooks/bitbucket/`).



См.также:

`POST /hooks/bitbucket/`, *Доступ к репозиториям из Hosted Weblate*

Автоматическое получение изменений из GitLab'a

Weblate поддерживает обработчики GitLab'a, добавьте веб-обработчик проекта с адресом назначения, равным адресу вашей установки Weblate с суффиксом `/hooks/gitlab/` (например, `https://hosted.weblate.org/hooks/gitlab/`).

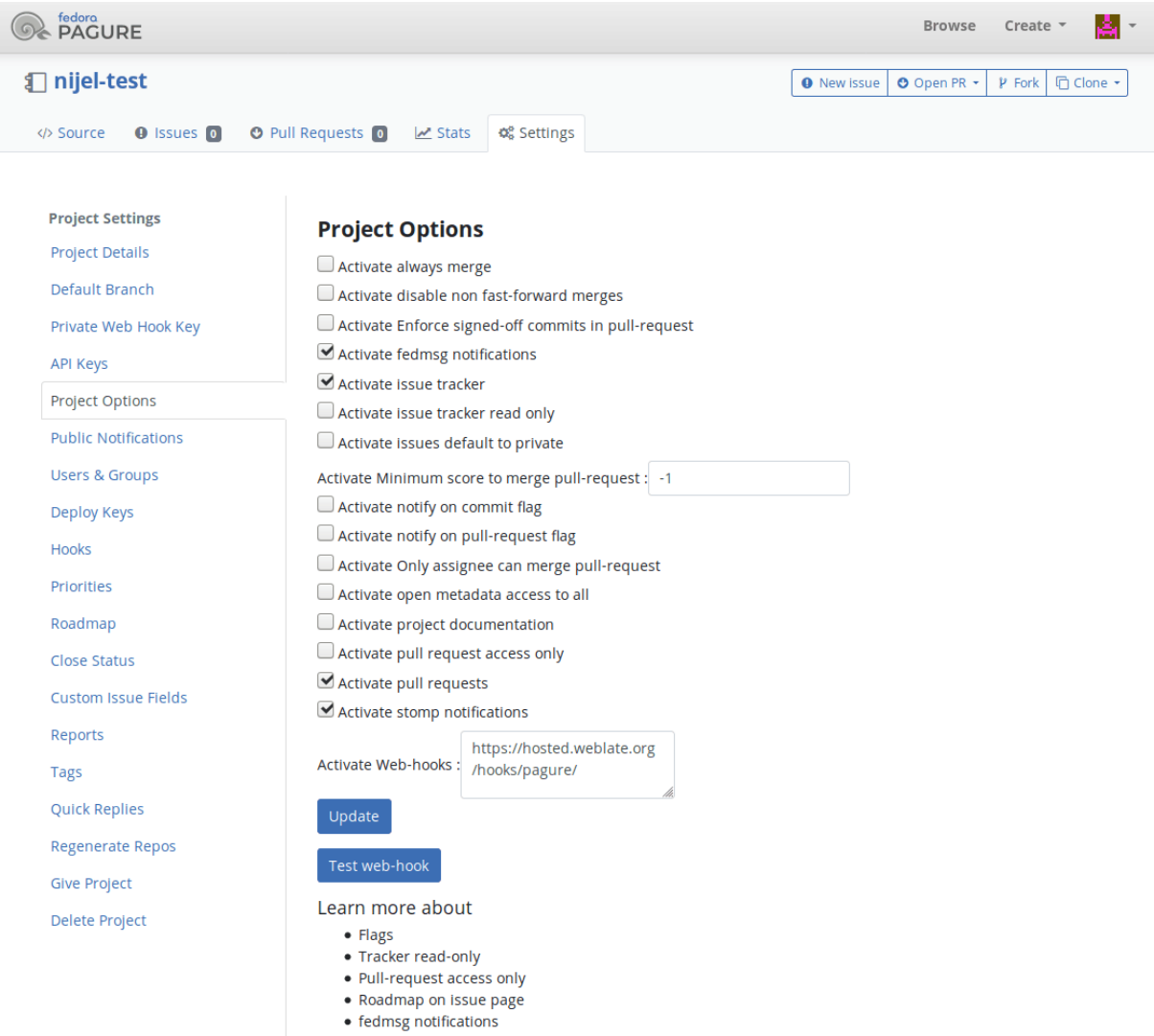
См.также:

`POST /hooks/gitlab/`, *Доступ к репозиториям из Hosted Weblate*

Автоматическое получение изменений из Pagure

Добавлено в версии 3.3.

Weblate поддерживает обработчики Pagure, добавьте веб-обработчик проекта с адресом назначения, равным адресу вашей установки Weblate с суффиксом `/hooks/pagure/` (например, `https://hosted.weblate.org/hooks/pagure/`). Добавлять этот адрес нужно в поле *Activate Web-hooks*, расположенное в блоке *Project options*:



См.также:

POST /hooks/pagure/, *Доступ к репозиториям из Hosted Weblate*

Автоматическое получение изменений из Azure Repos

Добавлено в версии 3.8.

Weblate поддерживает веб-обработчики Azure Repos, добавьте веб-обработчик для события *Code pushed* с адресом назначения, равным адресу вашей установки Weblate с суффиксом `/hooks/azure/` (например, `https://hosted.weblate.org/hooks/azure/`). Добавлять этот адрес нужно в поле *Service hooks*, расположенное в блоке *Project settings*.

См.также:

Веб-обработчики в руководстве по Azure DevOps, *POST /hooks/azure/*, *Доступ к репозиториям из Hosted Weblate*

Автоматическое получение изменений из репозиториев Gitea

Добавлено в версии 3.9.

Weblate поддерживает веб-обработчики Gitea, добавьте веб-обработчик *Gitea Webhook* для события *Push events* с адресом назначения, равным адресу вашей установки Weblate с суффиксом `/hooks/gitea/` (например, `https://hosted.weblate.org/hooks/gitea/`). Добавлять этот адрес нужно в поле *Webhooks*, расположенное в блоке *Settings*.

См.также:

Веб-обработчики в руководстве по Gitea, `POST /hooks/gitea/`, *Доступ к репозиториям из Hosted Weblate*

Автоматическое получение изменений из репозиториев Gitee

Добавлено в версии 3.9.

Weblate поддерживает веб-обработчики Gitee, добавьте веб-обработчик *WebHook* для события *Push* с адресом назначения, равным адресу вашей установки Weblate с суффиксом `/hooks/gitee/` (например, `https://hosted.weblate.org/hooks/gitee/`). Добавлять этот адрес нужно в поле *WebHooks*, расположенное в блоке *Management*.

См.также:

Веб-обработчики в руководстве по Gitee, `POST /hooks/gitee/`, *Доступ к репозиториям из Hosted Weblate*

Автоматическое ночное обновление репозиториев

Weblate по ночам автоматически извлекает изменения из удалённых репозиториев для повышения производительности при последующем слиянии изменений. По желанию вы можете превратить его и в ночное слияние, включив параметр `AUTO_UPDATE`.

2.9.2 Отправка изменений из Weblate'а

Каждый компонент перевода может иметь настроенный URL-адрес для отправки (смотрите описание параметра *URL для отправки в репозиторий*), в этом случае Weblate будет способен отсылать изменения в удалённый репозиторий. Также Weblate может быть настроен на автоматическую отсылку изменений при каждом коммите (это поведение по умолчанию, смотрите описание параметра *Отправлять при коммите*). Если вы не хотите, чтобы изменения отправлялись автоматически, вы можете отправлять их вручную в разделе *Обслуживание репозитория* или через API командой `wlc push`.

Параметры отправки отличаются в зависимости от используемой *системы контроля версий*, подробнее об этом читайте в указанной главе.

Если вам не хотите отправлять изменения из Weblate напрямую, то поддерживается также создание запросов на извлечение для *GitHub'а*, *GitLab'а*, *Pagure* или рецензий *Gerrit*. Вы можете активировать их, выбрав в *конфигурациях компонента* в качестве *системы контроля версий* *GitHub*, *GitLab* или *Gerrit* или *Pagure* соответственно.

В целом, для Git'а, GitHub'а и GitLab'а доступны следующие варианты:

Желаемая настройка	*Система контроля версий*	*URL для отправки в репозиторий*	*Ветка для отправки*
Без отправки	*Git*	*пусто*	*пусто*
Отправка напрямую	*Git*	URL-адрес SSH	*пусто*
Отправка в отдельную ветку	*Git*	URL-адрес SSH	Имя ветки
GitHub'овский запрос на извлечение из форка	*GitHub*	*пусто*	*пусто*
GitHub'овский запрос на извлечение из ветки	*GitHub*	URL-адрес SSH[1]	Имя ветки
GitLab 'овский запрос на слияние из форка	*GitLab*	*пусто*	*пусто*
GitLab 'овский запрос на слияние из ветки	*GitLab*	URL-адрес SSH[c. 63, 1]	Имя ветки
Pagure'вский запрос на слияние из форка	*Pagure*	*пусто*	*пусто*
Pagure 'вский запрос на слияние из ветки	*Pagure*	URL-адрес SSH[c. 63, 1]	Имя ветки

Примечание: Также вы можете включить автоматическую отправку изменений после коммитов Weblate'а, это можно сделать установкой параметра *Отправлять при коммите*.

См.также:

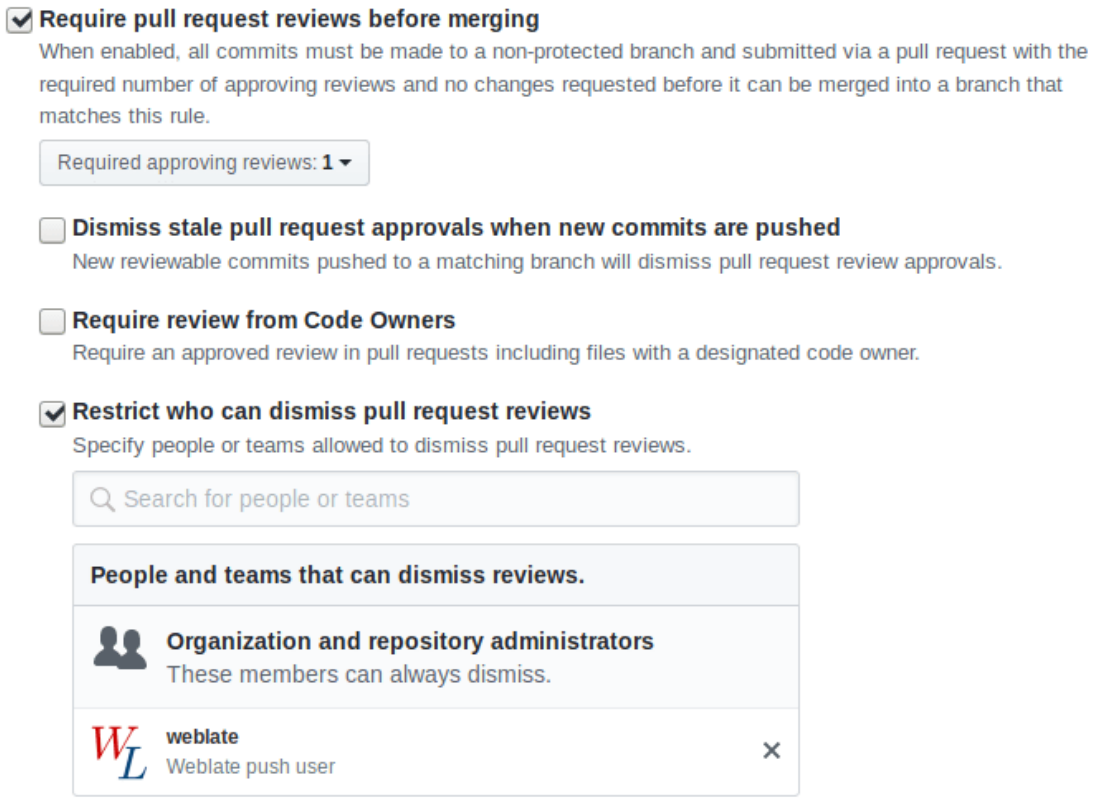
Для настройки ключей SSH смотрите раздел *Доступ к репозиториям*, а для получения информации о том, когда Weblate решает закоммитить изменения — раздел *Отложенные коммиты*.

Защищённые ветки

Если вы используете Weblate на защищённой ветке, вы можете настроить его на использование запросов на извлечение и выполнение рецензирования переводов (что может быть проблематично для языков, которых вы не знаете). Альтернативный подход заключается в отмене этого ограничения для пользователя Weblate.

Например, на GitHub'е это можно сделать в настройках репозитория:

[1] Может быть пустым в случае, если *репозиторий исходного кода* поддерживает отправку изменений.



2.9.3 Слияние или перебазирование

По умолчанию Weblate вливает вышестоящий репозиторий в свой собственный. Это самый безопасный способ в случае, если вы помимо этого работаете с нижележащим репозиторием и другими способами. Если вам это не нужно, вы можете включить перебазирование изменений на голову вышестоящего репозитория, что позволит создавать историю с меньшим количеством коммитов слияния.

Примечание: В случае сложных слияний перебазирование может доставить вам проблем, поэтому тщательно подумайте, хотите ли вы его включать или нет.

2.9.4 Взаимодействие с другими пользователями

Weblate облегчает взаимодействие с другими пользователями с помощью своего API.

См.также:

REST API Weblate'a

2.9.5 Отложенные коммиты

Поведение Weblate'а заключается в группировке коммитов одного и того же автора в один коммит (если это возможно). Такая группировка значительно сокращает количество коммитов, однако вам может понадобиться явно приказать ему сделать коммиты в случае, если вы хотите синхронизировать репозиторий, например, для выполнения слияния (по умолчанию это действие разрешено для группы *Управляющие*, смотрите раздел *Список привилегий*).

Изменения в этом режиме коммитятся при выполнении любого из следующих условий:

- Кто-то другой изменяет уже изменённую строку.
- Происходит слияние с вышестоящим репозиторием.
- Запрошен явный коммит изменений.
- Изменения старше периода, определённого *возрастом изменений для коммита* в *конфигурации компонента*.

Подсказка: Коммиты создаются для каждого компонента. Так что в случае, если у вас много компонентов, у вас всё равно будет множество коммитов. В этом случае вы можете использовать надстройку *Уплотнение Git-коммитов*.

Если вы хотите коммитить изменения чаще и без проверки их возраста, вы можете запланировать для выполнения коммита обычное задание:

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

2.9.6 Обработка репозитория скриптами

Настройка взаимодействия Weblate'а с репозиторием заключается в использовании *надстроек*. Для получения информации о том, как через надстройки выполнять внешние скрипты, обратитесь к разделу *Выполнение скриптов из надстройки*.

2.9.7 Поддержание единого перевода в разных компонентах

Если у вас несколько компонентов перевода, вы возможно захотите убедиться, что одни и те же строки имеют один и тот же перевод. Этого можно достичь на нескольких уровнях.

Распространение перевода

При включённой функции распространения перевода (которая включена по умолчанию, смотрите раздел *Настройки компонента*), все новые переводы автоматически копируются во все компоненты с совпадающими строками. Такие переводы должным образом засчитываются текущему пользователю-переводчику во всех компонентах.

Примечание: Распространение перевода требует, чтобы в одноязычных форматах файлов перевода ключи строк совпадали, так что об этом следует помнить при создании ключей перевода.

Проверка согласованности

Проверка *Противоречия* срабатывает всякий раз, когда строки отличаются друг от друга. Вы можете использовать это, чтобы отрецензировать такие различия вручную и выбрать правильный перевод.

Автоматический перевод

Автоматический перевод одних компонентов на основе переводов в других может быть способом синхронизации переводов между компонентами. Вы можете либо запустить его вручную (смотрите раздел *Автоматический перевод*), либо заставить его автоматически запускаться при обновлении репозитория с помощью надстройки *Автоматический перевод*.

2.10 Лицензирование переводов

Вы можете указать, по каким лицензиям предоставляются переводы. Особенно оно важно, если переводы находятся в открытом доступе, так как оно помогает определить, для чего они могут быть использованы.

Вы должны указать информацию о лицензии *компонента*. Вам следует избегать требования заключения лицензионного соглашения с участниками, хотя это вполне возможно.

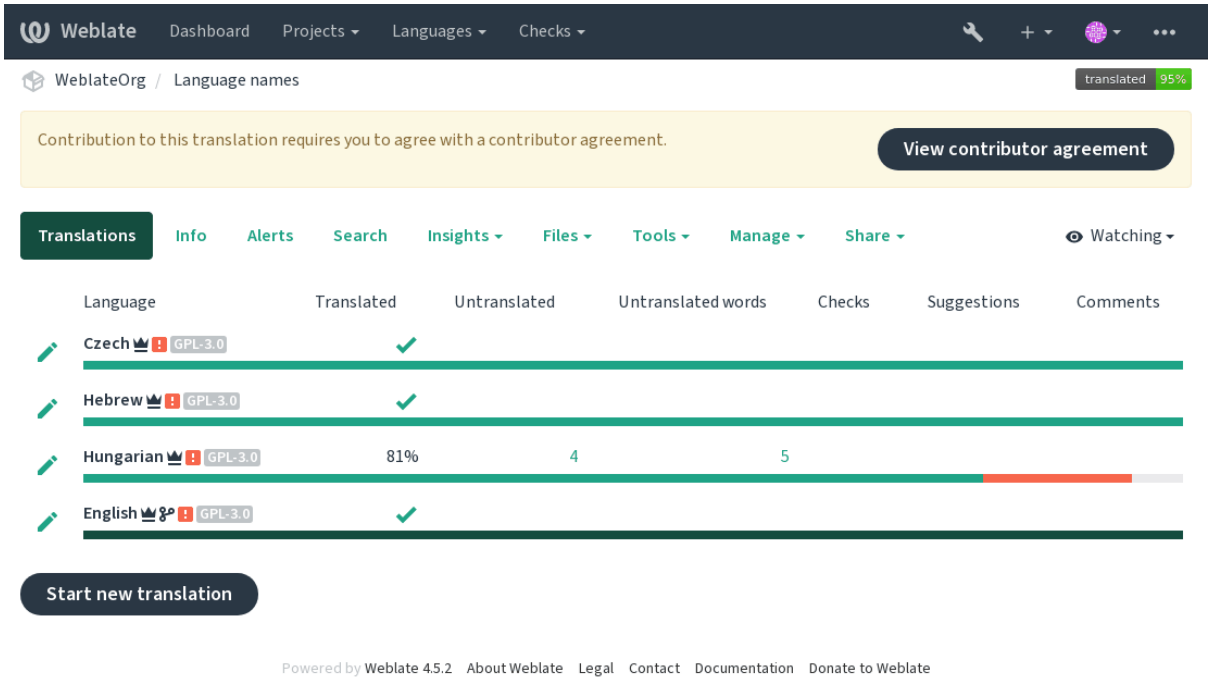
2.10.1 Информация о лицензии

После указания информации о лицензии (её названия и URL-адреса) эта информация отображается в разделе «Инфо» соответствующего *компонента*.

Обычно, если на перевод не требуется явного согласия, это лучшее место для размещения информации о лицензии. Если ваш проект или перевод не являются свободными, то, скорее всего, предварительное согласие вам необходимо.

2.10.2 Соглашение участника

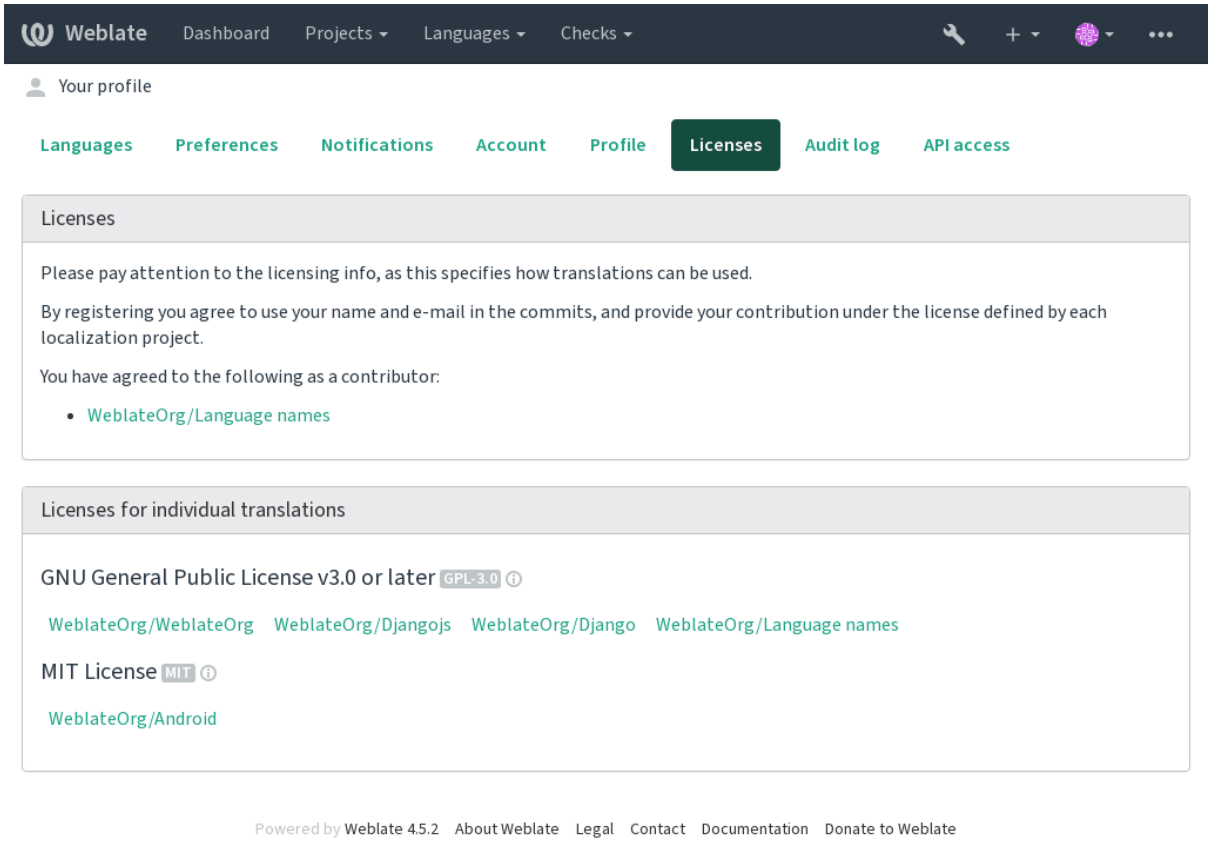
Если вы укажете лицензионное соглашение участника, то вносить свой вклад в перевод смогут только пользователи, с ним согласившиеся. Это хорошо заметный шаг, появляющийся при доступе к переводу:



Введённый текст разбивается на абзацы и может содержать внешние ссылки. Разметку HTML в нём использовать нельзя.

2.10.3 Пользовательские лицензии

Любой пользователь может просмотреть из своего профиля все лицензии на перевод всех публичных проектов на экземпляре:



2.11 Процесс перевода

2.11.1 Голосование по предложению

По умолчанию кто угодно может добавлять предложения, которые могут быть приняты зарегистрированными пользователями. Если вы хотите, чтобы для принятия предложения требовалось, чтобы с ним были согласны более одного зарегистрированного пользователя, то можно использовать систему голосования по предложениям, которое можно включить в *настройках компонента* параметром *Голосование по предложению* и настроить порог голосов, которые требуются для собственно принятия предложения в параметре *Автоматическое принятие предложений* (считается также голос пользователя, сделавшего это предложение).

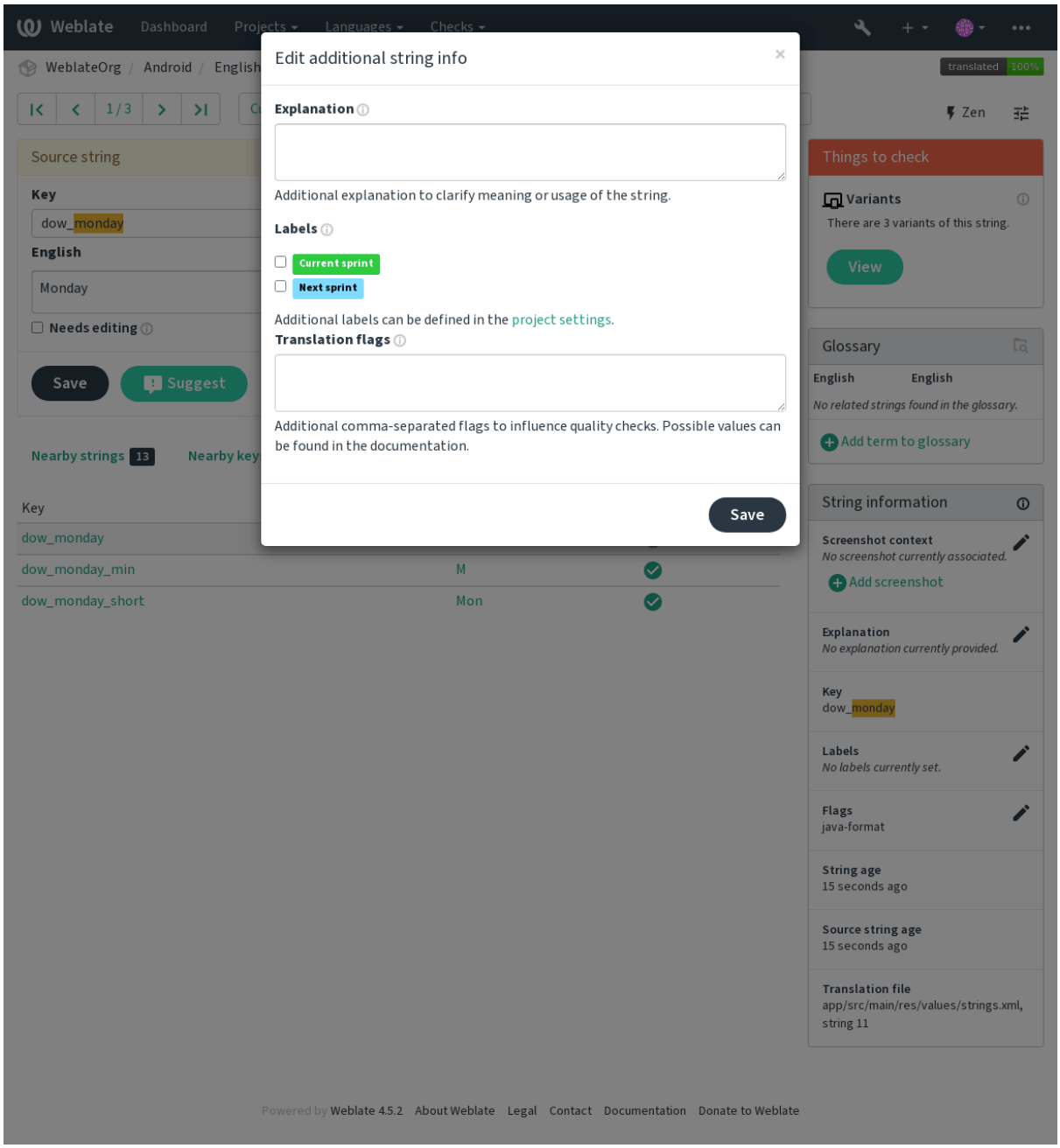
Примечание: Once automatic acceptance is set up, normal users lose the privilege to directly save translations or accept suggestions. This can be overridden with the *Edit string when suggestions are enforced* *permission*.

You can combine these with *access control* into one of the following setups:

- Пользователи предлагают и голосуют за предложения, а ограниченная группа контролирует то, что принимается. — Включите голосование. — Отключите автоматическое принятие. — Не позволяйте пользователям сохранять переводы.
- Пользователи предлагают и голосуют за предложения с автоматическим принятием предложения, как только с ним согласится определённое количество пользователей. — Включите голосование. — Установите желаемое количество голосов для автоматического принятия предложения.
- Необязательное голосование по предложениям (по желанию может использоваться пользователями, когда они не уверены в переводе, и делают вместо него несколько предложений). — Включите только голосование.

2.11.2 Дополнительная информация об исходных строках

Сделайте процесс перевода проще и улучшите его качество, добавив дополнительную информацию о строках, а именно пояснения, приоритеты строк, флаги проверок и визуальный контекст. Часть этой информации может быть извлечена непосредственно из файлов переводов, а часть можно добавить отредактировав дополнительную информацию о строках:



Доступ к нему можно получить непосредственно из интерфейса перевода, нажав на иконку правки у разделов *Контекст — снимок экрана* или *Флаги*.

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 Clone source NBS

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

Needs editing 169/1570 · 157

Save Suggest Skip

Nearby strings 26 Comments Automatic suggestions Other languages 4 History

Context	English	Czech	State
	Files	Soubory	
	Automatic translation	Automatický překlad	
	Add new translation string	Add new translation string	
	Translation status	Stav překladu	
	%(count)s word	%(count)s slovo	
	Other components	Další součásti	
	Translation file	Soubor s překladem	
	Download	Stáhnout	
	Browse all translation changes	Procházet všechny změny v překladu.	
	Automatic translation takes existing translations in this project and applies them to the current component. It can be used to push translations to a different branch, to fix inconsistent translations or to translate a new component using translation memory.	Automatický překlad použije stávající překlady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí překladové paměti.	
	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.	
	You can add new translation string here, it will automatically appear in all translations.	Zde můžete přidat nový řetězec k překladu, automaticky se objeví ve všech jazycích.	
	The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.	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 správně, protože fulltextový index pro tento překlad ještě není plně zpracován.	
	Review	Kontrola	
	Review translations touched by other users.	Zkontrolovat překlady od ostatních uživatelů.	
	Start review	Začít kontrolu	
	Percent	Procenta	
	Total	Celkem	
	Failing check	Neúspěšných kontrol	
	Last activity	Poslední 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
3 seconds ago

Source string age
3 seconds ago

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

Powered by Weblate 4.5.2 About Weblate Legal Contact Documentation Donate to Weblate

Приоритизация строк

Добавлено в версии 2.0.

Приоритет строк может быть изменён с помощью флага `priority`, чтобы более приоритетные строки предлагались для перевода раньше.

Подсказка: Это можно использовать для перевода строк в логическом порядке.

См.также:

Проверки качества

Флаги перевода

Добавлено в версии 2.4.

Изменено в версии 3.3: Ранее поле называлось *Отметки о проверке качества*, но теперь оно настраивает не только проверки.

Умолчательный набор флагов перевода определяется *переводимым компонентом* и файлом перевода. Тем не менее, вы можете использовать данное поле для настройки этих флагов для каждой исходной строки.

См.также:

Проверки качества

Пояснение

Изменено в версии 4.1: В предыдущих версиях это поле называлось *Дополнительный контекст*.

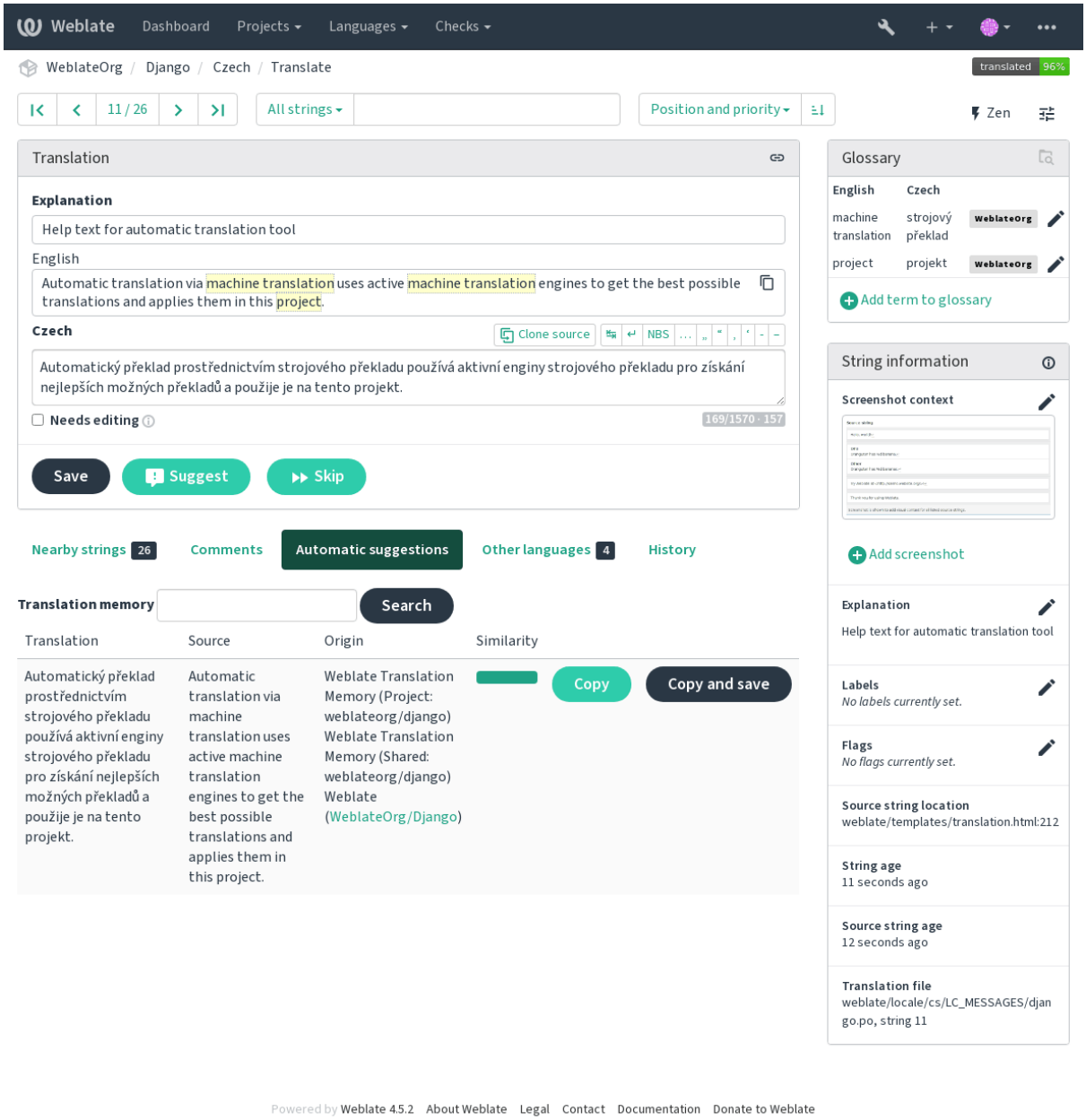
Используйте пояснения, чтобы уточненять сферу применения или детали использования строки. В поле поддерживается разметка Markdown, так что вы можете добавлять в него ссылки или другие элементы разметки текста.

Визуальный контекст для строк

Добавлено в версии 2.9.

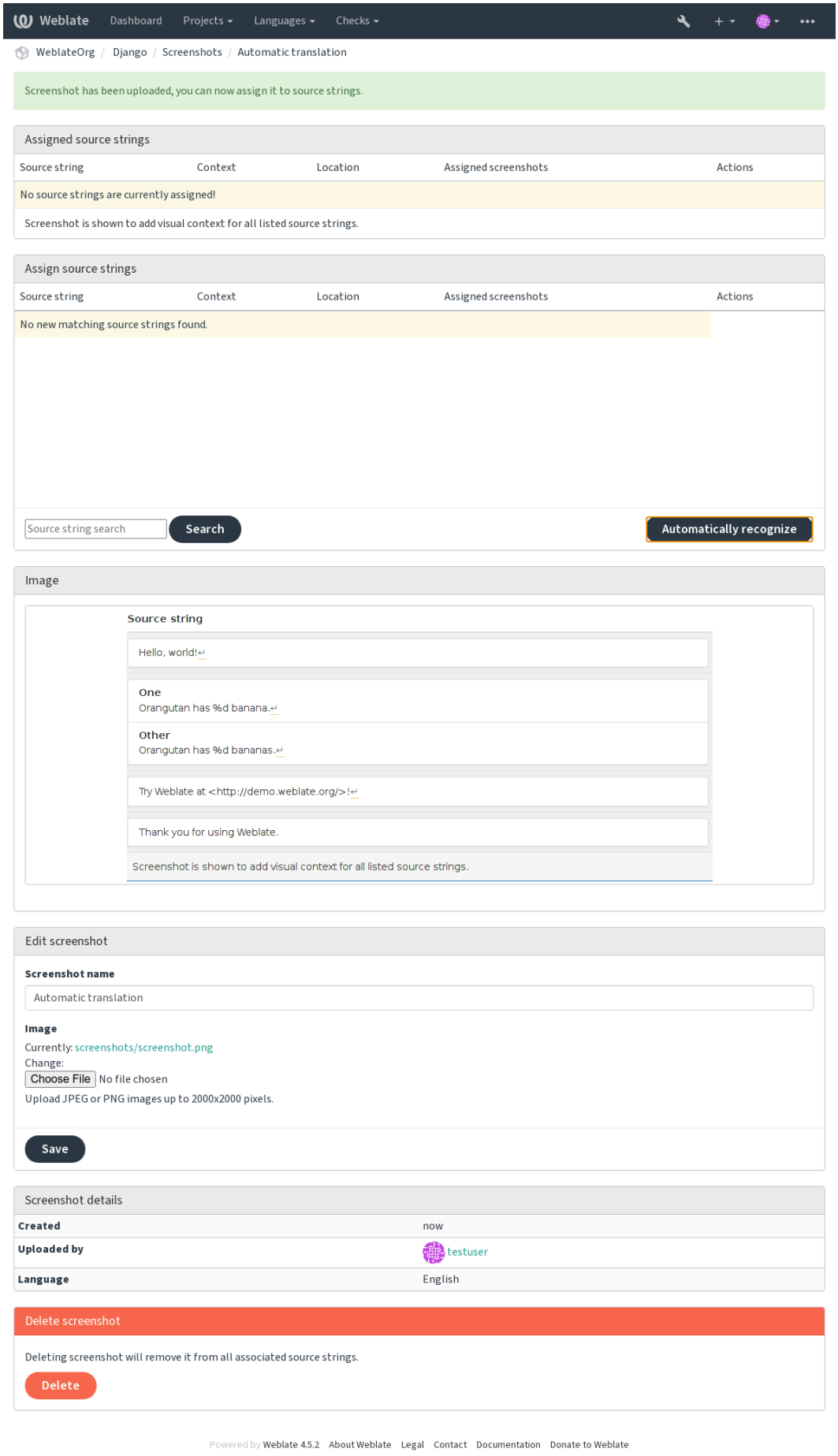
Вы можете загрузить снимок экрана, показывающий данную исходную строку в действии в вашей программе. Это поможет переводчикам понять, где она используется и как её следует переводить.

Загруженный снимок экрана отображается в боковой панели контекста перевода:



В добавок к окну «source-review», есть ещё один интерфейс управления снимками экрана в меню *Инструменты*. Загрузите снимки экрана и вручную назначьте, какие исходные строки им соответствуют, либо сделайте это в автоматическом режиме с помощью распознавания текста.

После загрузки снимка экрана через этот интерфейс можно управлять снимком и задать его связь с исходной строкой:


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
Source string
Context
Location
Assigned screenshots
Actions
No source strings are currently assigned!
Screenshot is shown to add visual context for all listed source strings.
Assign source strings
Source string
Context
Location
Assigned screenshots
Actions
No new matching source strings found.
Source string search
Search
Automatically recognize
Image
Source string
Hello, world!
One
Orangutan has %d banana.
Other
Orangutan has %d bananas.
Try Weblate at <http://demo.weblate.org/>!
Thank you for using Weblate.
Screenshot is shown to add visual context for all listed source strings.
Edit screenshot
Screenshot name
Automatic translation
Image
Currently: screenshots/screenshot.png
Change:
Choose File
No file chosen
Upload JPEG or PNG images up to 2000x2000 pixels.
Save
Screenshot details
Created
now
Uploaded by
testuser
Language
English
Delete screenshot
Deleting screenshot will remove it from all associated source strings.
Delete
Powered by Weblate 4.5.2
About Weblate
Legal
Contact
Documentation
Donate to Weblate

2.12 Проверки и исправления

2.12.1 Пользовательские автоматические исправления

Также в дополнение к стандартным вы можете реализовать своё собственное автоматическое исправление и включить его в список *AUTOFIX_LIST*.

Автоматические исправления — это очень мощный инструмент, но они могут повредить перевод; будьте осторожны при их написании.

Например, следующее автоматическое исправление заменит каждое появление строки `foo` в переводе на строку `bar`:

```
#
# Copyright © 2012 - 2021 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
```

Для установки пользовательских проверок необходимо в списке *AUTOFIX_LIST* указать полный путь к классу Python, смотрите раздел *Пользовательские проверки качества, надстройки и автоматические исправления*.

2.12.2 Настройка поведения с помощью флагов

Вы можете тонко настроить поведение Weblate'а (в основном проверки) для каждой исходной строки (в рецензировании исходных строк, смотрите раздел *Дополнительная информация об исходных строках*) или в *конфигурации компонента* (*Флаги перевода*). Некоторые форматы файлов также позволяют указывать флаги непосредственно в формате (смотрите раздел *Поддерживаемые форматы файлов*).

Флаги разделяются запятыми, параметры флагов разделяются двоеточием. Для включения в строку пробела или специальных символов вы можете использовать кавычки. Например:

```
placeholders:"special:value":"other value", regex:.*
```

Вот список принимаемых в настоящее время флагов:

rst-text Рассматривать текст как документ reStructuredText, влияет на проверку *Перевод без изменения*.

md-text Рассматривать текст как документ Markdown.

dos-eol Использовать маркеры конца строки DOS вместо маркеров Unix (комбинацию символов `\r\n` вместо символа `\n`).

url Строка должна состоять только из URL-адреса.

safe-html Строка должна быть безопасной для HTML, смотрите проверку *Небезопасный HTML-код*.

read-only Строка доступна только для чтения и не должна редактироваться в Weblate, смотрите проверку *Строки только для чтения*.

priority:N Приоритет строки. Строки с более высоким приоритетом предлагаются на перевод первыми. По умолчанию приоритет равен 100, чем выше приоритет строки, тем раньше она будет предложена для перевода.

max-length:N Ограничить максимальную длину строки N символами, смотрите проверку *Максимальная длина перевода*.

xml-text Рассматривать текст как документ XML, влияет на проверки *Синтаксис XML* и *XML-разметка*.

font-family:ШРИФТ Определяет семейство шрифтов для проверок отрисовки текста, смотрите раздел *Управление шрифтами*.

font-weight:НАСЫЩЕННОСТЬ Определяет насыщенность шрифта (свойство `font-weight`) для проверок отрисовки текста, смотрите раздел *Управление шрифтами*.

font-size:РАЗМЕР Определяет размер шрифта (свойство `font-size`) для проверок отрисовки текста, смотрите раздел *Управление шрифтами*.

font-spacing:ШИРИНА Определяет ширину межбуквенных интервалов (свойство `letter-spacing`) для проверок отрисовки текста, смотрите раздел *Управление шрифтами*.

placeholders:СТРОКА:СТРОКА2:... Ожидаемые в переводе строки-заполнители, смотрите проверку *Символы-заполнители*.

replacements:C:НА:C2:НА2... Замены, выполняемые при проверке параметров результирующего текста (например, в проверках *Максимальный размер перевода* или *Максимальная длина перевода*). Типовой случай использования этот флага — подстановка конкретных значений вместо заполнителей, чтобы гарантировать, что текст впишется в отведённые ему границы даже при использовании длинных строк, например: `replacements:%s:"Джон Доу"`.

variants:ИСТОЧНИК Добавить отметку, что данная строка является вариантом строки, указанной в качестве источника. Смотрите раздел «variants».

regex:РЕГУЛЯРНОЕ_ВЫРАЖЕНИЕ Регулярное выражение для сопоставления с переводом, смотрите раздел *Регулярное выражение*.

forbidden Используется для пометки переводов в глоссарии как нежелательных, см. *Нежелательные переводы*.

python-format, c-format, php-format, python-brace-format, javascript-format, c-sharp-format, ja
Рассматривать все строки как строки формата, влияет на следующие проверки: *Форматированные строки*, *Форматированные строки*, *Форматированные строки*, *Форматированные строки*, *Форматированные строки*, *Форматированные строки*, *Форматированные строки*, *Форматированные строки*, *Форматированные строки*, *Форматированные строки*, *Форматированные строки*, *Форматированные строки* и *Перевод без изменения*.

strict-same Сделать так, чтобы проверка «Перевод без изменения» не использовал встроенный чёрного список слов, см. *Перевод без изменения*.

check-glossary Включить проверку качества «Не соответствует глоссарию».

ignore-bbcode Пропускать проверку качества «BBcode-разметка».

ignore-duplicate Пропускать проверку качества «Повторяющиеся слова».

ignore-check-glossary Пропускать проверку качества «Не соответствует глоссарию».

ignore-double-space Пропускать проверку качества «Двойной пробел».

ignore-angularjs-format Пропускать проверку качества «Интерполяционная строка AngularJS».

ignore-c-format Пропускать проверку качества «C-формат».

ignore-c-sharp-format Пропускать проверку качества «C#-формат».

ignore-es-format Пропускать проверку качества «Шаблонные литералы ECMAScript».

ignore-i18next-interpolation Пропускать проверку качества «Интерполяция i18next».

ignore-java-format Пропускать проверку качества «Java-формат».

ignore-java-messageformat Пропускать проверку качества «Java MessageFormat».

ignore-javascript-format Пропускать проверку качества «JavaScript-формат».

ignore-lua-format Пропускать проверку качества «Lua-формат».

ignore-percent-placeholders Пропускать проверку качества «Символы-заполнители в процентной нотации».

ignore-perl-format Пропускать проверку качества «Perl-формат».

ignore-php-format Пропускать проверку качества «PHP-формат».

ignore-python-brace-format Пропускать проверку качества «Python-формат: фигурные скобки».

ignore-python-format Пропускать проверку качества «Python-формат».

ignore-qt-format Пропускать проверку качества «Qt-формат».

ignore-qt-plural-format Пропускает проверку качества «Формат множественного числа Qt».

ignore-ruby-format Пропускать проверку качества «Ruby-формат».

ignore-vue-format Пропускать проверку качества «Форматирование Vue I18n».

ignore-translated Пропускать проверку качества «Уже переведено».

ignore-inconsistent Пропускать проверку качества «Противоречия».

ignore-kashida Пропускать проверку качества «Используется кашида».

ignore-md-link Пропускать проверку качества «Ссылки Markdown».

ignore-md-reflink Пропускать проверку качества «Сноски Markdown».

ignore-md-syntax Пропускать проверку качества «Синтаксис Markdown».

ignore-max-length Пропускать проверку качества «Максимальная длина перевода».

ignore-max-size Пропускать проверку качества «Максимальный размер перевода».

ignore-escaped-newline Пропускать проверку качества «Несовпадающие n».

ignore-end-colon Пропускать проверку качества «Несовпадающие двоеточия».

ignore-end-ellipsis Пропускать проверку качества «Несовпадающие многоточия».

ignore-end-exclamation Пропускать проверку качества «Несовпадающие восклицательные знаки».

ignore-end-stop Пропускать проверку качества «Несовпадающие точки».

ignore-end-question Пропускать проверку качества «Несовпадающие вопросительные знаки».

ignore-end-semicolon Пропускать проверку качества «Несовпадающие точки с запятой».

ignore-newline-count Пропускать проверку качества «Несовпадающие переводы строк».

ignore-plurals Пропускать проверку качества «Нет форм множественного числа».

ignore-placeholders Пропускать проверку качества «Символы-заполнители».

ignore-punctuation-spacing Пропускать проверку качества «Пробелы и пунктуация».

ignore-regex Пропускать проверку качества «Регулярное выражение».

ignore-same-plurals Пропускать проверку качества «Одинаковые формы множественного числа».

ignore-begin-newline Пропускать проверку качества «Пустая строка в начале».

ignore-begin-space Пропускать проверку качества «Пробелы в начале».

ignore-end-newline Пропускать проверку качества «Пустая строка в конце».

ignore-end-space Пропускать проверку качества «Пробелы в конце».

ignore-same Пропускать проверку качества «Перевод без изменения».

ignore-safe-html Пропускать проверку качества «Небезопасный HTML-код».

ignore-url Пропускать проверку качества «URL-адреса».

ignore-xml-tags Пропускать проверку качества «XML-разметка».

ignore-xml-invalid Пропускать проверку качества «Синтаксис XML».

ignore-zero-width-space Пропускать проверку качества «Пробел нулевой ширины».

ignore-ellipsis Пропускать проверку качества «Многоточие».

ignore-long-untranslated Пропускать проверку качества «Давно не переводилась».

ignore-multiple-failures Пропускать проверку качества «Множество неудачных проверок».

ignore-unnamed-format Пропускать проверку качества «Несколько переменных без названий».

ignore-optional-plural Пропускать проверку качества «Формы единственного числа».

Примечание: В общем случае правило называется `ignore-*` для любой проверки, где вместо звёздочки подставляется её идентификатор, так что вы можете использовать такое правило даже для своих собственных проверок.

Эти флаги понимаются как в параметрах *компонента*, так и в параметрах исходной строки или в самом файле перевода (например, в файле GNU gettext).

2.12.3 Принудительные проверки

Добавлено в версии 3.11.

Вы можете настроить список проверок, которые нельзя проигнорировать, задав в *настройках компонента* параметр *Принудительные проверки*. Все перечисленные проверки не смогут быть проигнорированы из пользовательского интерфейса, а любая строка, для которой такая проверка сработала, помечается флагом *На правку* (смотрите раздел *Состояния перевода*).

2.12.4 Управление шрифтами

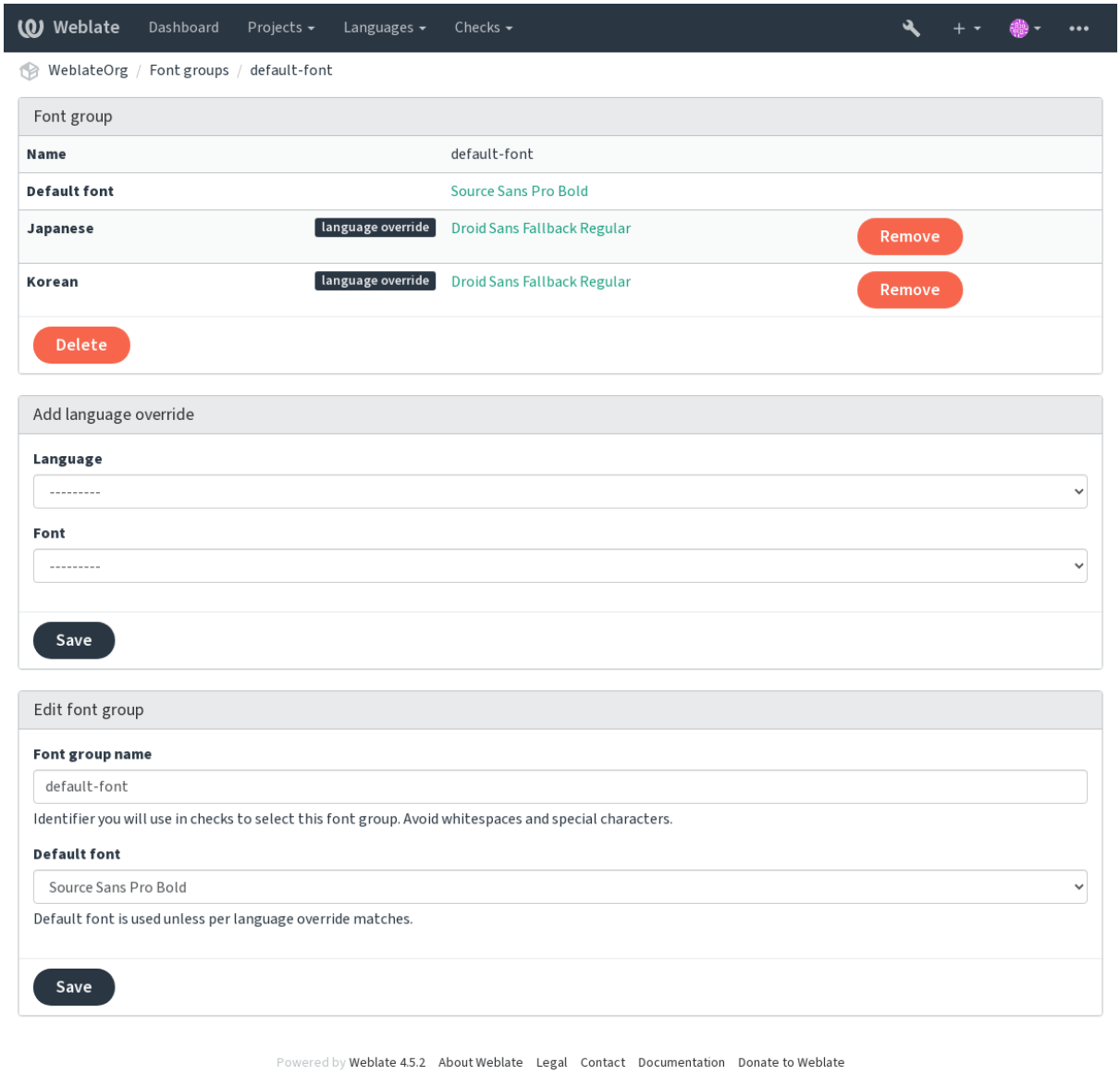
Добавлено в версии 3.7.

Подсказка: Загруженные в Weblate шрифты используются исключительно для проверки *максимального размера перевода*; на пользовательский интерфейс они никак не влияют.

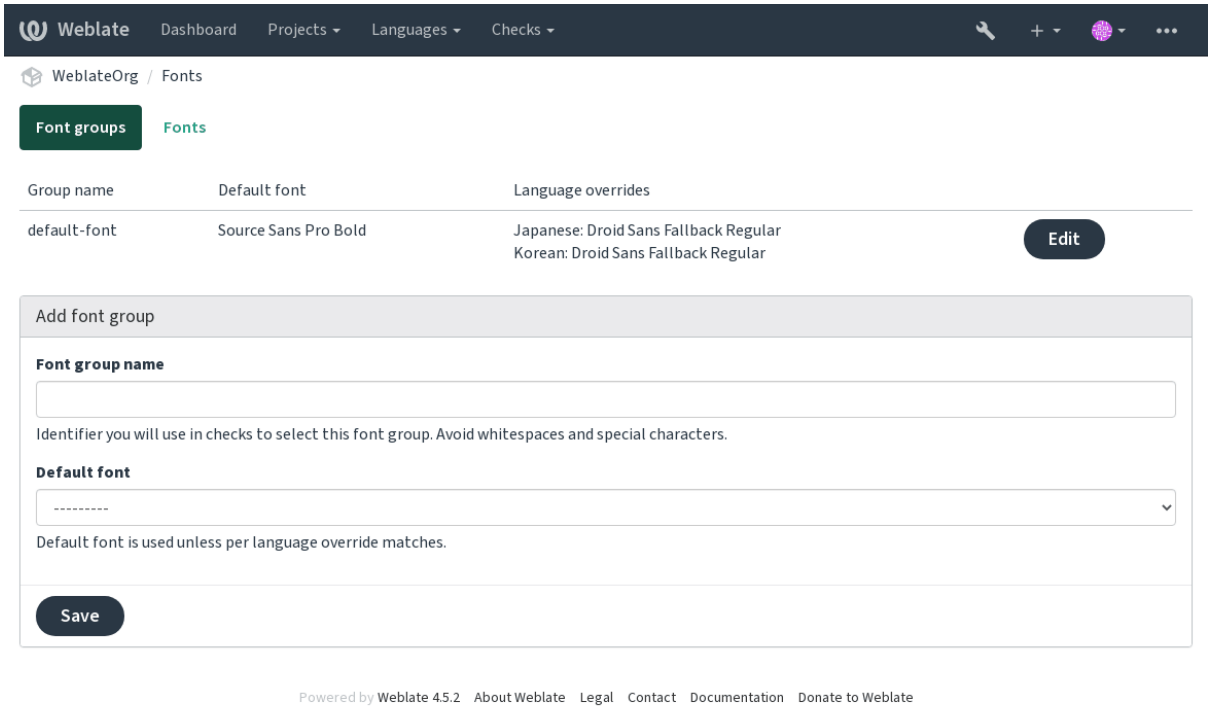
Проверке *Максимальный размер перевода*, которая вычисляет размер текста в пикселях после его отрисовки требуется, чтобы соответствующий шрифт был загружен в Weblate. Его также необходимо выбрать с помощью флагов перевода (см. *Настройка поведения с помощью флагов*).

Загрузить шрифты в Weblate и управлять ими можно в пункте *Шрифты* меню *Управлять* вашего проекта перевода. Создайте группы шрифтов и используйте их в проверках. Поддерживаются шрифты TrueType и OpenType.

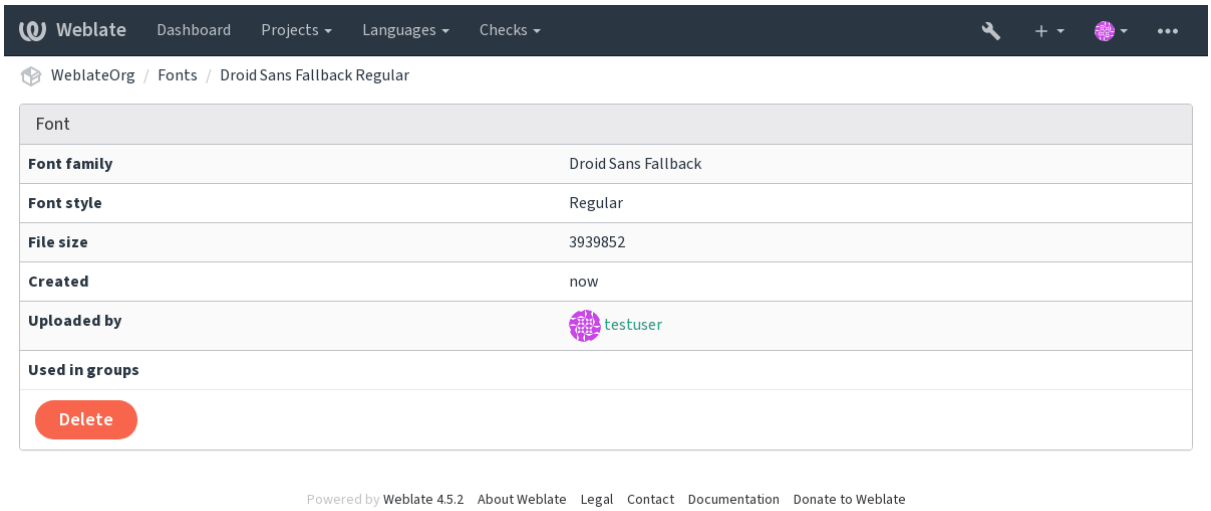
Группы шрифтов позволяют вам определить различные шрифты для разных языков, что обычно необходимо для языков, алфавит которых не использует латиницу:



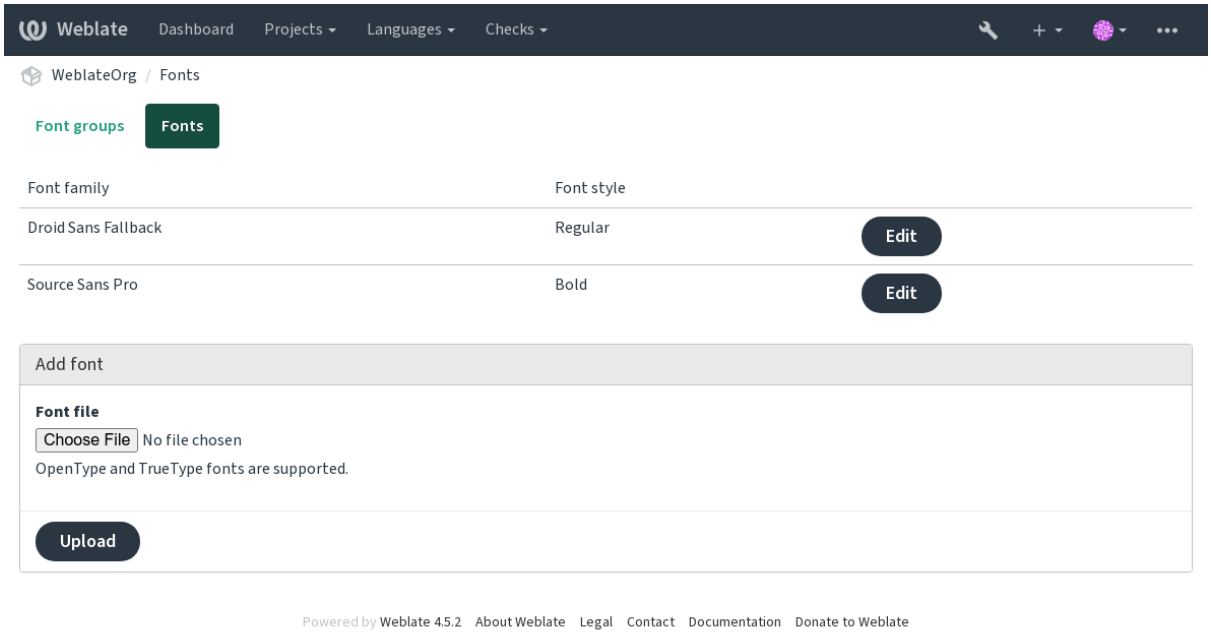
Группы шрифтов идентифицируются по имени, которое не может содержать пробелы или специальные символы, так что их можно легко использовать в определении проверки:



Семейство и начертание шрифта распознаются автоматически после его загрузки:



Вы можете загрузить в Weblate несколько шрифтов:



Чтобы использовать шрифты в проверке длины строки, передайте проверке соответствующие флаги (смотрите раздел *Настройка поведения с помощью флагов*). Скорее всего, вам понадобятся следующие из них:

`max-size:500` Определяет максимальную ширину.

`font-family:ubuntu` Определяет используемую группу шрифтов путём указания её идентификатора.

`font-size:22` Определяет размер шрифта.

2.12.5 Написание собственных проверок

Хотя в Weblate встроен широкий спектр проверок качества (смотрите раздел *Проверки качества*), они могут охватывать не всё, что вы хотите проверить. Список выполняемых проверок можно настроить с помощью параметра `CHECK_LIST`, кроме того, вы можете добавить собственные проверки.

1. Создайте подкласс *weblate.checks.Check*
2. Установите несколько атрибутов.
3. Реализуйте либо метод `check` (если вы хотите работать в своём коде с формами множественного числа), либо метод `check_single` (который сам выполняет такую обработку).

Несколько примеров:

Для установки пользовательских проверок необходимо в списке `CHECK_LIST` указать полный путь к классу Python, смотрите раздел *Пользовательские проверки качества, надстройки и автоматические исправления*.

Проверка, что текст перевода не содержит подстроки «foo»

Это довольно простая проверка, которая просто проверяет, не пропущена ли в переводе строка «foo».

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# (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
```

Проверка того, что в переводе на чешский язык формы множественного числа отличаются друг от друга

Проверка использует информацию о языке, чтобы проверить, что две формы множественного числа в чешском языке не являются одинаковыми.

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#
"""Quality check example for Czech plurals."""
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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 Машинный перевод

В Weblate встроена поддержка нескольких сервисов машинного перевода, которая может быть включена администратором с помощью добавления соответствующих значений в список *MT_SERVICES*. Эти сервисы идут с собственными условиями своего использования, поэтому убедитесь, что вам разрешено использовать их так, как вы хотите.

Исходный язык можно настроить в *настройках проекта*.

2.13.1 amaGama

Специальная установка *tmserver*, запущенная авторами Virtaal.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.tmserver.AmagamaTranslation`.

См.также:

Установка amaGama, Amagama, Память переводов amaGama

2.13.2 Apertium

Свободная платформа ПО машинного перевода, обеспечивающая переводы на ограниченный набор языков.

Рекомендуемый способ использования Apertium — запуск собственного сервера Apertium-APy.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.apertium.ApertiumAPYTranslation` и заданием параметра *MT_APERTIUM_APY*.

См.также:

MT_APERTIUM_APY, Веб-сайт Apertium, Документация Apertium APy

2.13.3 AWS

Добавлено в версии 3.1.

Amazon Translate — это сервис машинного перевода при помощи нейронных сетей для перевода текста на английский язык и с английского языка на широкое множество поддерживаемых языков.

1. Turn on this service by adding `weblate.machinery.aws.AWSTranslation` to *MT_SERVICES*.
2. Установите модуль *boto3*.
3. Настройте Weblate.

См.также:

MT_AWS_REGION, *MT_AWS_ACCESS_KEY_ID*, *MT_AWS_SECRET_ACCESS_KEY* Документация по Amazon Translate

2.13.4 Машинный перевод через API Baidu

Добавлено в версии 3.2.

Сервис машинного перевода, предоставляемый Baidu.

Этот сервис использует API, и чтобы его использовать, вам нужно получить от Baidu идентификатор и ключ API.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.baidu.BaiduTranslation` и заданием параметров *MT_BAIDU_ID* и *MT_BAIDU_SECRET*.

См.также:

MT_BAIDU_ID, *MT_BAIDU_SECRET* API переводчика Baidu

2.13.5 DeepL

Добавлено в версии 2.20.

DeepL — это платный сервис, предоставляющий хороший машинный перевод на несколько языков. Вам необходимо приобрести подписку на *DeepL API* или воспользоваться устаревшим планом *DeepL Pro (classic)*.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.deepl.DeepLTranslation` и заданием параметра *MT_DEEPL_KEY*.

Подсказка: В случае, если у вас есть подписка на CAT-инструменты, вы должны использовать «API v1» вместо «v2», которую по умолчанию использует Weblate (в данном случае это не совсем версия API). Вы можете изменить его с помощью параметра *MT_DEEPL_API_VERSION*.

См.также:

MT_DEEPL_KEY, *MT_DEEPL_API_VERSION*, Веб-сайт DeepL, Цены на DeepL, Документация по API DeepL

2.13.6 Glosbe

Бесплатный словарь и память переводов для почти всех живых языков.

API можно использовать бесплатно, но при условии наличия лицензии на используемый источник данных. Для борьбы со злоупотреблениями существует ограничение на количество вызовов API, которое можно выполнить с одного IP в течение установленного периода времени.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.glosbe.GlosbeTranslation`.

См.также:

Веб-сайт Glosbe

2.13.7 Google Translate

Сервис машинного перевода, предоставляемый Google.

Этот сервис использует API Переводчика Google, поэтому вам нужно получить ключ API и включить тарифный план в консоли Google API.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.google.GoogleTranslation` и заданием параметра *MT_GOOGLE_KEY*.

См.также:

MT_GOOGLE_KEY, Документация по переводчику Google

2.13.8 Google Translate API V3 (расширенный)

Сервис машинного перевода, предоставляемый облачными сервисами Google Cloud.

Этот сервис отличается от предыдущего способом аутентификации. Для включения сервиса добавьте в список *MT_SERVICES* значение `weblate.machinery.googlev3.GoogleV3Translation` и задайте параметры

- *MT_GOOGLE_CREDENTIALS*
- *MT_GOOGLE_PROJECT*

Если без *location* у вас не заработает, вам также может потребоваться указать параметр *MT_GOOGLE_LOCATION*.

См.также:

MT_GOOGLE_CREDENTIALS, *MT_GOOGLE_PROJECT*, *MT_GOOGLE_LOCATION* Документация по переводчику Google

2.13.9 Переводчик Microsoft Cognitive Services

Добавлено в версии 2.10.

Сервис машинного перевода, предоставляемый Microsoft на портале Azure в качестве одного из сервисов Cognitive Services.

Weblate реализует API переводчика версии V3.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.microsoft.MicrosoftCognitiveTranslation` и заданием параметра *MT_MICROSOFT_COGNITIVE_KEY*.

API переводчика текста версии V2

Ключ, который вы используете для API переводчика версии V2, можно использовать с API версии 3.

API переводчика текста версии V3

Вам необходимо зарегистрироваться на портале Azure и использовать полученный там ключ. При использовании новых ключей Azure вам также потребуется установить параметр *MT_MICROSOFT_REGION* в локаль вашего сервиса.

См.также:

MT_MICROSOFT_COGNITIVE_KEY, *MT_MICROSOFT_REGION*, Cognitive Services — API переводчика текста, Портал Microsoft Azure

2.13.10 Microsoft Terminology Service (Терминологическая служба Майкрософт)

Добавлено в версии 2.19.

API терминологической службы Майкрософт позволяет программно получать доступ к терминологии, определениям и строкам пользовательского интерфейса (UI), доступным на языковом портале, через веб-сервис.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.microsoftterminology.MicrosoftTerminologyService`.

См.также:

API терминологической службы Майкрософт

2.13.11 ModernMT

Добавлено в версии 4.2.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.modernmt.ModernMTTranslation` и заданием параметра *MT_MODERNMT_KEY*.

См.также:

ModernMT API, *MT_MODERNMT_KEY*, *MT_MODERNMT_URL*

2.13.12 MyMemory

Огромная память переводов с машинным переводом.

Бесплатное анонимное использование в настоящее время ограничено 100 запросами в день, или 1000 запросами в день при указании контактного адреса электронной почты в параметре *MT_MYMEMORY_EMAIL*. Также вы можете запросить у них увеличение лимитов.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.mymemory.MyMemoryTranslation` и заданием параметра *MT_MYMEMORY_EMAIL*.

См.также:

MT_MYMEMORY_EMAIL, *MT_MYMEMORY_USER*, *MT_MYMEMORY_KEY*, Веб-сайт MyMemory

2.13.13 API машинного перевода NetEase Sight

Добавлено в версии 3.3.

Сервис машинного перевода, предоставляемый Netease.

Этот сервис использует API, поэтому вам необходимо получить от NetEase ключ и секрет.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.youdao.NeteaseSightTranslation` и заданием параметров *MT_NETEASE_KEY* и *MT_NETEASE_SECRET*.

См.также:

MT_NETEASE_KEY, *MT_NETEASE_SECRET* Платформа перевода Netease Sight

2.13.14 tmserver

Вы можете запустить свой собственный сервер памяти переводов, используя сервер, сходящий в поставку с Translate-toolkit, и настроив общение Weblate'а с ним. Также вы можете использовать его с сервером amaGama, который является расширенной версией tmserver'а.

1. Сперва вам нужно импортировать в память переводов некоторые данные:

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.tmserver.TMServerTranslation`.

```
build_tmdb -d /var/lib/tm/db -s en -t cs locale/cs/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t de locale/de/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t fr locale/fr/LC_MESSAGES/django.po
```

3. Запустите tmserver, чтобы он слушал ваши запросы:

```
tmserver -d /var/lib/tm/db
```

4. Настройте Weblate для общения с ним:

```
MT_TMSERVER = "http://localhost:8888/tmserver/"
```

См.также:

MT_TMSERVER, tmserver Установка amaGama, Amagama, Память переводов Amagama

2.13.15 Yandex Translate (Яндекс.Переводчик)

Сервис машинного перевода, предоставляемый Яндексом.

Этот сервис использует API переводчика, поэтому вам необходимо получить от Яндекса ключ API.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.yandex.YandexTranslation` и заданием параметра *MT_YANDEX_KEY*.

См.также:

MT_YANDEX_KEY, API Яндекс.Переводчика, Яндекс.Переводчик

2.13.16 API машинного перевода Youdao Zhiyun

Добавлено в версии 3.2.

Сервис машинного перевода, предоставляемый Youdao.

Этот сервис использует API, поэтому вам необходимо получить от Youdao идентификатор и ключ API.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.youdao.YoudaoTranslation` и заданием параметров *MT_YOUDAO_ID* и *MT_YOUDAO_SECRET*.

См.также:

MT_YOUDAO_ID, *MT_YOUDAO_SECRET*, Сервис переводов Youdao Zhiyun с естественных языков

2.13.17 Weblate

Weblate также сам может быть источником машинных переводов. Этот перевод основан на движке полнотекстового поиска Woosh и выдаёт как точные, так и приблизительные соответствия.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.weblatetm.WeblateTranslation`.

2.13.18 Память переводов Weblate'а

Добавлено в версии 2.20.

Память переводов также может использоваться в качестве источника предложений машинного перевода.

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.machine.WeblateMemory`. Он включён по умолчанию.

2.13.19 SAP Translation Hub

Сервис машинного перевода, предоставляемый SAP.

Для использования этой службы вам необходимо иметь учётную запись SAP (и включить SAP Translation Hub в облачной платформе SAP).

Этот сервис включается добавлением в список *MT_SERVICES* значения `weblate.machinery.saptranslationhub.SAPTranslationHub` и заданием доступа либо к песочнице, либо к рабочему API.

Примечание: Для доступа к API песочницы необходимо задать параметры *MT_SAP_BASE_URL* и *MT_SAP_SANDBOX_APIKEY*.

Для доступа к рабочему API необходимо задать параметры *MT_SAP_BASE_URL*, *MT_SAP_USERNAME* и *MT_SAP_PASSWORD*.

См.также:

MT_SAP_BASE_URL, *MT_SAP_SANDBOX_APIKEY*, *MT_SAP_USERNAME*, *MT_SAP_PASSWORD*, *MT_SAP_USE_MT* SAP Translation Hub API

2.13.20 Пользовательский машинный перевод

Также вы можете реализовать собственные сервисы машинного перевода, написав всего несколько строк кода на Python. В этом примере реализован машинный перевод на фиксированный список языков через модуль Python `dictionary`:

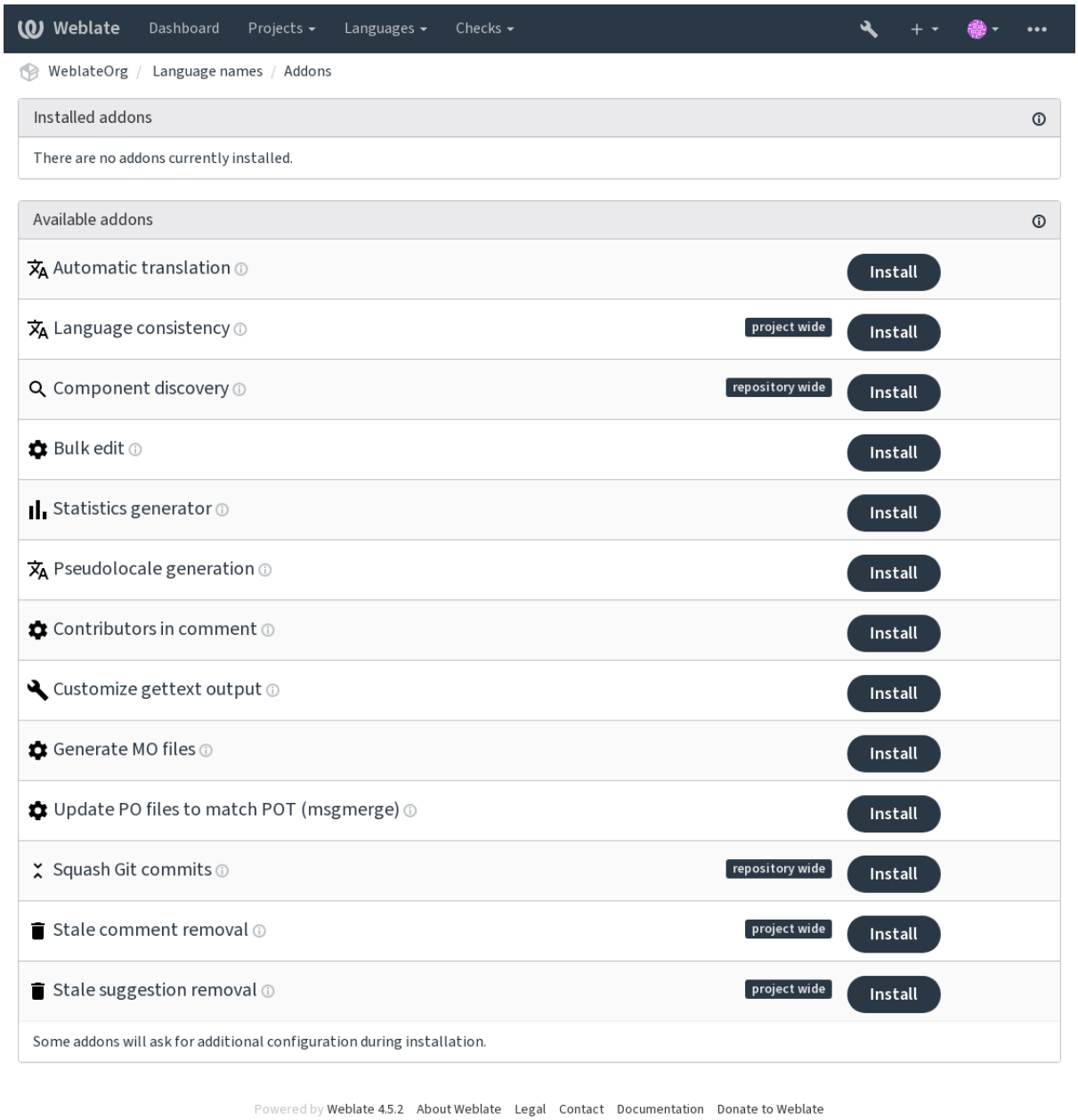
```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either 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}
```

Вы можете добавить свой класс в `MT_SERVICES` и Weblate начнёт его использовать.

2.14 Надстройки

Добавлено в версии 2.19.

Надстройки предоставляют способы изменения и автоматизации рабочего процесса перевода. Администратор может добавлять и управлять надстройками из меню *Управлять* ↓ *Надстройки* соответствующего компонента перевода.



2.14.1 Встроенные надстройки

Автоматический перевод

Добавлено в версии 3.9.

Автоматически переводит строки при помощи машинного перевода или других компонентов.

It is triggered:

- When new strings appear in a component.
- Once in a month for every component, this can be configured using `BACKGROUND_TASKS`.

См.также:

Автоматический перевод, *Поддержание единого перевода в разных компонентах*

CDN локализации JavaScript'a

Добавлено в версии 4.2.

Публикует переводы в сети доставки содержимого (CDN); используется при локализации JavaScript и HTML.

Его можно использовать для локализации статических HTML-страниц или для загрузки локализации в код JavaScript.

Создаёт для вашего компонента уникальный URL-адрес, который вы можете использовать на своих HTML-страницах для локализации. Подробности смотрите в разделе «weblate-cdn».

См.также:

cdn-addon-config, weblate-cdn, cdn-addon-extract, cdn-addon-html

Удаление пустых строк

Добавлено в версии 4.4.

Удаляет из файлов перевода строки без перевода.

Используйте эту надстройку, чтобы пустые строки не попадали в файлы перевода (например, в случае, если ваша библиотека локализации отображает их как отсутствующие вместо того, чтобы подставлять исходные).

См.также:

Обновляет ли Weblate помимо самих переводов также и файлы перевода?

Очистка файлов перевода

Обновляет все файлы переводов до соответствия одноязычному базовому файлу. Для большинства форматов файлов это означает удаление устаревших ключей переводов, которых больше нет в базовом файле.

См.также:

Обновляет ли Weblate помимо самих переводов также и файлы перевода?

Языковая согласованность

Гарантирует, что для всех компонентов в рамках одного проекта будут добавляться все языки перевода, которые есть хотя бы у одного проекта. Переводы на новые языки будут создаваться пустыми.

Отсутствующие языки проверяются каждые 24 часа и при добавлении нового языка через Weblate.

В отличие от большинства других надстроек, эта влияет на весь проект.

Подсказка: Автоматически переводите новые добавленные строки с помощью надстройки *Автоматический перевод*.

Обнаружение компонентов

Автоматически добавляет или удаляет компоненты проекта в соответствии с изменениями файлов в системе контроля версий.

Она срабатывает при каждом обновлении репозитория СКВ, а в остальном её работа аналогична команде управления `import_project`. Таким образом, внутри одной системы контроля версий вы можете отслеживать несколько компонентов перевода.

Сопоставление осуществляется с помощью регулярных выражений, что даёт возможность задать достаточно сложную конфигурацию, хотя это и требует некоторого количества начальных знаний. Для распространённых случаев в справочном разделе надстройки можно найти ряд примеров.

Когда вы нажмёте *Сохранить*, надстройка предварительно покажет найденные компоненты, так что вы сможете проверить, действительно ли такая конфигурация соответствует вашим потребностям:

Weblate Dashboard Projects Languages Checks

WeblateOrg / Language names / Addons / Component discovery

Configure addon

- Please review and confirm the matched components.

Component	Matched files
Following components would be created	
Djangojs	weblate/locale/hu/LC_MESSAGES/djangojs.po (hu) weblate/locale/he/LC_MESSAGES/djangojs.po (he) weblate/locale/cs/LC_MESSAGES/djangojs.po (cs)
Django	weblate/locale/cs/LC_MESSAGES/django.po (cs) weblate/locale/hu/LC_MESSAGES/django.po (hu) 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 filemask.

☑ **Clone addons from the main component to the newly created ones**

☐ **Remove components for inexistant files**

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description

`(?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 4.5.2 About Weblate Legal Contact Documentation Donate to Weblate

Подсказка: Надстройка «Обнаружение компонентов» использует ref:*внутренние адреса Weblate <internal-urls>*. Это удобный способ совместного использования системы контроля версий разными компонентами. Привязать дополнительные компоненты к основному так, чтобы они использовали его локальный репозиторий, можно, задав в качестве *репозитория исходного кода* адрес вида `weblate://проект/основной-компонент` (в разделе *Управление* ↓ *Параметры* ↓ *Система контроля версий*) для каждого такого компонента. Это сэкономит не только ваше время на настройку, но и системные ресурсы.

См.также:

Шаблонная разметка

Массовая правка

Добавлено в версии 3.11.

Позволяет массово править флаги, метки или состояния строк.

Для того чтобы начать автоматизацию расстановки меток, можно найти все строки по запросу `NOT has:label` и добавлять нужные метки до тех пор, пока не будет достигнут нужный эффект. Аналогично вы можете выполнять любые другие автоматические операции с метаданными Weblate.

Примеры:

Таблица 4: Автоматическое присвоение метки новым строкам

Поисковый запрос	`NOT has:label`
Добавляемые метки	*recent*

Таблица 5: Отмечать все списки изменений appstore доступными только для чтения

Поисковый запрос	`language:en AND key:changelogs/`
Добавляемые флаги перевода	`read-only`

См.также:

Массовая правка, *Настройка поведения с помощью флагов*, labels

Пометка неизменённых переводов флагом «На правку»

Добавлено в версии 3.1.

При импорте новой строки для перевода из системы контроля версий и при её совпадении с исходной строкой она помечается в Weblate флагом «На правку». Особенно это полезно для тех форматов файлов, которые для непереведённых строк хранят исходные строки вместо перевода.

Подсказка: Возможно, вы также захотите ужесточить проверку на наличие *изменений в переводе*, добавив `strict-same` во *флаги перевода*.

См.также:

Состояния перевода

Пометка новых исходных строк флагом «На правку»

При импорте новой исходной строки из системы контроля версий она помечается в Weblate флагом «На правку». Таким образом, вы можете легко фильтровать и редактировать исходные строки, добавленные разработчиками.

См.также:

Состояния перевода

Пометка новых переводов флагом «На правку»

При импорте строки для перевода из системы контроля версий она помечается в Weblate флагом «На правку». Таким образом, вы можете легко фильтровать и редактировать переводы, созданные разработчиками.

См.также:

Состояния перевода

Генератор статистики

Генерирует файл, содержащий детальную информацию о состоянии перевода.

И в качестве имени файла и в качестве его содержимого вы можете использовать шаблоны Django; подробное описание разметки шаблона смотрите в разделе «*Шаблонная разметка*».

К примеру, сгенерировать файл со сводкой по каждому переводу можно со следующими настройками:

Имя сгенерированного файла `locale/{{ language_code }}.json`

Содержимое

```
{
   "language": "{{ language_code }}",
   "strings": "{{ stats.all }}",
   "translated": "{{ stats.translated }}",
   "last_changed": "{{ stats.last_changed }}",
   "last_author": "{{ stats.last_author }}",
}
```

См.также:

Шаблонная разметка

Генератор псевдолокалей

Генерирует дополнительный перевод, автоматически добавляя префикс и суффикс к исходным строкам.

Псевдолокали могут быть полезны для поиска строк, которые небыли подготовлены для локализации должным образом. Это происходит с помощью автоматического изменения всех переводимых исходных строк, дабы те строки, которые не были изменены, было бы достаточно просто заметить при запуске приложения.

Также с помощью этого возможно отследить те строки, чьи локализованные версии могут не уместиться в отведённое под них место и ломать вёрстку.

Подсказка: Хотя вы и можете использовать настоящие языки для проверки, но для этой цели в Weblate уже есть специальные псевдолокали: *en_XA* и *ar_XB*.

Участники в комментарии

Обновляет комментарий в заголовке PO-файла, включая в него имена участников и годы их участия.

Заголовок PO-файла будет выглядеть примерно так:

```
# 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.
```

Обновление переменной ALL_LINGUAS в файле «configure»

Обновляет переменную ALL_LINGUAS в файлах `configure`, `configure.in` и `configure.ac` при добавлении нового перевода.

Настройка вывода gettext

Позволяет настроить поведение вывода gettext, например, заворот строк.

Она предлагает следующие варианты:

- Переносить строки на 77-м символе и на символе перевода строки
- Переносить строки только на символах перевода строки
- Не переносить строки

Примечание: По умолчанию gettext переносит строки на 77-м символе и на символе переноса строкию. С параметром `--no-wrap`, они будут переносится только на символе переноса строки.

Обновление файла LINGUAS

Обновляет файл LINGUAS при добавлении нового перевода.

Создание MO-файлов

Автоматически генерирует MO-файл для каждого изменённого PO-файла.

В соответствующем поле можно задать местоположение сгенерированного MO-файла. Формат этого поля — *Шаблонная разметка*.

Обновлять PO-файлы до соответствия POT (msgmerge)

Обновляет все PO-файлы (в соответствии с *масками файлов*) так, чтобы они соответствовали POT-файлу (заданному *шаблоном для новых переводов*) с помощью **msgmerge**.

Это происходит всякий раз, когда новые изменения извлекаются из вышестоящего репозитория. Большинство параметров командной строки msgmerge можно задать в настройках.

См.также:

Обновляет ли Weblate помимо самих переводов также и файлы перевода?

Уплотнение Git-коммитов

Уплотняет (squash) Git-коммиты перед отправкой изменений.

Git-коммиты могут быть уплотнены перед отправкой изменений в одном из следующих режимов:

Добавлено в версии 3.4.

- Все коммиты в один
- По языку
- По файлу

Добавлено в версии 3.5.

- По автору

Исходные сообщения коммита сохраняются, но авторство коммитов теряется, если только не выбран режим *По автору*, или сообщение коммита не настроено на включение в него его авторов.

Добавлено в версии 4.1.

Исходные сообщения коммитов могут быть переопределены пользовательским сообщением коммита.

Дополнительно из исходных сообщений коммитов могут быть удалены завершители (строки коммита вида `Co-authored-by: ...`) с последующим их присоединением к концу сообщения уплотнённого коммита. Также в этом режиме для каждого переводчика генерируется соответствующий завершитель `Co-authored-by:`.

Настройка вывода в JSON

Позволяет подправить поведение вывода в JSON, например, отступы или сортировку ключей.

Форматирование файла свойств Java

Сортирует файл свойств Java.

Удаление устаревших комментариев

Добавлено в версии 3.7.

Устанавливает сроки удаления комментариев.

Может быть полезна для удаления старых комментариев, которые могли потерять актуальность. Используйте её с осторожностью, поскольку то, что комментарий старый, не означает, что он потерял свою значимость.

Удаление устаревших предложений

Добавлено в версии 3.7.

Устанавливает сроки удаления предложений.

В сочетании с голосованием по предложениям (смотрите раздел *Коллегиальное рецензирование*) может быть очень полезна для удаления предложений, которые не получили достаточного количества положительных голосов в заданный промежуток времени.

Обновление файлов RESX

Добавлено в версии 3.9.

Обновляет все файлы перевода, приводя их в соответствие с одноязычным исходным базовым файлом. Неиспользуемые строки удаляются, а новые добавляются как копии исходных строк.

Подсказка: Если вы хотите только удалить устаревшие ключи перевода, используйте надстройку *Очистка файлов перевода*.

См.также:

Обновляет ли Weblate помимо самих переводов также и файлы перевода?

Настройка вывода в YAML

Добавлено в версии 3.10.2.

Позволяет подправить поведение вывода в YAML, например, длину строк и символы перевода строк.

2.14.2 Настройка списка надстроек

Список надстроек настраивается параметром `WEBLATE_ADDONS`. Для добавления ещё одной надстройки просто включите в этот параметр полное имя класса.

2.14.3 Создание надстройки

Также вы можете создать свою собственную надстройку: просто создайте подкласс `weblate.addons.base.BaseAddon`, чтобы определить метаданные надстройки, а затем реализуйте нужный обратный вызов, который будет выполнять обработку.

См.также:

Разработка надстроек

2.14.4 Выполнение скриптов из надстройки

Надстройки также могут использоваться для выполнения внешних скриптов. Раньше выполнение было интегрировано в сам Weblate, но теперь вам нужно написать код для заворачивания вашего скрипта в надстройку.

```
#
# Copyright © 2012 - 2021 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/>.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
#
"""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 addon executes a script.")

    # Script to execute
    script = "/bin/true"
    # File to add in commit (for pre commit event)
    # does not have to be set
    add_file = "po/{{ language_code }}.po"
```

Инструкции по установке смотрите в разделе *Пользовательские проверки качества, надстройки и автоматические исправления*.

Скрипт выполняется с текущим каталогом, установленном в корневой каталог репозитория системы контроля версий заданного компонента.

Кроме того, в скрипте доступны следующие переменные окружения:

WL_VCS
Используемая система контроля версий.

WL_REPO
URL-адрес вышестоящего репозитория.

WL_PATH
Абсолютный путь к репозиторию системы контроля версий.

WL_BRANCH
Добавлено в версии 2.11.

Ветка репозитория, настроенная для текущего компонента.

WL_FILEMASK
Маска файлов для текущего компонента.

WL_TEMPLATE
Имя файла шаблона для одноязычных переводов (может быть пустым).

WL_NEW_BASE
Добавлено в версии 2.14.

Имя файла, используемого для создания новых переводов (может быть пустым).

WL_FILE_FORMAT
Формат файла, используемый в текущем компоненте.

WL_LANGUAGE
Язык обрабатываемого в данный момент перевода (недоступен для обработчиков уровня компонента).

WL_PREVIOUS_HEAD
Предыдущая HEAD после обновления (доступна только после запуска обработчика «после обновления»).

WL_COMPONENT_SLUG
Добавлено в версии 3.9.

Плашка компонента, используемая для построения URL-адреса.

WL_PROJECT_SLUG
Добавлено в версии 3.9.

Плашка проекта, используемая для построения URL-адреса.

WL_COMPONENT_NAME
Добавлено в версии 3.9.

Название компонента.

WL_PROJECT_NAME
Добавлено в версии 3.9.

Название проекта.

WL_COMPONENT_URL
Добавлено в версии 3.9.

URL-адрес компонента.

WL_ENGAGE_URL
Добавлено в версии 3.9.

URL-адрес для присоединения к переводу проекта.

См.также:

Настройки компонента

Обработка репозитория после обновления

Может использоваться для обновления файлов перевода при изменении исходных файлов в вышестоящем репозитории. При этом помните, что Weblate видит только те файлы, которые были закоммичены в систему контроля версий, поэтому в скрипте вам необходимо закоммитить все изменения.

Например, при использовании Gulp это можно сделать следующим кодом:

```
#! /bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Обработка переводов перед коммитом

Используйте скрипт коммита для автоматического внесения изменений в какой-либо перевод перед их коммитом в репозиторий.

Перевод передается в виде единственного параметра, содержащего имя файла текущего перевода.

2.15 Память переводов

Добавлено в версии 2.20.

Weblate поставляется со встроенной памятью переводов, состоящей из следующих частей:

- Вручную импортированная память переводов (смотрите раздел *Пользовательский интерфейс*).
- Автоматически сохранённые переводы, выполненные в Weblate'е (зависят от *областей действия памяти переводов*).
- Автоматически импортированные прошлые переводы.

Содержимое памяти переводов может быть применено одним из двух способов:

- Вручную во время перевода, на вкладке *«Автоматизированный перевод»*.
- Автоматически, путём перевода строк с помощью инструмента *Автоматический перевод* или надстройки *Автоматический перевод*.

Подсказки по установке смотрите в *памяти переводов Weblate'а*, которая включена по умолчанию.

2.15.1 Области действия памяти переводов

Добавлено в версии 3.2: В более ранних версиях память переводов могла быть загружена только из файла, соответствующего области действия текущей импортированной памяти переводов.

Области действия памяти переводов позволяют обеспечить как конфиденциальность, так и совместное использование переводов в соответствии с желаемым поведением.

Импортированная память переводов

Импорт произвольных данных памяти переводов с помощью команды `import_memory` делает содержимое памяти переводов доступным для всех пользователей и проектов.

Память переводов пользователя

Автоматически сохраняет все переводы пользователя в персональной памяти переводов соответствующего пользователя.

Память переводов проекта

Все переводы в рамках проекта автоматически сохраняются в памяти переводов проекта, доступной только для этого проекта.

Совместная память переводов

Все переводы в проектах с включённой совместной памятью переводов сохраняются в совместной памяти переводов, доступной для всех проектов.

Пожалуйста, внимательно подумайте, следует ли включать эту функцию для совместно используемых установок Weblate, поскольку это может иметь серьёзные последствия:

- Переводы могут быть использованы кем угодно.
- Это может привести к разглашению секретной информации.

2.15.2 Управление памятью переводов

Пользовательский интерфейс

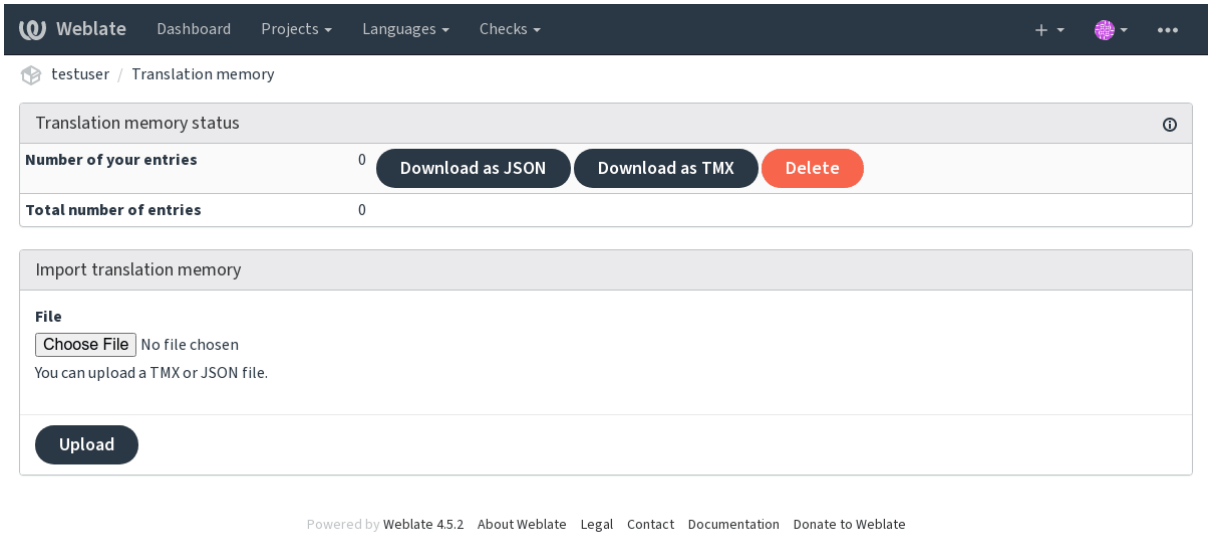
Добавлено в версии 3.2.

В базовом пользовательском интерфейсе вы можете управлять памятью переводов пользователя и проекта. Через него вы можете загружать, стирать или импортировать память переводов.

Подсказка: В Weblate может быть импортирована память переводов в формате JSON, формат TMX предусмотрен для взаимодействия с другими инструментами.

См.также:

Схема памяти переводов Weblate'а



Интерфейс управления

Для управления содержимым памяти переводов существует несколько команд управления. Они работают с памятью переводов целиком, не применяя фильтр по областям (если только этот фильтр не запрошен параметрами):

dump_memory Экспортирует память переводов в виде JSON'а

import_memory Импортирует файлы TMX или JSON в память переводов

2.16 Конфигурация

Все настройки хранятся в файле `settings.py` (как обычно для проектов Django).

Примечание: После изменения любого из этих параметров вам необходимо перезапустить Weblate — как процесс WSGI, так и процесс Celery.

В случае, если Weblate работает через `mod_wsgi`, для перезагрузки конфигурации вам нужно перезапустить Apache.

См.также:

Также, пожалуйста, ознакомьтесь с документацией Django по настройке параметров самого Django.

2.16.1 AKISMET_API_KEY

Weblate может использовать Akismet для проверки входящих анонимных предложений на спам. Посетите akismet.com, чтобы приобрести ключ API и связать его с сайтом.

2.16.2 ANONYMOUS_USER_NAME

Имена пользователей, которые не входят в систему.

См.также:

Управление доступом

2.16.3 AUDITLOG_EXPIRY

Добавлено в версии 3.6.

Сколько дней Weblate должен хранить журналы аудита, в которых содержится информация о деятельности учётной записи.

По умолчанию установлен в 180 дней.

2.16.4 AUTH_LOCK_ATTEMPTS

Добавлено в версии 2.14.

Максимальное количество неудачных попыток аутентификации до применения ограничения частоты запросов.

В настоящее время оно применяется в следующих местах:

- Страница входа. Удаляет пароль учётной записи, не позволяя пользователю войти в систему без запроса нового пароля.
- Страница сброса пароля. Предотвращает отправку новых писем, избегая рассылки спама пользователям со слишком большим количеством попыток сброса пароля.

По умолчанию установлен в 10.

См.также:

Ограничение частоты запросов,

2.16.5 AUTO_UPDATE

Добавлено в версии 3.2.

Изменено в версии 3.11: Первоначальные возможные значения параметра включено/выключено были изменены для дифференциации принимаемых строк.

Обновление всех репозиториев на ежедневной основе.

Подсказка: Полезно, если вы не используете *обработчики* для автоматического обновления Weblate'овских репозиториев.

Примечание: Варианты включено/выключено в дополнение к строкам существуют для обратной совместимости.

Варианты:

"none" Никаких ежедневных обновлений.

"remote" или False Только вытягиваются изменения из удалённых репозиториев.

"full" или True Вытягиваются изменения из удалённых репозиториев и сливаются с рабочей копией.

Примечание: Для включения этого параметра необходима работа *фоновых задач Celery*, также он вступит в силу после их перезапуска.

2.16.6 AVATAR_URL_PREFIX

Префикс для генерирования URL-адресов аватаров в выражении `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. Сервисы аватаров работают следующим образом:

Gravatar (по умолчанию), согласно https://gravatar.com/ `AVATAR_URL_PREFIX = 'https://www.gravatar.com/'`

Libravatar, согласно https://www.libravatar.org/ `AVATAR_URL_PREFIX = 'https://www.libravatar.org/'`

См.также:

Кеширование аватаров, `ENABLE_AVATARS`, *Аватары*

2.16.7 AUTH_TOKEN_VALID

Добавлено в версии 2.14.

Сколько времени действует токен аутентификации и временный пароль из писем сброса пароля. Устанавливается в количество секунд, по умолчанию равен 172800 (2 дня).

2.16.8 AUTH_PASSWORD_DAYS

Добавлено в версии 2.15.

Сколько дней должно быть разрешено использовать один и тот же пароль.

Примечание: Смена паролей, сделанная в версиях Weblate до версии 2.15, в этой политике учтена не будет.

По умолчанию установлен в 180 дней.

2.16.9 AUTOFIX_LIST

Список автоматических исправлений, применяемых при сохранении строки.

Примечание: Предоставьте полный путь к классу Python, реализующему интерфейс автоматического исправления.

Доступные исправления:

weblate.trans.autofixes.whitespace.SameBookendingWhitespace Выравнивает количество начальных и конечных пробельных символов в переводе с исходной строкой.

weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis Заменяет конечные точки (…) на многоточие (…), если оно есть в исходной строке.

weblate.trans.autofixes.chars.RemoveZeroSpace Удаляет пробельные символы нулевой ширины, если исходная строка их не содержит.

weblate.trans.autofixes.chars.RemoveControlChars Удаляет управляющие символы, если исходная строка их не содержит.

weblate.trans.autofixes.html.BleachHTML Удаляет небезопасную HTML-разметку из строк, помеченных флагом `safe-html` (смотрите проверку *Небезопасный HTML-код*).

Вы можете выбрать, какие исправления использовать:

```
AUTOFIX_LIST = (
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
)
```

См.также:

Автоматические исправления, *Пользовательские автоматические исправления*

2.16.10 BACKGROUND_TASKS

Добавлено в версии 4.5.2.

Defines how often lengthy maintenance tasks should be triggered for a component.

Right now this controls:

- *Автоматический перевод* addon
- *Проверки и исправления* recalculation

Possible choices:

- `monthly` (this is the default)
- `weekly`
- `daily`
- `never`

Примечание: Increasing the frequency is not recommended when Weblate contains thousands of components.

2.16.11 BASE_DIR

Базовый каталог, в котором находятся исходники Weblate'а. Используется для построения нескольких других умолчательных путей:

- *DATA_DIR*

Значение по умолчанию: каталог верхнего уровня исходников Weblate'а.

2.16.12 BASIC_LANGUAGES

Добавлено в версии 4.4.

Список языков, который будет предлагаться пользователям при создании нового перевода. Если не задан, то будет использоваться встроенный список, который включает все распространённые языки, но без добавления вариантов для конкретных стран.

Это не даёт непривилегированным пользователям добавлять нежелательные языки. Это не влияет на администраторов проектов — они всё также могут выбирать из полного списка языков, доступных в Weblate.

Примечание: Это также не определяет новые языки в Weblate, а только фильтрует те, что уже присутствуют в базе данных.

Пример:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

См.также:

Определение языков

2.16.13 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Настраивает заголовок `Content-Security-Policy` для Weblate'а. Заголовок автоматически генерируется на основе включённых интеграций со сторонними сервисами (Matomo, Google Analytics, Sentry, …).

Все они по умолчанию установлены в пустой список.

Пример:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

См.также:

Политика безопасности содержимого, Политика безопасности содержимого (Content Security Policy — CSP)

2.16.14 CHECK_LIST

Список выполняемых для перевода проверок качества.

Примечание: Предоставьте полный путь к классу Python, реализующему интерфейс проверки.

Подправьте список проверок так, чтобы включить в него только нужные вам.

По умолчанию включены все встроенные *проверки*, оттуда вы можете изменить эти настройки. В *примере файла настроек* они по умолчанию закомментированы, так что используются значения по умолчанию. В каждой новой версии Weblate'а туда добавляются новые проверки, если таковые появляются.

Вы можете отключить все проверки:

```
CHECK_LIST = ()
```

Можете включить только некоторые:

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

Примечание: Изменение этого параметра влияет только на недавно изменённые переводы, результаты существующих проверок будут по-прежнему сохранены в базе данных. Чтобы также применить изменения к сохранённым переводам, запустите команду `updatechecks`.

См.также:

Проверки качества, *Настройка поведения с помощью флагов*

2.16.15 COMMENT_CLEANUP_DAYS

Добавлено в версии 3.6.

Удаление комментариев через заданное количество дней. По умолчанию `None`, что означает, что комментарии не удаляются.

2.16.16 COMMIT_PENDING_HOURS

Добавлено в версии 2.10.

Количество часов до коммита ожидающих изменений фоновой задачей.

См.также:

Настройки компонента, *Возраст изменений для коммита*, *Выполнение задач технического обслуживания*, `commit_pending`

2.16.17 DATA_DIR

Каталог, в котором Weblate хранит все свои данные. Он содержит ссылки на репозитории систем контроля версий, индекс для полнотекстового поиска и различные файлы настроек для внешних инструментов.

Обычно существуют следующие подкаталоги:

home Домашний каталог, используемый при вызове скриптов.

ssh Ключи и конфигурация SSH.

static Default location for static Django files, specified by `STATIC_ROOT`. See *Обслуживание статических файлов*.

media Default location for Django media files, specified by `MEDIA_ROOT`. Contains uploaded screenshots.

vcs Version control repositories for translations.

backups Данные ежедневных резервных копий, для получения подробной информации обратитесь к разделу *Дампы данных для резервного копирования*.

celery Celery scheduler data, see *Фоновые задачи с использованием Celery*.

fonts: User-uploaded fonts, see *Управление шрифтами*.

Примечание: Этот каталог должен быть доступен Weblate'у для записи. Запуск его в качестве приложения uWSGI означает, что пользователь `www-data` должен иметь к нему доступ с правами на запись.

Простейший способ этого достичь — сделать пользователя владельцем каталога:

```
sudo chown www-data:www-data -R $DATA_DIR
```

По умолчанию установлен в `$BASE_DIR/data`.

См.также:

`BASE_DIR`, *Права доступа к файлам*, *Резервное копирование и перенос Weblate*

2.16.18 DATABASE_BACKUP

Добавлено в версии 3.1.

Должны ли резервные копии базы данных храниться в виде простого текста, сжатого файла или не храниться вовсе. Допустимые значения:

- `"plain"`
- `"compressed"`
- `"none"`

См.также:

Резервное копирование и перенос Weblate

2.16.19 DEFAULT_ACCESS_CONTROL

Добавлено в версии 3.3.

Умолчательное значение параметра управления доступом для новых проектов:

0 *Публичный*

1 *Защищённый*

100 *Частный*

200 *Настраиваемый*

Используйте *Настраиваемый*, если списками контроля доступа вы управляете вручную, что означает, что вы не полагаетесь на внутреннее Weblate'овское управление ими.

См.также:

Управление доступом к проекту, *Управление доступом*

2.16.20 DEFAULT_AUTO_WATCH

Добавлено в версии 4.5.

Настраивает, должен ли параметр *Автоматически добавлять проект в отслеживаемые при правке* быть включён по умолчанию для новых пользователей. Значение по умолчанию: `True`.

См.также:

Уведомления

2.16.21 DEFAULT_RESTRICTED_COMPONENT

Добавлено в версии 4.1.

Значение по умолчанию для ограничения компонента.

См.также:

Ограниченный доступ, *Scope of groups*

2.16.22 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DEFAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DEFAULT_MERGE_MESSAGE

Умолчательные сообщения коммита для различных операций, для получения подробной информации обратитесь к разделу *Настройки компонента*.

См.также:

Шаблонная разметка, *Настройки компонента*, *Сообщения коммита, добавления, удаления, слияния и сообщения надстроек*

2.16.23 DEFAULT_ADDONS

Умолчательные надстройки, устанавливаемые в каждый созданный компонент.

Примечание: Этот параметр влияет только на новые создаваемые компоненты.

Пример:

```
DEFAULT_ADDONS = {
    # Addon with no parameters
    "weblate.flags.target_edit": {},
    # Addon with parameters
    "weblate.autotranslate.autotranslate": {
        "mode": "suggest",
        "filter_type": "todo",
        "auto_source": "mt",
        "component": "",
        "engines": ["weblate-translation-memory"],
        "threshold": "80",
    },
}
```

См.также:

`install_addon`, `WEBLATE_ADDONS`

2.16.24 DEFAULT_COMMITER_EMAIL

Добавлено в версии 2.4.

Committer e-mail address defaulting to `noreply@weblate.org`.

См.также:

DEFAULT_COMMITER_NAME

2.16.25 DEFAULT_COMMITER_NAME

Добавлено в версии 2.4.

Committer name defaulting to `Weblate`.

См.также:

DEFAULT_COMMITER_EMAIL

2.16.26 DEFAULT_LANGUAGE

Добавлено в версии 4.3.2.

Язык по умолчанию, который будет использоваться, например как *исходный язык*.

По умолчанию — *en*. Соответствующая запись для данного языка должна присутствовать в базе данных.

См.также:

Определение языков, *Исходный язык*

2.16.27 DEFAULT_MERGE_STYLE

Добавлено в версии 3.4.

Стиль слияния для любых новых компонентов.

- *rebase* — по умолчанию
- *merge*

См.также:

Настройки компонента, *Стиль слияния*

2.16.28 DEFAULT_SHARED_TM

Добавлено в версии 3.2.

Configures default value of *Использовать совместную память переводов* and *Вносить вклад в совместную память переводов*.

2.16.29 DEFAULT_TRANSLATION_PROPAGATION

Добавлено в версии 2.5.

Значение по умолчанию для настройки распространения перевода, по умолчанию установлен в `True`.

См.также:

Настройки компонента, *Разрешить распространение перевода*

2.16.30 DEFAULT_PULL_MESSAGE

Заголовок для новых запросов на извлечение, по умолчанию равен `'Update from Weblate'`.

2.16.31 ENABLE_AVATARS

Включить ли для пользователей Gravatar-аватары. По умолчанию они включены.

Аватары извлекаются и кэшируются на сервере, что снижает риск утечки персональной информации и ускоряет взаимодействие с пользователем.

См.также:

Кеширование аватаров, `AVATAR_URL_PREFIX`, *Аватары*

2.16.32 ENABLE_HOOKS

Включить ли анонимные удалённые обработчики.

См.также:

Обработчики уведомлений

2.16.33 ENABLE_HTTPS

Отправлять ли ссылки на Weblate по протоколу HTTPS или HTTP. Этот параметр влияет на отправленные сообщения электронной почты и сгенерированные абсолютные URL-адреса.

В конфигурации по умолчанию он также используется для нескольких связанных с HTTPS параметров Django — он включает безопасные куки, переключает HSTS или включает перенаправление на HTTPS-адрес.

С перенаправлением на HTTPS в некоторых случаях могут быть проблемы: например, если вы используете обратный прокси, который выполняет SSL-терминацию из-за чего заголовки протокола не передаются Django в корректном виде, то вы можете столкнуться с бесконечным перенаправлением. В таком случае подкрутите настройке своего обратного прокси, дабы он добавлял заголовки `X-Forwarded-Proto` или `Forwarded`, или настройте `SECURE_PROXY_SSL_HEADER`, чтобы Django мог корректно определять статус SSL.

См.также:

`SESSION_COOKIE_SECURE`, `CSRF_COOKIE_SECURE`, `SECURE_SSL_REDIRECT`, `SECURE_PROXY_SSL_HEADER` *Установка правильного домена сайта*

2.16.34 ENABLE_SHARING

Включает/выключает меню *Поделиться*, с помощью которого пользователи могут делиться прогрессом перевода в социальных сетях.

2.16.35 GET_HELP_URL

Добавлено в версии 4.5.2.

URL where support for your Weblate instance can be found.

2.16.36 GITLAB_CREDENTIALS

Добавлено в версии 4.3.

Список учётных данных для серверов GitLab'а.

Подсказка: Используйте этот параметр для того, чтобы Weblate мог взаимодействовать с одной машины с несколькими разными серверами GitLab, а не только с одним, как при использовании `GITLAB_USERNAME` и `GITLAB_TOKEN`.

```
GITLAB_CREDENTIALS = {
    "gitlab.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitlab.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.37 GITLAB_USERNAME

Имя пользователя GitLab, используемое для отправки запросов на слияние для обновления перевода.

См.также:

`GITLAB_CREDENTIALS`, *GitLab*

2.16.38 GITLAB_TOKEN

Добавлено в версии 4.3.

GitLab'овский персональный токен доступа, используемый для вызовов API, необходимых для обновления перевода.

См.также:

`GITLAB_CREDENTIALS`, *GitLab*, GitLab: Персональный токен доступа

2.16.39 GITHUB_CREDENTIALS

Добавлено в версии 4.3.

Список учётных данных для серверов GitHub'а.

Подсказка: Используйте этот параметр для того, чтобы Weblate мог взаимодействовать с одной машины с несколькими разными серверами GitHub, а не только с одним, как при использовании *GITHUB_USERNAME* и *GITHUB_TOKEN*.

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "github.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.40 GITHUB_USERNAME

Имя пользователя GitHub, используемое для отправки запросов на извлечение для обновления перевода.

См.также:

GITHUB_CREDENTIALS, *GitHub*

2.16.41 GITHUB_TOKEN

Добавлено в версии 4.3.

GitHub'овский персональный токен доступа, используемый для вызовов API, необходимых для отправки запросов на извлечение для обновления перевода.

См.также:

GITHUB_CREDENTIALS, *GitHub*, Создание персонального токена доступа

2.16.42 GOOGLE_ANALYTICS_ID

Идентификатор аналитики Google для включения мониторинга Weblate'а с помощью Google Analytics.

2.16.43 HIDE_REPO_CREDENTIALS

Скрыть учётные данные репозитория в веб-интерфейсе. В случае, если в URL репозитория содержится пользователь и пароль, Weblate спрячет их, когда соответствующая информация будет показываться пользователям.

Например, вместо `https://user:password@git.example.com/repo.git` будет показано просто `https://git.example.com/repo.git`. Аналогичным образом он пытается очистить и сообщения об ошибках системы контроля версий.

Примечание: По умолчанию этот параметр включён.

2.16.44 HIDE_VERSION

Добавлено в версии 4.3.1.

Прячет информацию о версии от неаутентифицированных пользователей. Это также перенаправляет все ссылки на документацию последней версии вместо той, которая на самом деле установлена.

Скрытие версии — рекомендованная мера безопасности в некоторых корпорациях, хотя это и не предотвращает того, что атакующий сможет выяснить версию зондируя поведение системы.

Примечание: По умолчанию этот параметр выключен.

2.16.45 IP_BEHIND_REVERSE_PROXY

Добавлено в версии 2.14.

Указывает, работает ли Weblate за обратным прокси.

Если установлен в ``True`, то Weblate получает IP адрес из заголовка, определённого параметром *IP_PROXY_HEADER*.

Предупреждение: Убедитесь, что вы и в самом деле используете обратный прокси и что он устанавливает этот заголовок, иначе пользователи смогут подделать IP адрес.

Примечание: По умолчанию этот параметр не включён.

См.также:

Работа за обратным прокси, *Ограничение частоты запросов*, *IP_PROXY_HEADER*, *IP_PROXY_OFFSET*

2.16.46 IP_PROXY_HEADER

Добавлено в версии 2.14.

Указывает, из какого заголовка Weblate должен получить IP-адрес при включении параметра *IP_BEHIND_REVERSE_PROXY*.

По умолчанию установлен в `HTTP_X_FORWARDED_FOR`.

См.также:

Работа за обратным прокси, *Ограничение частоты запросов*, `SECURE_PROXY_SSL_HEADER`, *IP_BEHIND_REVERSE_PROXY*, *IP_PROXY_OFFSET*

2.16.47 IP_PROXY_OFFSET

Добавлено в версии 2.14.

Указывает, какая часть параметра *IP_PROXY_HEADER* используется в качестве IP-адреса клиента.

В зависимости от вашей установки, этот заголовок может состоять из нескольких IP адресов (например, `X-Forwarded-For: a, b, client-ip`), и здесь вы можете настроить, какой адрес из заголовка используется в качестве IP адреса клиента.

Предупреждение: Задание этого параметра влияет на безопасность вашей установки, вы должны настроить его на использование только доверенных прокси для определения IP адреса.

По умолчанию установлен в 0.

См.также:

Работа за обратным прокси, *Ограничение частоты запросов*, `SECURE_PROXY_SSL_HEADER`, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`

2.16.48 LEGAL_URL

Добавлено в версии 3.5.

URL-адрес, по которому ваш экземпляр Weblate показывает свои юридические документы.

Подсказка: Полезен, если вы размещаете свои юридические документы за пределами Weblate'а для дальнейшего их встраивания в Weblate, для получения подробной информации обратитесь к разделу *Правовые вопросы*.

Пример:

```
LEGAL_URL = "https://weblate.org/terms/"
```

2.16.49 LICENSE_EXTRA

Дополнительные лицензии для включения в список вариантов лицензий.

Примечание: Описание каждой лицензии должно быть кортежем, состоящим из её краткого наименования, длинного наименования и URL-адреса.

Например:

```
LICENSE_EXTRA = [
    (
        "AGPL-3.0",
        "GNU Affero General Public License v3.0",
        "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
    ),
]
```

2.16.50 LICENSE_FILTER

Изменено в версии 4.3: Присваивание данному параметру пустой строки теперь отключает предупреждение о лицензиях.

Список с фильтром лицензий, которые надлежит показывать. Если задано пустое множество, то предупреждение о лицензиях отключается.

Примечание: Этот фильтр использует сокращённые имена лицензий.

Например:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

Следующее отключает предупреждение о лицензиях:

```
LICENSE_FILTER = set()
```

См.также:

alerts

2.16.51 LICENSE_REQUIRED

Определяет, требуется ли атрибут лицензии в *конфигурации компонента*.

Примечание: По умолчанию этот параметр выключен.

2.16.52 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Должна ли длина данного перевода быть ограничена. Ограничением является количество символов исходной строки, увеличенное в 10 раз.

Подсказка: Установите его значение в `False` для разрешения более длинных переводов (до 10 000 символов) независимо от длины исходной строки.

Примечание: По умолчанию установлен в `True`.

2.16.53 LOCALIZE_CDN_URL и LOCALIZE_CDN_PATH

Эти параметры конфигурируют надстройку *CDN локализации JavaScript'a*. Параметр *`LOCALIZE_CDN_URL`* определяет корневой URL-адрес, по которому доступен CDN локализации, а параметр *`LOCALIZE_CDN_PATH`* определяет путь, по которому Weblate должен сохранять сгенерированные файлы, которые затем будут раздаваться с *`LOCALIZE_CDN_URL`*.

Подсказка: На Hosted Weblate для этого параметра используется значение `https://weblate-cdn.com/`.

См.также:

CDN локализации JavaScript'a

2.16.54 LOGIN_REQUIRED_URLS

Список URL-адресов, для посещения которых требуется выполнить вход в Weblate (помимо встроенных в Weblate стандартных правил).

Подсказка: Он позволяет вам защитить паролем всю установку:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

Подсказка: Также желательно заблокировать доступ к API, как показано в примере выше.

См.также:

REQUIRE_LOGIN

2.16.55 LOGIN_REQUIRED_URLS_EXCEPTIONS

Список исключений для *LOGIN_REQUIRED_URLS*. Если не указан, пользователям будет разрешён доступ к странице входа.

Некоторые исключения, которые вы, возможно, захотите включить:

```
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
)
```

2.16.56 MATOMO_SITE_ID

Идентификатор сайта в Matomo (бывший Piwik), который вы хотите отслеживать.

Примечание: Эта интеграция не поддерживает Matomo Tag Manager.

См.также:

MATOMO_URL

2.16.57 MATOMO_URL

Полный URL-адрес (включая завершающую косую черту) установки Matomo (бывшего Piwik), которую вы хотите использовать для отслеживания использования Weblate. Для получения более подробной информации обратитесь к <https://matomo.org/>.

Подсказка: Эта интеграция не поддерживает Matomo Tag Manager.

Например:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

См.также:

MATOMO_SITE_ID

2.16.58 MT_SERVICES

Изменено в версии 3.0: Параметр был переименован с `MACHINE_TRANSLATION_SERVICES` на *MT_SERVICES*` для соответствия названиям других параметров машинного перевода.

Список включённых сервисов машинного перевода.

Примечание: Многим сервисам нужна дополнительная настройка, например, задание ключей API, для получения более подробной информации обратитесь к их документации в разделе *Машинный перевод*.

```
MT_SERVICES = (
    "weblate.machinery.apertium.ApertiumAPYTranslation",
    "weblate.machinery.deepl.DeepLTranslation",
    "weblate.machinery.glosbe.GlosbeTranslation",
    "weblate.machinery.google.GoogleTranslation",
    "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    "weblate.machinery.mymemory.MyMemoryTranslation",
    "weblate.machinery.tmserver.AmagamaTranslation",
    "weblate.machinery.tmserver.TMServerTranslation",
    "weblate.machinery.yandex.YandexTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.machinery.saptranslationhub.SAPTranslationHub",
    "weblate.memory.machine.WeblateMemory",
)
```

См.также:

Машинный перевод, *Автоматические предложения*

2.16.59 MT_APERTIUM_APY

URL-адрес сервера Apertium-APy, https://wiki.apertium.org/wiki/Apertium-apy

См.также:

Apertium, *Машинный перевод*, *Автоматические предложения*

2.16.60 MT_AWS_ACCESS_KEY_ID

Идентификатор ключа доступа к Amazon Translate.

См.также:

AWS, *Машинный перевод*, *Автоматические предложения*

2.16.61 MT_AWS_SECRET_ACCESS_KEY

Секретный ключ API к Amazon Translate.

См.также:

AWS, *Машинный перевод*, *Автоматические предложения*

2.16.62 MT_AWS_REGION

Название используемого региона для Amazon Translate.

См.также:

AWS, *Машинный перевод*, *Автоматические предложения*

2.16.63 MT_BAIDU_ID

Идентификатор клиента для API Baidu Zhiyun, вы можете зарегистрироваться по адресу https://api.fanyi.baidu.com/api/trans/product/index

См.также:

Машинный перевод через API Baidu, *Машинный перевод*, *Автоматические предложения*

2.16.64 MT_BAIDU_SECRET

Секрет клиента для API Baidu Zhiyun, вы можете зарегистрироваться по адресу https://api.fanyi.baidu.com/api/trans/product/index

См.также:

Машинный перевод через API Baidu, *Машинный перевод*, *Автоматические предложения*

2.16.65 MT_DEEPL_API_VERSION

Добавлено в версии 4.1.1.

Версия API, используемая для сервиса DeepL. Версия ограничивает область использования:

v1 Предназначена для CAT-инструментов и может быть использована с пользовательской подпиской.

v2 Предназначена для использования через API, а подписка основана на использовании.

Ранее Weblate классифицировался DeepL как CAT-инструмент, поэтому предполагалось использовать API версии v1, но теперь предполагается использовать API версии v2. Поэтому по умолчанию параметр имеет значение v2, но вы можете изменить его на v1 в случае, если у вас есть существующая CAT-подписка и вы хотите, чтобы Weblate использовал её.

См.также:

DeepL, *Машинный перевод*, *Автоматические предложения*

2.16.66 MT_DEEPL_KEY

Ключ API для API DeepL, вы можете зарегистрироваться по адресу https://www.deepl.com/pro.html

См.также:

DeepL, *Машинный перевод*, *Автоматические предложения*

2.16.67 MT_GOOGLE_KEY

Ключ API для API переводчика Google версии v2, вы можете зарегистрироваться по адресу https://cloud.google.com/translate/docs

См.также:

Google Translate, *Машинный перевод*, *Автоматические предложения*

2.16.68 MT_GOOGLE_CREDENTIALS

JSON-файл учётных данных API версии v3, полученный в консоли облака Google. Пожалуйста, укажите полный путь в формате пути операционной системы. Учётные данные привязаны к сервису-учётной записи, связанной с определённым проектом. Для получения более подробной информации обратитесь по адресу https://cloud.google.com/docs/authentication/getting-started.

2.16.69 MT_GOOGLE_PROJECT

Идентификатор проекта API версии v3 облака Google с активированным тарифным планом и сервисом перевода. Для получения более подробной информации обратитесь по адресу https://cloud.google.com/appengine/docs/standard/nodejs/building-app/creating-project

2.16.70 MT_GOOGLE_LOCATION

API версии v3 App Engine облака Google может зависеть от местоположения. Измените его соответствующим образом, если значение по умолчанию `global` для вас не подходит.

Для получения более подробной информации обратитесь по адресу https://cloud.google.com/appengine/docs/locations

См.также:

Google Translate API V3 (расширенный)

2.16.71 MT_MICROSOFT_BASE_URL

Базовый URL-адрес региона, как определено в разделе «Базовые URL-адреса».

По умолчанию равен `api.cognitive.microsofttranslator.com` для Azure Global.

Для Azure China используйте, пожалуйста, `api.translator.azure.cn`.

2.16.72 MT_MICROSOFT_COGNITIVE_KEY

Клиентский ключ для API переводчика Microsoft Cognitive Services.

См.также:

Переводчик Microsoft Cognitive Services, *Машинный перевод*, *Автоматические предложения*, Cognitive Services — API перевода текста, Портал Microsoft Azure

2.16.73 MT_MICROSOFT_REGION

Префикс региона, как определено в разделе «Проверка подлинности с помощью ресурса с несколькими службами».

2.16.74 MT_MICROSOFT_ENDPOINT_URL

Конечная точка URL-адреса региона, как определено в разделе «Проверка подлинности с помощью маркера доступа».

По умолчанию равен `api.cognitive.microsoft.com` для Azure Global.

Для Azure China используйте, пожалуйста, свою конечную точку на портале Azure Portal.

2.16.75 MT_MODERNMT_KEY

Ключ API для движка машинного перевода ModernMT.

См.также:

ModernMT `MT_MODERNMT_URL`

2.16.76 MT_MODERNMT_URL

URL-адрес ModernMT. По умолчанию для облачного сервиса используется `https://api.modernmt.com/`.

См.также:

ModernMT `MT_MODERNMT_KEY`

2.16.77 MT_MYMEMORY_EMAIL

Идентификационный адрес электронной почты MyMemory. Разрешает выполнять 1000 запросов в день.

См.также:

MyMemory, *Машинный перевод*, *Автоматические предложения*, MyMemory: технические спецификации API

2.16.78 MT_MYMEMORY_KEY

Ключ доступа MyMemory для частной памяти переводов, используйте его совместно с параметром `MT_MYMEMORY_USER`.

См.также:

MyMemory, *Машинный перевод*, *Автоматические предложения*, MyMemory: генератор ключа API

2.16.79 MT_MYMEMORY_USER

Идентификатор пользователя MyMemory для частной памяти переводов, используйте его совместно с параметром `MT_MYMEMORY_KEY`.

См.также:

MyMemory, *Машинный перевод*, *Автоматические предложения*, MyMemory: генератор ключа API

2.16.80 MT_NETEASE_KEY

Ключ приложения для API NetEase Sight, вы можете зарегистрироваться по адресу https://sight.youdao.com/

См.также:

API машинного перевода NetEase Sight, *Машинный перевод*, *Автоматические предложения*

2.16.81 MT_NETEASE_SECRET

Секрет приложения для API NetEase Sight, вы можете зарегистрироваться по адресу https://sight.youdao.com/

См.также:

API машинного перевода NetEase Sight, *Машинный перевод*, *Автоматические предложения*

2.16.82 MT_TMSERVER

URL-адрес, по которому запущен tmserver.

См.также:

tmserver, *Машинный перевод*, *Автоматические предложения*, tmserver

2.16.83 MT_YANDEX_KEY

Ключ API для API Яндекс.Переводчика, вы можете зарегистрироваться по адресу https://yandex.ru/dev/translate/

См.также:

Yandex Translate (Яндекс.Переводчик), *Машинный перевод*, *Автоматические предложения*

2.16.84 MT_YOUDAO_ID

Идентификатор клиента для API Youdao Zhiyun, вы можете зарегистрироваться по адресу https://ai.youdao.com/product-fanyi-text.s.

См.также:

API машинного перевода Youdao Zhiyun, *Машинный перевод*, *Автоматические предложения*

2.16.85 MT_YOUDAO_SECRET

Секрет клиента для API Youdao Zhiyun, вы можете зарегистрироваться по адресу https://ai.youdao.com/product-fanyi-text.s.

См.также:

API машинного перевода Youdao Zhiyun, *Машинный перевод*, *Автоматические предложения*

2.16.86 MT_SAP_BASE_URL

URL-адрес API сервиса SAP Translation Hub.

См.также:

SAP Translation Hub, *Машинный перевод*, *Автоматические предложения*

2.16.87 MT_SAP_SANDBOX_APIKEY

Ключ API для использования API песочницы

См.также:

SAP Translation Hub, *Машинный перевод*, *Автоматические предложения*

2.16.88 MT_SAP_USERNAME

Ваше имя пользователя SAP

См.также:

SAP Translation Hub, *Машинный перевод*, *Автоматические предложения*

2.16.89 MT_SAP_PASSWORD

Ваш пароль SAP

См.также:

SAP Translation Hub, *Машинный перевод*, *Автоматические предложения*

2.16.90 MT_SAP_USE_MT

Использовать ли также сервисы машинного перевода в дополнение к терминологической базе данных. Возможные значения: `True` или `False`

См.также:

SAP Translation Hub, *Машинный перевод*, *Автоматические предложения*

2.16.91 NEARBY_MESSAGES

Сколько показывать строк около переводимой в настоящее время строки. Это просто значение по умолчанию, пользователи могут настроить его в *своём профиле*.

2.16.92 PAGURE_CREDENTIALS

Добавлено в версии 4.3.2.

Список учётных данных для серверов Pagure.

Подсказка: Используйте этот параметр для того, чтобы Weblate мог взаимодействовать с одной машины с несколькими разными серверами Pagure, а не только с одним, как при использовании `PAGURE_USERNAME` и `PAGURE_TOKEN`.

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.93 PAGURE_USERNAME

Добавлено в версии 4.3.2.

Имя пользователя Pagure, используемое для отправки запросов на слияние для обновления перевода.

См.также:

`PAGURE_CREDENTIALS`, *Pagure*

2.16.94 PAGURE_TOKEN

Добавлено в версии 4.3.2.

Pargue'вский персональный токен доступа, используемый для вызовов API, необходимых для обновления перевода.

См.также:

`PAGURE_CREDENTIALS`, *Pagure*, Pagure API

2.16.95 RATELIMIT_ATTEMPTS

Добавлено в версии 3.2.

Максимальное количество попыток аутентификации до применения ограничения частоты запросов.

По умолчанию установлен в 5.

См.также:

Ограничение частоты запросов, `RATELIMIT_WINDOW`, `RATELIMIT_LOCKOUT`

2.16.96 RATELIMIT_WINDOW

Добавлено в версии 3.2.

Как долго будет приниматься аутентификация после применения ограничения частоты запросов.

Количество секунд, по умолчанию равен 300 (5 минут).

См.также:

Ограничение частоты запросов, `RATELIMIT_ATTEMPTS`, `RATELIMIT_LOCKOUT`

2.16.97 RATELIMIT_LOCKOUT

Добавлено в версии 3.2.

Как долго аутентификация будет заблокирована после применения ограничения частоты запросов.

Количество секунд, по умолчанию равен 600 (10 минут).

См.также:

Ограничение частоты запросов, `RATELIMIT_ATTEMPTS`, `RATELIMIT_WINDOW`

2.16.98 REGISTRATION_ALLOW_BACKENDS

Добавлено в версии 4.1.

Список методов аутентификации (backends), с помощью которых доступна регистрация. Это вводит ограничения только на новые регистрации. Независимо от данного параметра пользователи смогут добавлять и использовать любые разрешённые в параметрах и настроенные методы аутентификации.

Рекомендуется держать параметр *REGISTRATION_OPEN* включённым, но ограничить доступные методы регистрации, иначе пользователи смогут регистрироваться, но Weblate не будет показывать ссылки на регистрацию в пользовательском интерфейсе.

Пример:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

Подсказка: Имена бэкэндов совпадают с именами, используемыми в URL-адресах для аутентификации.

См.также:

REGISTRATION_OPEN, *Аутентификация*

2.16.99 REGISTRATION_CAPTCHA

Значение, которое может быть равно только `True` или `False`, указывает, защищена ли регистрация новых учётных записей капчей. Этот параметр является необязательным и если он не указан, будет принято значение по умолчанию `True`.

Если он включён, ко всем страницам, на которых пользователь вводит свой адрес электронной почты, добавляется капча:

- Регистрация новой учётной записи.
- Восстановление пароля.
- Добавление в учётную запись адреса электронной почты.
- Контактная форма для пользователей, не вошедших в систему.

2.16.100 REGISTRATION_EMAIL_MATCH

Добавлено в версии 2.17.

Позволяет вам отфильтровать допустимые для регистрации адреса электронной почты.

По умолчанию равен `.*`, что позволяет зарегистрироваться с любым адресом электронной почты.

Вы можете использовать этот параметр для ограничения регистрации одним доменом электронной почты:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.101 REGISTRATION_OPEN

Разрешена ли в настоящее время регистрация новых учётных записей. Этот необязательный параметр может остаться в значении по умолчанию `True` или быть изменённым на `False`.

Этот параметр влияет на встроенную аутентификацию по адресу электронной почты или через Python Social Auth (при помощи параметра *REGISTRATION_ALLOW_BACKENDS* вы можете занести определённые бэкэнды в белый список).

Примечание: При использовании сторонних методов аутентификации, вроде *аутентификации через LDAP*, этот параметр просто скрывает форму регистрации, но новые пользователи всё равно смогут входить в систему и создавать учётные записи.

См.также:

REGISTRATION_ALLOW_BACKENDS, *REGISTRATION_EMAIL_MATCH*, *Аутентификация*

2.16.102 REPOSITORY_ALERT_THRESHOLD

Добавлено в версии 4.0.2.

Порог срабатывания предупреждения для устаревших репозиториев или репозиториев, содержащих слишком много изменений. По умолчанию равен 25.

См.также:

alerts

2.16.103 REQUIRE_LOGIN

Добавлено в версии 4.1.

Этот параметр включает *LOGIN_REQUIRED_URLS* и настраивает его так, что будет запрашиваться авторизация пользователя для всех API-вызовов REST.

Примечание: Этот параметр реализован в *примере файла настроек*. В Docker'е используйте переменную. *WEBLATE_REQUIRE_LOGIN*.

2.16.104 SENTRY_DSN

Добавлено в версии 3.9.

DSN Sentry для *сбора отчётов об ошибках*.

См.также:

Интеграция Sentry с Django

2.16.105 SESSION_COOKIE_AGE_AUTHENTICATED

Добавлено в версии 4.3.

Задать время истечения сессии аутентифицированных пользователей. Этот параметр дополняет `SESSION_COOKIE_AGE`, который используется для неаутентифицированных пользователей.

См.также:

`SESSION_COOKIE_AGE`

2.16.106 SIMPLIFY_LANGUAGES

Использовать простые коды языка для умолчательных комбинаций языка/страны. Например, при переводе `ru_RU` будет использоваться код языка `ru`. Обычно это желательное поведение, поскольку для таких умолчательных комбинаций оно упрощает перечисление языков.

Выключите этот параметр, если вы хотите разделять переводы для каждого варианта.

2.16.107 SITE_DOMAIN

Настраивает домен сайта. Это необходимо для получения корректных абсолютных ссылок во многих местах (например, в письмах активации, в уведомлениях или в RSS-каналах).

Если Weblate работает на нестандартном порту, добавьте его и сюда тоже.

Примеры:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

Примечание: Этот параметр должен содержать только доменное имя. Для настройки протокола (включения и принудительного включения HTTPS) используйте параметр *`ENABLE_HTTPS`*, а для изменения URL-адреса — параметр *`URL_PREFIX`*.

Подсказка: На контейнере Docker'а домен сайта настраивается через переменную окружения *`WEBLATE_ALLOWED_HOSTS`*.

См.также:

Установка правильного домена сайта, *Настройка разрешённых хостов*, *Правильная настройка HTTPS* *`WEBLATE_SITE_DOMAIN`*, *`ENABLE_HTTPS`*

2.16.108 SITE_TITLE

Заголовок сайта, который будет использоваться как на самом сайте, так и указываться в электронных письмах.

2.16.109 SPECIAL_CHARS

Дополнительные символы для включения в визуальную клавиатуру, смотрите раздел *Визуальная клавиатура*.

Значение по умолчанию:

```
SPECIAL_CHARS = ("\t", "\n", "…")
```

2.16.110 SINGLE_PROJECT

Добавлено в версии 3.8.

Перенаправляет пользователей непосредственно в проект или компонент вместо отображения панели управления. Вы можете установить параметр в `True`, в этом случае он будет работать только в том случае, если в Weblate'е и в самом деле есть только один проект. Или же можно задать плашку проекта, тогда пользователь будет безоговорочно перенаправлен в этот проект.

Изменено в версии 3.11: Теперь параметр также принимает плашку проекта для принудительного отображения этого единственного проекта.

Пример:

```
SINGLE_PROJECT = "test"
```

2.16.111 STATUS_URL

URL-адрес, по которому ваш экземпляр Weblate сообщает о своем статусе.

2.16.112 SUGGESTION_CLEANUP_DAYS

Добавлено в версии 3.2.1.

Автоматическое удаление предложения через заданное количество дней. По умолчанию установлен в ``None`, что означает отсутствие удалений.

2.16.113 UPDATE_LANGUAGES

Добавлено в версии 4.3.2.

Определяет, должна ли база данных языков обновляться во время общей миграции базы данных. По умолчанию включено. Этот параметр никак не влияет на выполнение `setuplang`.

См.также:

Встроенные определения языков

2.16.114 URL_PREFIX

Этот параметр позволяет вам запускать Weblate по какому-либо пути (в противном случае он считает, что запущен из корня веб-сервера).

Примечание: Для использования этого параметра вам также нужно настроить ваш сервер на срезание этого префикса. Например, при использовании WSGI это можно сделать, установив псевдоним `WSGIScriptAlias`.

Подсказка: Префикс должен начинаться с символа `/`.

Пример:

```
URL_PREFIX = "/translations"
```

Примечание: Этот параметр не работает со встроенным сервером Django, чтобы он реагировал на этот префикс, вам придётся подправить `urls.py`.

2.16.115 VCS_BACKENDS

Настройка доступных систем контроля версий.

Примечание: Weblate пытается использовать все поддерживаемые системы, для которых у вас есть инструменты.

Подсказка: При помощи этого параметра вы можете ограничить их выбор или добавить пользовательские системы контроля версий.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

См.также:

Интеграция с системой контроля версий

2.16.116 VCS_CLONE_DEPTH

Добавлено в версии 3.10.2.

Настраивает, насколько глубоким должно быть клонирование репозиториев Weblate'ом.

Примечание: В настоящее время этот параметр поддерживается только в *Git*. По умолчанию Weblate делает поверхностные клоны репозиториев, чтобы сделать клонирование быстрее и сэкономить место на диске. В зависимости от использования (например, при использовании пользовательских *надстроек*), вы можете захотеть увеличить глубину клонирования или полностью отключить поверхностное клонирование, что можно сделать, установив этот параметр в значение 0.

Подсказка: В случае, если вы при отправке коммитов из Weblate'а получили ошибку `fatal: protocol error: expected old/new/ref, got 'shallow <commit hash>'`, полностью отключите поверхностные клоны, установив соответствующий параметр:

```
VCS_CLONE_DEPTH = 0
```

2.16.117 WEBLATE_ADDONS

Список доступных для использования надстроек. Чтобы использовать надстройку, нужно также включить её для конкретного компонента перевода. По умолчанию в список включены все встроенные надстройки. Если вы будете расширять список, то, вероятно, вы захотите оставить эти дополнения включёнными, например:

```
WEBLATE_ADDONS = (
    # Built-in addons
    "weblate.addons.gettext.GenerateMoAddon",
    "weblate.addons.gettext.UpdateLinguasAddon",
    "weblate.addons.gettext.UpdateConfigureAddon",
    "weblate.addons.gettext.MsgmergeAddon",
    "weblate.addons.gettext.GettextCustomizeAddon",
    "weblate.addons.gettext.GettextAuthorComments",
    "weblate.addons.cleanup.CleanupAddon",
    "weblate.addons.consistency.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",
    # Addon you want to include
    "weblate.addons.example.ExampleAddon",
)
```

Примечание: Удаление дополнения из этого списка не уберёт его из компонентов. В этом случае Weblate завершится крахом. Пожалуйста, уберите дополнение из всех компонентов прежде чем удалять его из данного списка.

См.также:

Надстройки, `DEFAULT_ADDONS`

2.16.118 WEBLATE_EXPORTERS

Добавлено в версии 4.2.

Список доступных экспортеров, предлагающих скачивание переводов или глоссариев в различных форматах файлов.

См.также:

Поддерживаемые форматы файлов

2.16.119 WEBLATE_FORMATS

Добавлено в версии 3.0.

Список доступных для использования форматов файлов.

Примечание: В списке по умолчанию уже есть распространённые форматы.

См.также:

Поддерживаемые форматы файлов

2.16.120 WEBLATE_GPG_IDENTITY

Добавлено в версии 3.1.

Идентификатор, используемый Weblate'ом для подписывания коммитов Git, например:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

В GPG'шной связке ключей Weblate ищет соответствующий ключ (файл `home/.gnupg` в каталоге `DATA_DIR`). Если ключ не найден, он генерируется, подробнее смотрите в разделе *Подписание коммитов Git с помощью GnuPG*.

См.также:

Подписание коммитов Git с помощью GnuPG

2.16.121 WEBSITE_REQUIRED

Defines whether *Веб-сайт проекта* has to be specified when creating a project. Turned on by default as that suits public server setups.

2.17 Пример файла настроек

Следующий пример поставляется с Weblate'ом в виде файла `weblate/settings_example.py`:

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#


import os
import platform
from logging.handlers import SysLogHandler

#
# Django settings for Weblate project.
#

DEBUG = True

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password.
        "PASSWORD": "",
        # Set to empty string for localhost.
        "HOST": "127.0.0.1",
        # Set to empty string for default.
        "PORT": "",
        # Customizations for databases.
        "OPTIONS": {
            # In case of using an older MySQL server,
            # which has MyISAM as a default storage
            # "init_command": "SET storage_engine=INNODB",
            # Uncomment for MySQL older than 5.7:
            # "init_command": "SET sql_mode='STRICT_TRANS_TABLES'",
            # Set emoji capable charset for MySQL:
            # "charset": "utf8mb4",
            # Change connection timeout in case you get MySQL gone away error:
            # "connect_timeout": 28800,
        },
    }
}

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# Data directory
DATA_DIR = os.path.join(BASE_DIR, "data")

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name
# although not all choices may be available on all operating systems.
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = "UTC"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

LANGUAGES = (
    ("ar", "العربية"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@latin", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("da", "Dansk"),
    ("de", "Deutsch"),
    ("en", "English"),
    ("el", "Ελληνικά"),
    ("en-gb", "English (United Kingdom)"),
    ("es", "Español"),
    ("fi", "Suomi"),
    ("fr", "Français"),
    ("gl", "Galego"),
    ("he", "עברית"),
    ("hu", "Magyar"),
    ("hr", "Hrvatski"),
    ("id", "Indonesia"),
    ("is", "Íslenska"),
    ("it", "Italiano"),
    ("ja", "日本語"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "简体中文"),
    ("zh-hant", "繁体中文"),
)

SITE_ID = 1
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
USE_L10N = True

# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True

# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(DATA_DIR, "static")

# URL prefix for static files.
STATIC_URL = f"{URL_PREFIX}/static/"

# Additional locations of static files
STATICFILES_DIRS = (
    # Put strings here, like "/home/html/static" or "C:/www/django/static".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    "django.contrib.staticfiles.finders.FileSystemFinder",
    "django.contrib.staticfiles.finders.AppDirectoriesFinder",
    "compressor.finders.CompressorFinder",
)

# Make this unique, and don't share it with anybody.
# You can generate it using weblate/examples/generate-secret-key
SECRET_KEY = ""

_TEMPLATE_LOADERS = [
    "django.template.loaders.filesystem.Loader",
    "django.template.loaders.app_directories.Loader",
]
if not DEBUG:
    _TEMPLATE_LOADERS = [("django.template.loaders.cached.Loader", _TEMPLATE_
↪LOADERS)]
TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
            "loaders": _TEMPLATE_LOADERS,
        },
    }
]


# GitHub username for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None

# GitLab username for sending merge requests.
# Please see the documentation for more details.
GITLAB_USERNAME = None
GITLAB_TOKEN = None

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

SOCIAL_AUTH_BITBUCKET_KEY = ""
SOCIAL_AUTH_BITBUCKET_SECRET = ""
SOCIAL_AUTH_BITBUCKET_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "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},
    },
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.CharsPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.PastPasswordsValidator"},
    # Optional password strength validation by django-zxcvbn-password
    # {
    #     "NAME": "zxcvbn_password.ZXCVBNValidator",
    #     "OPTIONS": {
    #         "min_score": 3,
    #         "user_attributes": ("username", "email", "full_name")
    #     }
    # },
]

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
    "social_django.middleware.SocialAuthExceptionMiddleware",
    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",
    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "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.
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
↪"},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "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
        # "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},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        # Django Authentication Using LDAP
        "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # SAML IdP
        "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    },
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    #     "weblate.machinery.apertium.ApertiumAPYTranslation",
    #     "weblate.machinery.baidu.BaiduTranslation",
    #     "weblate.machinery.deepl.DeepLTranslation",
    #     "weblate.machinery.glosbe.GlosbeTranslation",
    #     "weblate.machinery.google.GoogleTranslation",
    #     "weblate.machinery.googlev3.GoogleV3Translation",
    #     "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    #     "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    #     "weblate.machinery.modernmt.ModernMTTranslation",
    #     "weblate.machinery.mymemory.MyMemoryTranslation",
    #     "weblate.machinery.netease.NeteaseSightTranslation",
    #     "weblate.machinery.tmserver.AmagamaTranslation",
    #     "weblate.machinery.tmserver.TMServerTranslation",
    #     "weblate.machinery.yandex.YandexTranslation",
    #     "weblate.machinery.saptranslationhub.SAPTranslationHub",
    #     "weblate.machinery.youdao.YoudaoTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APy server
MT_APERTIUM_APY = None

# DeepL API key
MT_DEEPL_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

# ModernMT
MT_MODERNMT_KEY = None

# MyMemory identification email, see
# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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

# Title of site to use
SITE_TITLE = "Weblate"

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

# Use HTTPS when creating redirect URLs for social authentication, see
# documentation for more details:
# https://python-social-auth-docs.readthedocs.io/en/latest/configuration/settings.
↪html#processing-redirects-and-urlopen
SOCIAL_AUTH_REDIRECT_IS_HTTPS = ENABLE_HTTPS

# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view
CSRF_FAILURE_VIEW = "weblate.trans.views.error.csrf_failure"
SESSION_COOKIE_SECURE = ENABLE_HTTPS
SESSION_COOKIE_HTTPONLY = True
# SSL redirect
SECURE_SSL_REDIRECT = ENABLE_HTTPS
# Sent referrrer only for same origin links
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",)  # Allowing HTTP access to health check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
```

(continues on next page)

(продолжение с предыдущей страницы)

```
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer
# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
SIMPLIFY_LANGUAGES = True

# Render forms using bootstrap
CRISPY_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
# CHECK_LIST = (
```

(continues on next page)

(продолжение с предыдущей страницы)

```
#     "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.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.qt.QtFormatCheck",
#     "weblate.checks.qt.QtPluralCheck",
#     "weblate.checks.ruby.RubyFormatCheck",
#     "weblate.checks.consistency.PluralsCheck",
#     "weblate.checks.consistency.SamePluralsCheck",
#     "weblate.checks.consistency.ConsistencyCheck",
#     "weblate.checks.consistency.TranslatedCheck",
#     "weblate.checks.chars.EscapedNewlineCountingCheck",
#     "weblate.checks.chars.NewLineCountCheck",
#     "weblate.checks.markup.BBCodeCheck",
#     "weblate.checks.chars.ZeroWidthSpaceCheck",
#     "weblate.checks.render.MaxSizeCheck",
#     "weblate.checks.markup.XMLValidityCheck",
#     "weblate.checks.markup.XMLTagsCheck",
#     "weblate.checks.markup.MarkdownRefLinkCheck",
#     "weblate.checks.markup.MarkdownLinkCheck",
#     "weblate.checks.markup.MarkdownSyntaxCheck",
#     "weblate.checks.markup.URLCheck",
#     "weblate.checks.markup.SafeHTMLCheck",
#     "weblate.checks.placeholders.PlaceholderCheck",
#     "weblate.checks.placeholders.RegexCheck",
#     "weblate.checks.duplicate.DuplicateCheck",
#     "weblate.checks.source.OptionalPluralCheck",
#     "weblate.checks.source.EllipsisCheck",
#     "weblate.checks.source.MultipleFailingCheck",
#     "weblate.checks.source.LongUntranslatedCheck",
#     "weblate.checks.format.MultipleUnnamedFormatsCheck",
#     "weblate.checks.glossary.GlossaryCheck",
# )
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# 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.autotranslate.AutoTranslateAddon",
#     "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.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": {
        "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,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            "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},
    {"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 = (
```

(continues on next page)

(продолжение с предыдущей страницы)

```
#    rf"{URL_PREFIX}/accounts/(.*)$",  # Required for login
#    rf"{URL_PREFIX}/admin/login/(.*)$",  # Required for admin login
#    rf"{URL_PREFIX}/static/(.*)$",  # Required for development mode
#    rf"{URL_PREFIX}/widgets/(.*)$",  # Allowing public access to widgets
#    rf"{URL_PREFIX}/data/(.*)$",  # Allowing public access to data exports
#    rf"{URL_PREFIX}/hooks/(.*)$",  # Allowing public access to notification hooks
#    rf"{URL_PREFIX}/healthz/$",  # Allowing public access to health check
#    rf"{URL_PREFIX}/api/(.*)$",  # Allowing access to API
#    rf"{URL_PREFIX}/js/i18n/$",  # JavaScript localization
#    rf"{URL_PREFIX}/contact/$",  # Optional for contact form
#    rf"{URL_PREFIX}/legal/(.*)$",  # Optional for legal app
# )

# Silence some of the Django system checks
SILENCED_SYSTEM_CHECKS = [
    # We have modified django.contrib.auth.middleware.AuthenticationMiddleware
    # as weblate.accounts.middleware.AuthenticationMiddleware
    "admin.E408"
]

# Celery worker configuration for testing
# CELERY_TASK_ALWAYS_EAGER = True
# CELERY_BROKER_URL = "memory://"
# CELERY_TASK_EAGER_PROPAGATES = True
# Celery worker configuration for production
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

# Celery settings, it is not recommended to change these
CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_SCHEDULE_FILENAME = os.path.join(DATA_DIR, "celery", "beat-schedule")
CELERY_TASK_ROUTES = {
    "weblate.trans.tasks.auto_translate": {"queue": "translate"},
    "weblate.accounts.tasks.notify_*": {"queue": "notify"},
    "weblate.accounts.tasks.send_mails": {"queue": "notify"},
    "weblate.utils.tasks.settings_backup": {"queue": "backup"},
    "weblate.utils.tasks.database_backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup_service": {"queue": "backup"},
    "weblate.memory.tasks.*": {"queue": "memory"},
}

# Enable plain database backups
DATABASE_BACKUP = "plain"

# Enable auto updating
AUTO_UPDATE = False

# PGP commits signing
WEBLATE_GPG_IDENTITY = None

# Third party services integration
MATOMO_SITE_ID = None
MATOMO_URL = None
GOOGLE_ANALYTICS_ID = None
SENTRY_DSN = None
AKISMET_API_KEY = None
```

2.18 Команды управления

Примечание: Запуск команд управления от имени пользователя, отличного от того, под которым работает ваш веб-сервер, может привести к тому, что права доступа на файлы будут выставлены неверно. Для получения более подробной информации обратитесь к разделу *Права доступа к файлам*.

Основные команды управления доступны из скрипта `./manage.py` в исходниках Django, или их расширенный набор из скрипта **weblate**, устанавливаемом поверх Weblate.

2.18.1 Вызов команд управления

Как упоминалось ранее, вызов команд зависит от того, как вы установили Weblate.

При использовании для Weblate'а виртуального окружения вы можете либо указать полный путь к команде **weblate**, либо перед вызовом активировать виртуальное окружение:

```
# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

Если вы напрямую используете исходный код (либо из tar-архива, либо из Git'а), то скрипт управления `./manage.py` доступен из исходников Weblate'а. Для его запуска выполните следующую команду:

```
python ./manage.py list_versions
```

Если вы установили Weblate с помощью программы установки pip или pip3 или скрипта `./setup.py`, команда **weblate** устанавливается в каталог, доступный через системную переменную окружения PATH (или через переменную PATH виртуального окружения), откуда вы можете использовать её для управления Weblate'ом:

```
weblate list_versions
```

В случае использования образа Docker'а скрипт устанавливается так же, как и выше, и вы можете запустить его с помощью команды **docker exec**:

```
docker exec --user weblate <container> weblate list_versions
```

Для команды **docker-compose** процесс аналогичен, достаточно использовать вызов **docker-compose exec**:

```
docker-compose exec --user weblate weblate weblate list_versions
```

В случае, если вам нужно передать ему файл, вы можете временно добавить том:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /tmp/
↪users.json
```

См.также:

Установка с помощью Docker'а, *Установка на Debian и Ubuntu*, *Установка на SUSE и openSUSE*, *Установка на RedHat, Fedora и CentOS*, *Установка из исходников*

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

Добавлено в версии 2.5.

Импортирует перевод из файла, чтобы использовать его как предложение для данного перевода. Команда пропускает дубликаты перевода; добавляются только отличающиеся.

--author USER@EXAMPLE.COM
Электронная почта автора предложений. До выполнения импорта следует убедиться, что этот пользователь существует (при необходимости его можно создать через интерфейс администратора).

Пример:

```
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>

Добавлено в версии 2.5.

Выполняет автоматический перевод на основе переводов других компонентов.

--source PROJECT/COMPONENT
Задаёт компонент, используемый в качестве доступных для перевода исходных строк. Если не задан, то используются все компоненты проекта.

--user USERNAME
Задаёт имя пользователя, указываемое в качестве автора переводов. Если не задан, используется «анонимный пользователь».

--overwrite
Перезаписывать ли существующие переводы.

--inconsistent
Перезаписывать ли существующие переводы с непоследовательными переводами (смотрите проверку *Противоречия*).

--add
Автоматически добавлять язык, если указанный перевод не существует.

--mt MT
В качестве машинного перевода использовать машинный перевод, а не перевод из других компонентов.

--threshold THRESHOLD
Порог сходства для машинного перевода, по умолчанию установлен в 80.

Пример:

```
weblate auto_translate --user nijel --inconsistent --source weblate/application␣
↪weblate website cs
```

См.также:

Автоматический перевод

2.18.4 celery_queues

weblate celery_queues

Добавлено в версии 3.7.

Отображает длину очереди задач Celery.

2.18.5 checkgit

weblate checkgit <project|project/component>

Распечатывает текущее состояние внутреннего репозитория Git.

Вы можете либо задать конкретный проект или компонент для обновления (например, `weblate/application`), либо использовать ключ `--all` для обновления всех существующих компонентов.

2.18.6 commitgit

weblate commitgit <project|project/component>

Коммитит все возможные отложенные изменения во внутренний репозиторий Git.

Вы можете либо задать конкретный проект или компонент для обновления (например, `weblate/application`), либо использовать ключ `--all` для обновления всех существующих компонентов.

2.18.7 commit_pending

weblate commit_pending <project|project/component>

Коммитит отложенные изменения старше указанного возраста.

Вы можете либо задать конкретный проект или компонент для обновления (например, `weblate/application`), либо использовать ключ `--all` для обновления всех существующих компонентов.

--age `HOURS`
: Возраст в часах для коммита. Если не указан, то используется значение, заданное в *настройках компонента*.

Примечание: Она автоматически выполняется Weblate'ом в фоновом режиме, поэтому вручную вызывать эту команду нет никакой реальной необходимости, кроме разве что принудительного выполнения коммита в более раннее время, чем задано в *конфигурации компонента*.

См.также:

Выполнение задач технического обслуживания, `COMMIT_PENDING_HOURS`

2.18.8 cleanuptrans

weblate cleanuptrans

Очищает осиротевшие проверки и предложения перевода. Обычно запускать её вручную нет необходимости, поскольку очистка происходит автоматически в фоновом режиме.

См.также:

Выполнение задач технического обслуживания

2.18.9 createadmin

weblate createadmin

Создаёт учётную запись `admin` со случайным паролем, если он не указан явно.

--password `PASSWORD`
Взять пароль из командной строки, случайный пароль в этом случае не генерируется.

--no-password
Не устанавливать пароль, это может быть полезно при запуске с ключом `--update`.

--username `USERNAME`
Использовать указанное имя вместо `admin`.

--email `USER@EXAMPLE.COM`
Указывает адрес электронной почты администратора.

--name
Указывает (отображаемое) имя администратора.

--update
Обновление существующего пользователя (вы можете использовать этот ключ для изменения паролей).

Изменено в версии 2.9: Добавлены параметры `--username`, `--email`, `--name` и `--update`.

2.18.10 dump_memory

weblate dump_memory

Добавлено в версии 2.20.

Экспортирует JSON-файл, содержащий содержимое памяти переводов Weblate'а.

См.также:

Память переводов, *Схема памяти переводов Weblate'а*

2.18.11 dumpuserdata

weblate dumpuserdata <file.json>

Сбрасывает данные пользователя в файл для последующего их использования командой *importuserdata*

Подсказка: Она пригодится при переносе или слиянии экземпляров Weblate'а.

2.18.12 import_demo

weblate import_demo

Добавлено в версии 4.1.

Создаёт демонстрационный проект с компонентами на основе проекта <https://github.com/WeblateOrg/demo>.

Может быть полезна при разработке Weblate'а.

2.18.13 import_json

weblate import_json <json-file>

Добавлено в версии 2.7.

Пакетный импорт компонентов, основанных на данных из JSON.

Структура JSON импортируемого файла в значительной степени соответствует объекту компонента (смотрите *GET /api/components/(string:project)/(string:component)/*). Вы должны включить поля `name` и `filemask`.

--project `PROJECT`
Указывает, откуда будут импортироваться компоненты.

--main-component `COMPONENT`
Использовать указанный репозиторий системы контроля версий из этого компонента для всех остальных.

--ignore
Пропускать (уже) импортированные компоненты.

--update
Обновлять (уже) импортированные компоненты.

Изменено в версии 2.9: Параметры `--ignore` и `--update` существуют для работы с уже импортированными компонентами.

Пример JSON-файла:

```
[
  {
    "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"
  }
]
```

См.также:

import_memory

2.18.14 import_memory

weblate import_memory <file>

Добавлено в версии 2.20.

Импортирует файл TMX или JSON в память переводов Weblate'а.

--language-map `LANGMAP`
Позволяет отобразить языки из TMX на память переводов Weblate'а. Коды языков отображаются после выполняемой Weblate'ом нормализации.

Например, `--language-map ru_RU:ru` импортирует все строки с языком `ru_RU` как строки с языком `ru`.

Это может быть полезно в том случае, если локали в ваших файлах TMX не совпадают с теми, которые вы используете в Weblate'е.

См.также:

Память переводов, *Схема памяти переводов Weblate'а*

2.18.15 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

Изменено в версии 3.0: Команда import_project теперь основана на надстройке *Обнаружение компонентов*, что привело к некоторым изменениям в поведении и принимаемых параметрах.

Пакетный импорт компонентов в проект на основе маски файлов.

Параметр *<project>* определяет имя существующего проекта, в который будут импортированы компоненты.

Параметр *<gitrepo>* определяет URL-адрес используемого Git-репозитория, а *<branch>* — его ветку. Для импорта дополнительных компонентов перевода из существующего компонента Weblate'а используйте для параметра *<gitrepo>* адрес вида *weblate://<проект>/<компонент>*.

Параметр *<filemask>* определяет маску искомых файлов в репозитории. Он может быть либо простой строкой с подстановочными знаками, либо использовать всю мощь регулярных выражений.

Простое сопоставление использует `**` для имени компонента и `*` для языка, например: `**/*.po`

Регулярное выражение должно содержать группы с именами *component* и *language*. Например: `(?P<language>[^/]*)/(?P<component>[^-/]*)\.po`

Импорт сопоставляет найденые файлы с существующими компонентами на основе имён файлов перевода в них и добавляет новые компоненты, которые до этого не существовали. Уже существующие компоненты он не меняет.

--name-template TEMPLATE
Задаёт имя компонента, используя синтаксис шаблонов Django.

Например: `Документация: {{ component }}`

--base-file-template TEMPLATE
Задаёт базовый файл для одноязычных переводов.

Например: `{{ component }}/res/values/string.xml`

--new-base-template TEMPLATE
Задаёт базовый файл для добавления новых переводов.

Например: `{{ component }}/ts/en.ts`

--file-format FORMAT
Также вы можете указать формат файла (смотрите раздел *Поддерживаемые форматы файлов*), по умолчанию используется автоматическое определение.

--language-regex REGEX
С помощью этого параметра можно указать языковой фильтр (смотрите раздел *Настройки компонента*). Должен быть правильным регулярным выражением.

--main-component
Вы можете указать, какой компонент будет выбран в качестве основного — тот, который фактически содержит репозиторий системы контроля версий.

--license NAME
Указывает общую лицензию, лицензию проекта или компонента перевода.

--license-url URL
Указывает URL-адрес, по которому может быть найдена лицензия перевода.

--vcs NAME

В случае, если вам необходимо указать, какую использовать систему контроля версий, вы можете сделать это здесь. Система контроля версий по умолчанию — Git.

Чтобы привести вам несколько примеров, давайте попробуем импортировать два проекта.

Первый — перевод Руководства Debian, в котором для каждого языка имеется отдельный каталог с переводами каждой главы:

```
weblate import_project \
    debian-handbook \
    git://anonscm.debian.org/debian-handbook/debian-handbook.git \
    squeeze/master \
    '*/**.po'
```

Далее инструмент Tanaguru, для которого необходимо указать формат файла, шаблон базового файла, а также то, что все компоненты и переводы расположены в одном каталоге:

```
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
```

Более сложный пример разбора имён файлов для получения правильного компонента и языка из имени файла вида `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$'
```

Импорт переводов только на указанные языки:

```
./manage import_project \
    --language-regex '^(cs|sk)$' \
    weblate \
    https://github.com/WeblateOrg/weblate.git \
    'weblate/locale/*/LC_MESSAGES/**.po'
```

Импорт Sphinx-документации, разбитой на несколько файлов:

```
$ weblate import_project --name-template 'Documentation: %s' \
    --file-format po \
    project https://github.com/project/docs.git master \
    'docs/locale/*/LC_MESSAGES/**.po'
```

Импорт Sphinx-документации, разбитой на несколько файлов и каталогов:

```
$ 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'
```

См.также:

Более подробные примеры можно найти в главе starting, в качестве альтернативы вы можете использовать

команду *import_json*.

2.18.16 importuserdata

weblate importuserdata <file.json>

Импортирует данные пользователя из файла, созданного командой *dumpuserdata*

2.18.17 importusers

weblate importusers --check <file.json>

Импортирует пользователей из JSON-дампа Django базы данных auth_users.

--check
С этим параметром команда просто проверит, возможно ли импортировать данный файл, и сообщит о возможных конфликтах, связанных с именами пользователей или их адресами электронной почты.

Выгрузить пользователей из существующей установки Django вы можете при помощи команды:

```
weblate dumpdata auth.User > users.json
```

2.18.18 install_addon

Добавлено в версии 3.2.

weblate install_addon --addon ADDON <project|project/component>

Устанавливает надстройку для компонентов.

--addon ADDON
Названия устанавливаемого дополнения. Например, `weblate.gettext.customize`.

--configuration CONFIG
Настройки надстройки в виде JSON.

--update
Обновляет существующую конфигурацию надстройки.

Вы можете либо задать для какого конкретного компонента или проекта устанавливать надстройку (например, `weblate/application`), либо использовать ключ `--all`, чтобы установить её для всех компонентов.

Например, чтобы установить надстройку «*Настройка вывода gettext*» для всех компонентов, выполните:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --
↪update --all
```

См.также:

Надстройки

2.18.19 list_languages

weblate list_languages <locale>

Выводит список поддерживаемых языков в разметке MediaWiki: коды языков, английские названия и локализованные названия.

Используется для генерации словаря с названиями языков.

2.18.20 list_translators

weblate list_translators <project|project/component>

Показывает переводчиков по языкам проекта:

```
[French]
Jean Dupont <jean.dupont@example.com>
[English]
John Doe <jd@example.com>
```

--language-code
Сортировать список по кодам языков, а не по их названиям.

Вы можете либо задать конкретный проект или компонент (например, `weblate/application`), либо использовать ключ `--all`, чтобы получить список всех переводчиков всех существующих компонентов.

2.18.21 list_versions

weblate list_versions

Выводит список всех зависимостей weblate и их версии.

2.18.22 loadpo

weblate loadpo <project|project/component>

Перезагружает переводы с диска (например, если вы внесли обновления в свою систему контроля версий).

--force
Принудительно обновлять файлы, даже если эти файлы и так уже должны были бы быть в актуальном состоянии.

--lang `LANGUAGE`
Ограничится обработкой только заданного языка.

Вы можете либо задать конкретный проект или компонент для обновления (например, `weblate/application`), либо использовать ключ `--all` для обновления всех существующих компонентов.

Примечание: Вообще говоря, вам не часто придётся пользоваться данной командой: Weblate автоматически загружает изменившиеся файлы во время каждого обновления системы контроля версий. Однако, это может потребоваться, если вы вручную внесли изменения в репозиторий СКВ, управляемый Weblate, или в некоторых особых случаях после обновления.

2.18.23 lock_translation

weblate lock_translation <project|project/component>

Запрещает последующий перевод компонента.

Подсказка: Эта команда бывает полезна, когда вы хотите провести какое-нибудь тех. обслуживание нижележащего репозитория.

Вы можете либо задать конкретный проект или компонент для обновления (например, `weblate/application`), либо использовать ключ `--all` для обновления всех существующих компонентов.

См.также:

unlock_translation

2.18.24 move_language

weblate move_language source target

Добавлено в версии 3.0.

Позволяет вам объединить содержимое разных языков. Это может быть полезно при переходе на новую версию, в которой появились псевдонимы для языков, созданных вручную ранее, тем, которым был присвоен суффикс *(generated)*. Эта команда перемещает всё содержимое языка *source* в язык *target*.

Пример:

```
weblate move_language cze cs
```

После перемещения содержимого вам следует проверить, не осталось ли что-то не перенесённое (т.к. при этом могло возникнуть состояние гонки, когда кто-то одновременно с вами обновляет репозиторий), и удалить язык, помеченный как *(generated)*.

2.18.25 pushgit

weblate pushgit <project|project/component>

Отправляет закоммиченные изменения в вышестоящий репозиторий.

--force-commit

Принудительно закоммитить все ожидающие изменения перед тем как отправить их в вышестоящий репозиторий.

Вы можете либо задать конкретный проект или компонент для обновления (например, `weblate/application`), либо использовать ключ `--all` для обновления всех существующих компонентов.

Примечание: Weblate отправляет изменения автоматически, если *отправка при коммите* включена в *настройках компонента* (по умолчанию это так).

2.18.26 unlock_translation

weblate unlock_translation <project|project/component>

Разблокировывает данный компонент, делая его доступным для перевода.

Подсказка: Эта команда бывает полезна, когда вы хотите провести какое-нибудь тех. обслуживание нижележащего репозитория.

Вы можете либо задать конкретный проект или компонент для обновления (например, `weblate/application`), либо использовать ключ `--all` для обновления всех существующих компонентов.

См.также:

lock_translation

2.18.27 setupgroups

weblate setupgroups

Задаёт группу по умолчанию и, опционально, переносит всех пользователей в эту группу.

--no-privs-update
Отключает автообновление существующих групп (только добавляет новые).

--no-projects-update
Предотвращает автоматическое обновление групп уже существующих проектов. Это даёт возможность добавлять свежесозданные группы к существующим проектам, см. *Управление доступом к проекту*.

См.также:

Список привилегий

2.18.28 setuplang

weblate setuplang

Обновляет список определенных языков в Weblate.

--no-update
Отключает автообновление существующих языков (только добавляет новые).

2.18.29 updatechecks

weblate updatechecks <project|project/component>

Обновляет все проверки всех строк.

Подсказка: Полезна после обновлений Weblate, которые в значительной степени меняют проверки.

Вы можете либо задать конкретный проект или компонент для обновления (например, `weblate/application`), либо использовать ключ `--all` для обновления всех существующих компонентов.

2.18.30 updategit

`weblate updategit <project|project/component>`

Извлекает изменения из удалённого репозитория системы контроля версий в локальный и обновляет внутренний кеш.

Вы можете либо задать конкретный проект или компонент для обновления (например, `weblate/application`), либо использовать ключ `--all` для обновления всех существующих компонентов.

Примечание: Обычно лучше настроить *обработчики уведомлений* в репозитории вместо того, чтобы постоянно дёргать `updategit`.

2.19 Объявления

Изменено в версии 4.0: В предыдущих выпусках эта функция называлась «доской сообщений».

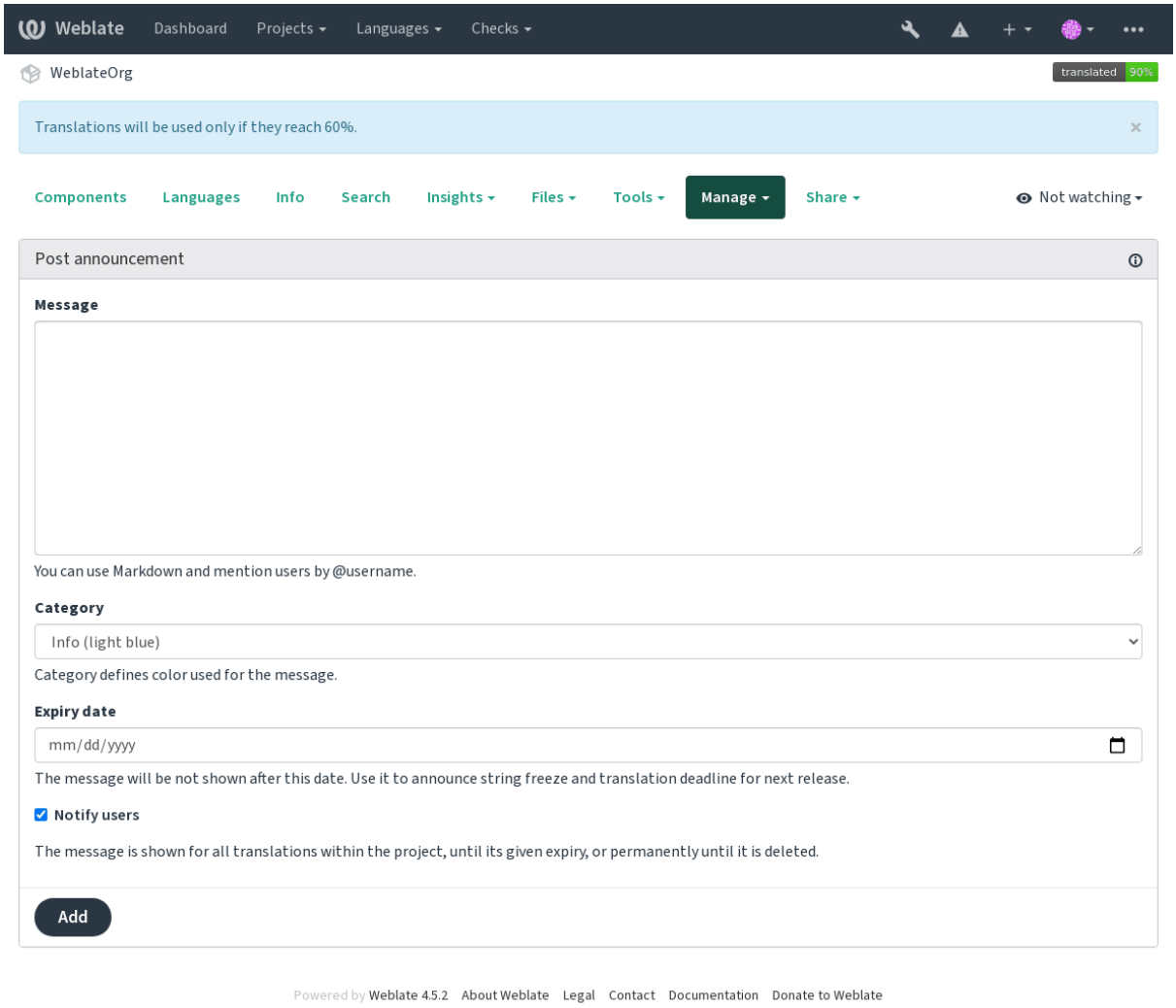
Предоставляет информацию вашим переводчикам путём размещения объявлений для всего сайта, проекта, компонента или языка.

Объявите цель, сроки, статус или укажите цели перевода.

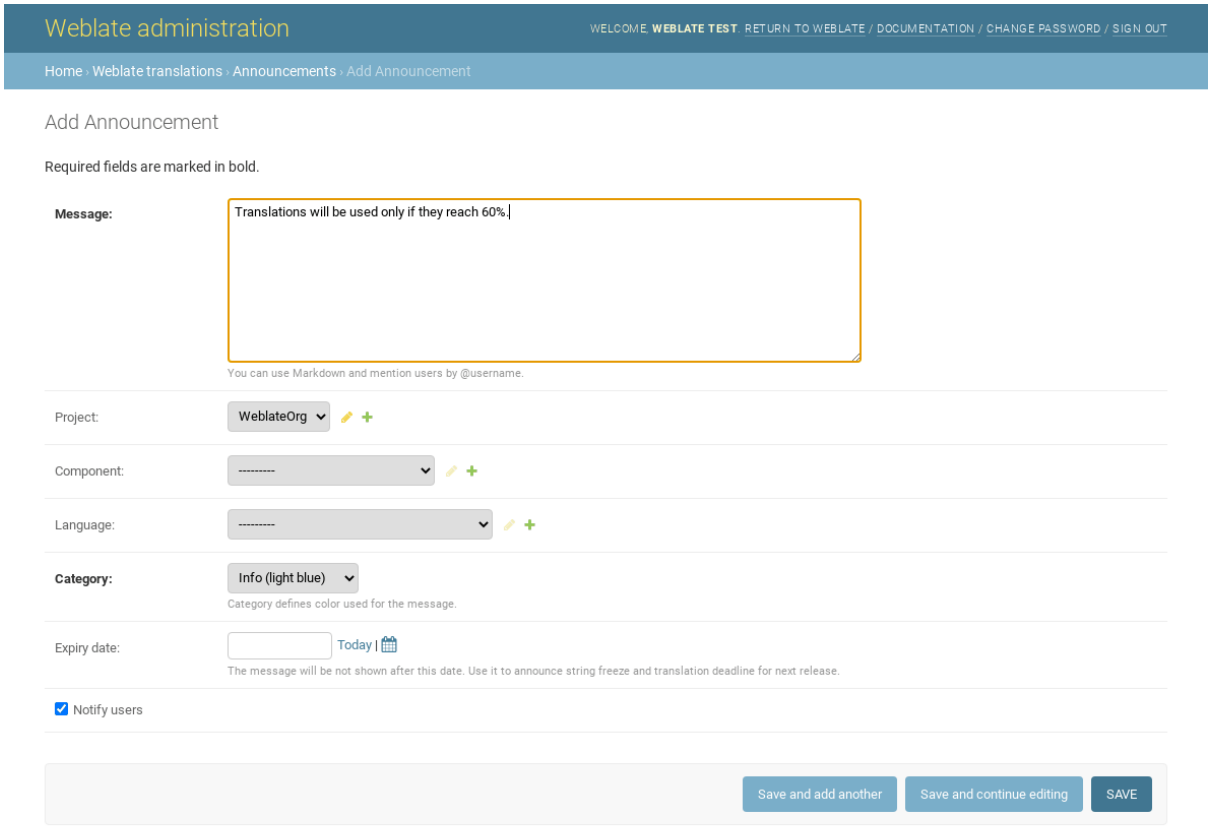
Пользователи будут получать уведомления об объявлениях в отслеживаемых ими проектах (если только они не откажутся от получения таких уведомлений).

Это может быть полезно для различных вещей, начиная от объявления намерений веб-сайта и заканчивая указанием целей для перевода.

Объявления могут быть размещены на каждом уровне через меню *Управлять* при помощи его пункта *Опубликовать объявление*:



Также его можно добавить через интерфейс администратора:



Объявления показываются в соответствии с заданным для них контекстом:

Контекст не указан

Показываются в панели управления (на странице входа).

Указан проект

Показываются внутри проекта, включая все его компоненты и переводы.

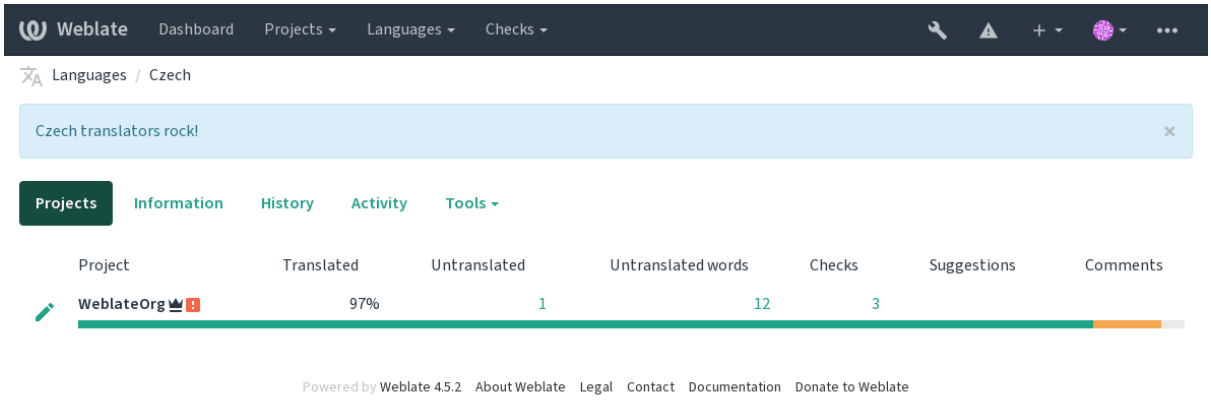
Указан компонент

Показываются для данного компонента и всех его переводов.

Указан язык

Показываются в обзоре по языку и во всех переводах на этот язык.

Вот как оно выглядит на странице обзора языка:



2.20 Списки компонентов

Укажите несколько списков компонентов, чтобы они появились в качестве вариантов выбора на панели управления пользователя, из которых пользователи смогут выбрать один из них, как своё умолчательное представление панели. Для получения более подробной информации смотрите раздел *Панель управления*.

Изменено в версии 2.20: Для каждого списка компонентов, присутствующего в панели управления, будет представлен свой статус.

Названия и содержимое списков компонентов можно указать в интерфейсе администратора в разделе *Списки компонентов*. Каждый список компонентов должен иметь название, отображаемое пользователю, и плашку, представляющую его в URL-адресе.

Изменено в версии 2.13: Для анонимных пользователей настройки панели управления изменяются из интерфейса администратора, в котором определяется, какая панель управления будет показана не аутентифицированным пользователям.

2.20.1 Автоматические списки компонентов

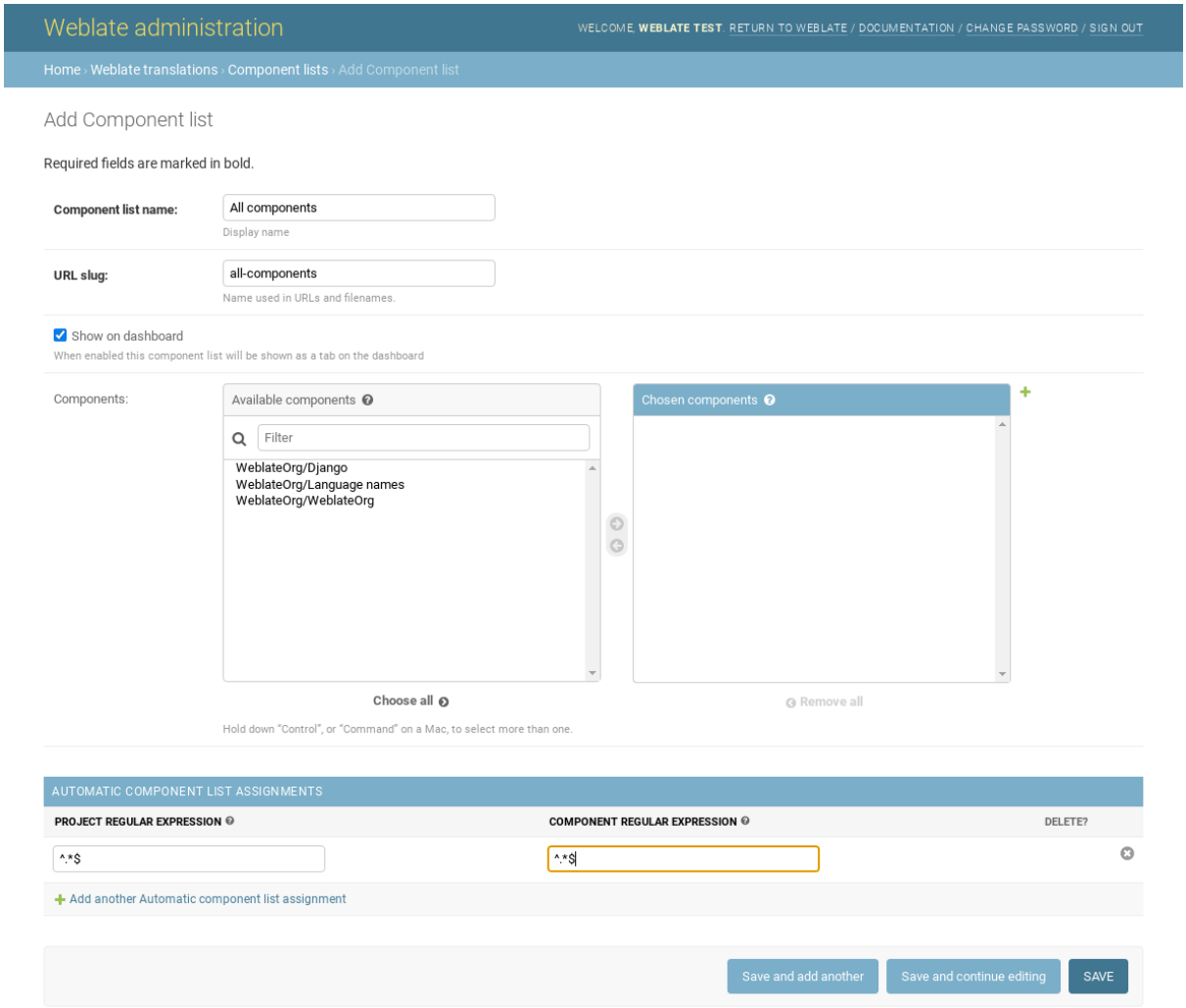
Добавлено в версии 2.13.

Автоматическое добавление компонентов в список на основе их плашки настраивается созданием правил *Автоматического прикрепления списка компонентов*.

- Полезно для ведения списков компонентов для крупных установок, либо в случае, если вы хотите иметь в своей установке Weblate один список компонентов со всеми компонентами.

Подсказка: Создайте список компонентов, содержащий все компоненты вашей установки Weblate.

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 Дополнительные модули Weblate

Для вашей установки доступно несколько дополнительных модулей.

2.21.1 Экспортер Git

Добавлено в версии 2.10.

Предоставляет доступ на чтение к нижележащему репозиторию Git через HTTP(S).

Установка

1. Добавьте `weblate.gitexport` в установленные приложения в файле `settings.py`:

```
INSTALLED_APPS += ("weblate.gitexport",)
```

2. Экспортируйте существующие репозитории, выполнив перено своей базы данных после установки:

```
weblate migrate
```

Использование

The module automatically hooks into Weblate and sets the exported repository URL in the *Настройки компонента*. The repositories are accessible under the `/git/` part of the Weblate URL, for example `https://example.org/git/weblate/main/`.

Репозитории для проектов, доступных публично, могут быть склонированы без аутентификации:

```
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/'
```

Подсказка: По умолчанию у членов группы *Пользователи* и анонимных пользователей есть доступ к репозиториям публичных проектов через роли *Доступ к репозиторию* и *Опытный пользователь*.

2.21.2 Тарифный план

Добавлено в версии 2.4.

Этот модуль используется на Облачном хостинге Weblate, для задания тарифных планов, а также отслеживания платежей и лимитов.

Установка

Добавьте `weblate.billing` в установленные приложения в файле `settings.py`:

```
INSTALLED_APPS += ("weblate.billing",)
```

2. Запустите миграцию базы данных для необязательной установки для модуля дополнительных структур базы данных:

```
weblate migrate
```

Использование

После установки вы сможете в интерфейсе администратора управлять тарификацией. У пользователей с включённой функцией тарификации в их *профиле* появится новая вкладка *Тарифный план*.

Также модуль тарифных планов даёт возможность администраторам проектов создавать новые проекты и компоненты без прав суперпользователя (см. *Добавление проектов и компонентов перевода*). Это становится возможным, если удовлетворены следующие условия:

- Тариф не исчерпал заданные лимиты (любое избыточное использование ресурсов влечёт блокировку создания проектов/компонентов) и оплачен (если он не бесплатный)
- Пользователь является администратором существующего тарифицируемого проекта или владельцем тарифного плана (последнее необходимо пользователям только подписавшимся на тариф для того, чтобы они могли загрузить новый проект).

При создании проекта пользователь может выбрать, к какому тарифу проект должен относиться, если у пользователя их несколько.

2.21.3 Правовые вопросы

Добавлено в версии 2.15.

Этот модуль используется на Облачном хостинге Weblate для вывода необходимых юридических документов. Он поставляется в комплекте с набором незаполненных документов. Предполагается, что вы самостоятельно заполните следующие шаблоны:

`legal/documents/tos.html` Документ условий обслуживания

`legal/documents/privacy.html` Документ политики конфиденциальности

`legal/documents/summary.html` Краткий обзор условий обслуживания и политики конфиденциальности

Примечание: Legal documents for the Hosted Weblate service are available in this Git repository <https://github.com/WeblateOrg/wllegal/tree/main/wllegal/templates/legal/documents>.

Скорей всего, вы не сможете просто «взять и использовать» их, так как они вам не подойдут. Однако, вы можете скорректировать их под свои нужды или они могут послужить отправной точкой при написании собственных документов.

Установка

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. Запустите миграцию базы данных для необязательной установки для модуля дополнительных структур базы данных:

```
weblate migrate
```

3. Отредактируйте юридические документы в каталоге `weblate/legal/templates/legal/` в соответствии с вашими условиями.

Использование

После установки и редактирования юридические документы будут отображаться в пользовательском интерфейсе Weblate.

2.21.4 Аватары

Аватары загружаются и кэшируются на стороне сервера, чтобы уменьшить утечку информации на сайты, обслуживающие их по умолчанию. Встроенную поддержку получения аватаров из настроенных для этого адресов электронной почты можно отключить с помощью параметра *ENABLE_AVATARS*.

В настоящее время Weblate поддерживает:

- Gravatar
- Libravatar

См.также:

Кеширование аватаров, *AVATAR_URL_PREFIX*, *ENABLE_AVATARS*

2.21.5 Защита от спама

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. Obtain the Akismet API key.
3. Store it as *AKISMET_API_KEY* or *WEBLATE_AKISMET_API_KEY* in Docker.

Following content is sent to Akismet for checking:

- Suggestions from unauthenticated users
- Project and component descriptions and links

Примечание: Защита (среди всего прочего) зависит от IP-адреса клиента, для правильной её настройки, пожалуйста, смотрите раздел *Работа за обратным прокси*.

См.также:

Работа за обратным прокси, *AKISMET_API_KEY*, *WEBLATE_AKISMET_API_KEY*

2.21.6 Подписание коммитов Git с помощью GnuPG

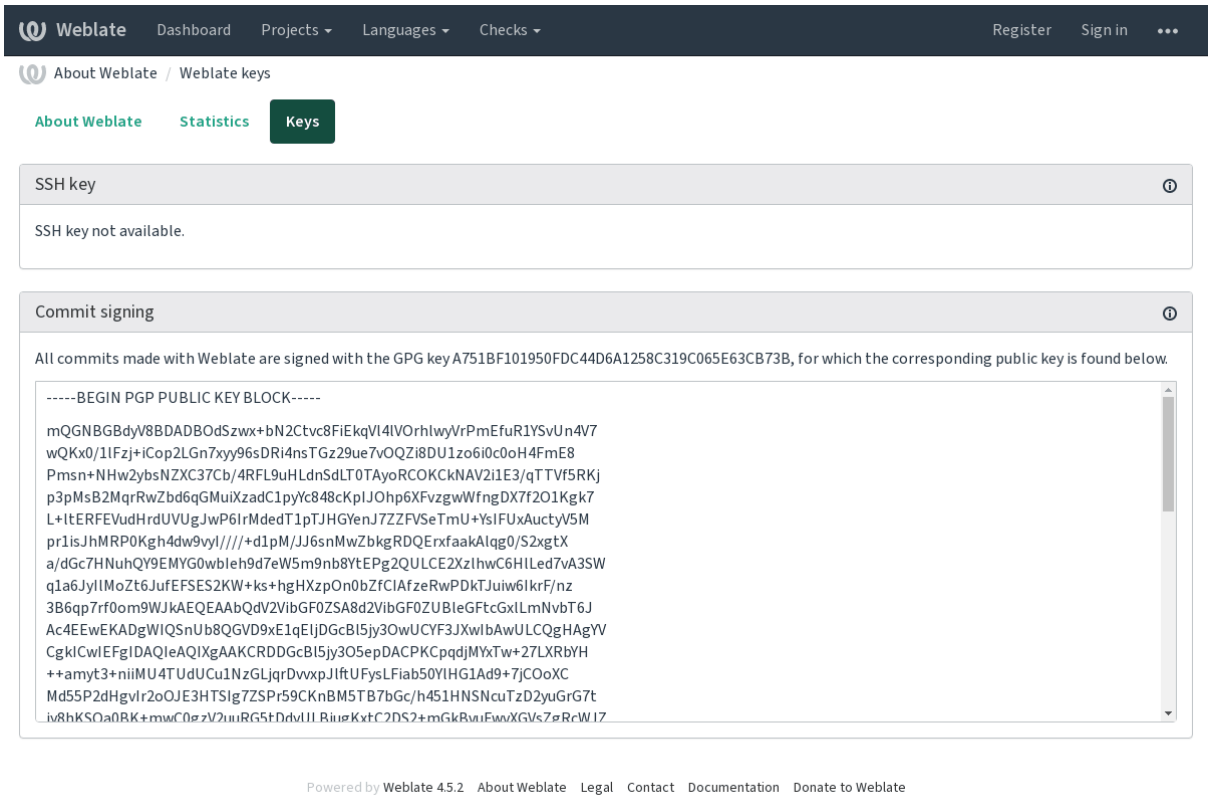
Добавлено в версии 3.1.

Все коммиты могут быть подписаны с помощью ключа GnuPG вашего экземпляра Weblate.

1. Turn on *WEBLATE_GPG_IDENTITY*. (Weblate will generate a GnuPG key when needed and will use it to sign all translation commits.)

Эта функция требует установки GnuPG 2.1 или новее.

Этот ключ будет хранится по пути, указанному в *DATA_DIR*, а открытый ключ будет отображаться на странице «О Weblate» (*/about/*):



2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking gpg.

См.также:

WEBLATE_GPG_IDENTITY

2.21.7 Ограничение частоты запросов

Изменено в версии 3.2: Модуль «Ограничение частоты запросов» теперь допускает более тонкую настройку.

Некоторые операции Weblate ограничены по частоте. Допустимо не более *RATELIMIT_ATTEMPTS* попыток за период *RATELIMIT_WINDOW* секунд. Если это количество превышено, то пользователь блокируется на *RATELIMIT_LOCKOUT* секунд. Существуют также параметры, задающие ограничения, специфические для конкретных действий, например `RATELIMIT_CONTACT_ATTEMPTS` или `RATELIMIT_TRANSLATE_ATTEMPTS`. Следующая таблица является исчерпывающим списком таких параметров.

Следующие операции подвержены ограничению частоты запросов:

Имя	Область	Допустимо попыток	Временное окно	Период блокировки
Регистрация	`REGISTRATION`	5	300	600
Отправка сообщения администраторам	`MESSAGE`	5	300	600
Аутентификация по паролю при входе	`LOGIN`	5	300	600
Поиск по всему сайту	`SEARCH`	6	60	60
Перевод	`TRANSLATE`	30	60	600
Добавление в глоссарий	`GLOSSARY`	30	60	600
Начало перевода на новый язык	`LANGUAGE`	2	300	600

Если у пользователя не получается залогинится с `AUTH_LOCK_ATTEMPTS` попыток, то аутентификация по паролю полностью отключается до тех пор пока пользователь не сбросит свой пароль.

Для API есть отдельные настройки ограничения колличества запросов, см. *Ограничение частоты запросов API*.

См.также:

Ограничение частоты запросов, *Работа за обратным прокси*, *Ограничение частоты запросов API*

2.21.8 Fedora Messaging integration

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 Настройка Weblate

Расширяйте и настраивайте Weblate, используя Django и Python. Вносите свои изменения в основной репозиторий Weblate'а, чтобы выгоду из них мог извлечь каждый. Кроме того, это снизит ваши затраты на поддержку этих изменений: в дальнейшем при изменении внутренних интерфейсов или рефакторинге поддержкой актуальности кода, включённого в Weblate, будут заниматься его сопровождающие.

> **Предупреждение:** Ни внутренние интерфейсы, ни шаблоны не считаются стабильным API. При каждом обновлении просматривайте внесённые вами изменения: интерфейсы и их семантика могут измениться без предварительного уведомления.

См.также:

Как внести свой вклад в Weblate

2.22.1 Создание модуля Python

Если вы не знакомы с Python'ом, возможно, вы захотите посмотреть статью Python для начинающих, в которой объясняются основы и указаны дальнейшие учебные пособия.

При написании собственного кода на Python (называемого модулем) его необходимо разместить в каталоге, находящемся либо где-то в системном пути (обычно это путь вида `/usr/lib/python3.7/site-packages/`), либо в каталоге Weblate'а, который также добавляется к путям поиска интерпретатора.

А ещё лучше превратить ваши модификации в полноценный пакет Python:

1. Создайте каталог для вашего пакета (мы будем использовать *weblate_customization*).
2. В нём создайте файл `setup.py` с описанием своего пакета:

```
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. Создайте каталог для модуля Python (также назовём его `weblate_customization`), в котором будет содержаться ваш код.
4. В нём создайте файл `__init__.py`, чтобы Python смог импортировать этот модуль.
5. Теперь этот пакет можно установить командой **`pip install -e`**. Более подробную информацию можно найти в документе «Редактируемые» установки.
6. После установки этот модуль можно использовать в настройках Weblate'а (например, `weblate_customization.checks.FooCheck`).

Структура вашего модуля должна выглядеть примерно так:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

Пример настройки Weblate'а можно найти по адресу <https://github.com/WeblateOrg/customize-example>, он охватывает все описанные ниже темы.

2.22.2 Изменение логотипа

1. Создайте простое приложение Django, содержащее статические файлы, которые вы хотите перезаписать (смотрите раздел *Создание модуля Python*).

   Стилизуемые элементы расположены в следующих файлах:

   `icons/weblate.svg` Логотип, отображаемый на панели навигации.

   `logo-*.png` Веб-иконки для разных разрешений экрана и веб-браузеров.

   `favicon.ico` Веб-иконка, используемая устаревшими браузерами.

   `weblate-*.png` Аватары ботов и анонимных пользователей. Некоторые веб-браузеры используют их также в качестве картинок для ярлыков на стартовой странице.

email-logo.png Используется в уведомлениях по электронной почте.

2. Добавьте его в `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    # Add your customization as first
    "weblate_customization",
    # Weblate apps are here…
)
```

3. Запустите `weblate collectstatic --noinput`, чтобы собрать статические файлы, предназначенные для передачи клиентам.

См.также:

Управление статическими файлами (изображениями, JavaScript, CSS и прочими подобными), *Обслуживание статических файлов*

2.22.3 Пользовательские проверки качества, надстройки и автоматические исправления

Для добавления в Weblate собственного кода для *автоматических исправлений*, *проверок* или *надстроек*:

1. Поместите файлы в модуль Python, содержащий вашу модификацию Weblate'а (смотрите раздел *Создание модуля Python*).
2. Добавьте полное имя его класса Python в список соответствующего параметра (*`WEBLATE_ADDONS`*, *`CHECK_LIST`* или *`AUTOFIX_LIST`*):

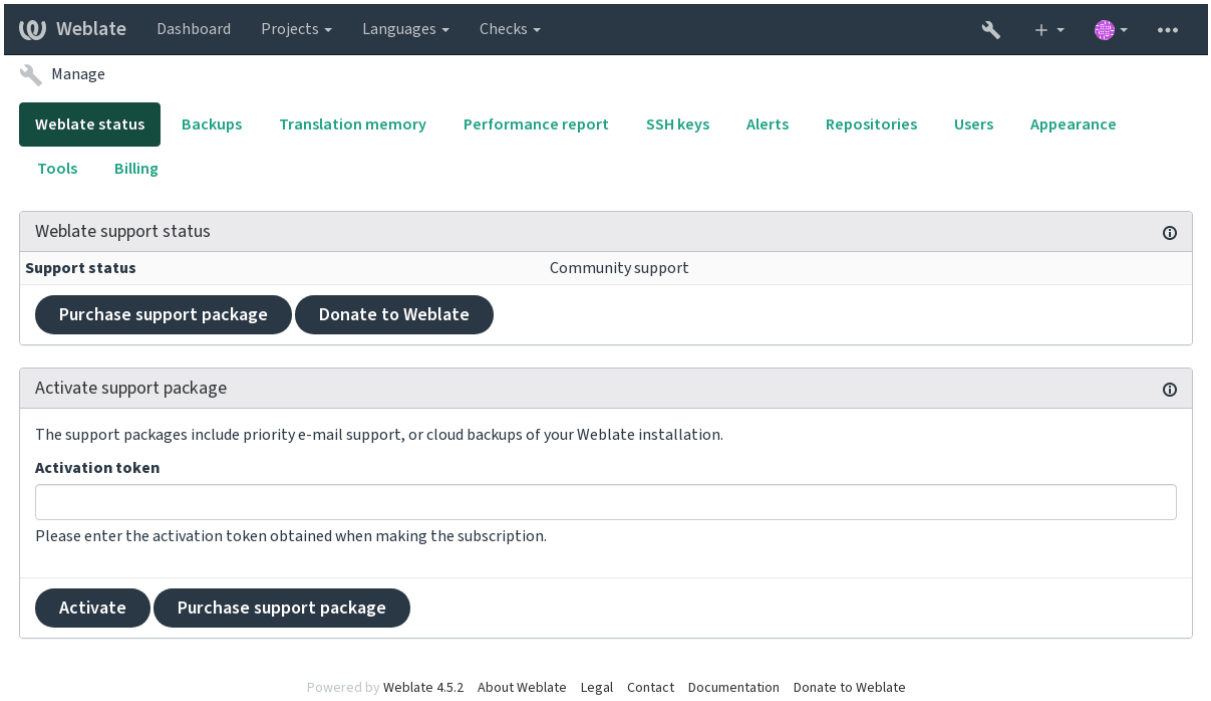
```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

# Addons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

См.также:

Пользовательские автоматические исправления, *Написание собственных проверок*, *Создание надстройки*, *Выполнение скриптов из надстройки*

2.23 Интерфейс управления

Интерфейс управления даёт доступ к настройкам для администраторов. Он расположен по URL-адресу `/manage/`. Доступ к нему может получить только пользователь, вошедший в систему с правами администратора, через иконку гаечного ключа справа вверху:



Он включает базовый обзор вашего Weblate:

- Статус поддержки, см. *Получение поддержки по Weblate*
- Резервные копии, см. *Резервное копирование и перенос Weblate*
- Совместная память переводов, см. *Память переводов*
- Отчёты о производительности, чтобы оценить состояние здоровья Weblate и длину очередей Celery
- Управление SSH-ключами, см. *Репозитории по SSH*
- Обзор предупреждений по всем компонентам, см. alerts

2.23.1 Интерфейс администратора Django

> **Предупреждение:** В будущем будет удалён, поскольку его использование не рекомендуется — большинством возможностей можно управлять непосредственно из Weblate'а.

Здесь вы можете управлять объектами, хранящимися в базе данных, такими как пользователи или переводы, а также другими настройками:

Weblate administration

WELCOME, **WEBLATE TEST**. RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT

Site administration

REPORTS
Weblate support status
Status of repositories
SSH keys
Performance report
Translation memory

ACCOUNTS		
Audit logs	+ Add	Change
Profiles	+ Add	Change
Verified emails	+ Add	Change

AUTH TOKEN		
Tokens	+ Add	Change

AUTHENTICATION		
Groups	+ Add	Change
Roles	+ Add	Change
Users	+ Add	Change

BILLING		
Billings	+ Add	Change
Invoices	+ Add	Change
Plans	+ Add	Change

FONTS		
Font groups	+ Add	Change
Fonts	+ Add	Change

LEGAL		
Agreements	+ Add	Change

PYTHON SOCIAL AUTH		
Associations	+ Add	Change
Nonces	+ Add	Change
User social auths	+ Add	Change

SCREENSHOTS		
Screenshots	+ Add	Change

TRANSLATION MEMORY		
Memorys	+ Add	Change

WEBLATE CONFIGURATION		
Settings	+ Add	Change

WEBLATE LANGUAGES		
Languages	+ Add	Change

WEBLATE TRANSLATIONS		
Announcements	+ Add	Change
Component lists	+ Add	Change
Components	+ Add	Change
Contributor agreements	+ Add	Change
Projects	+ Add	Change

Recent actions

My actions

None available

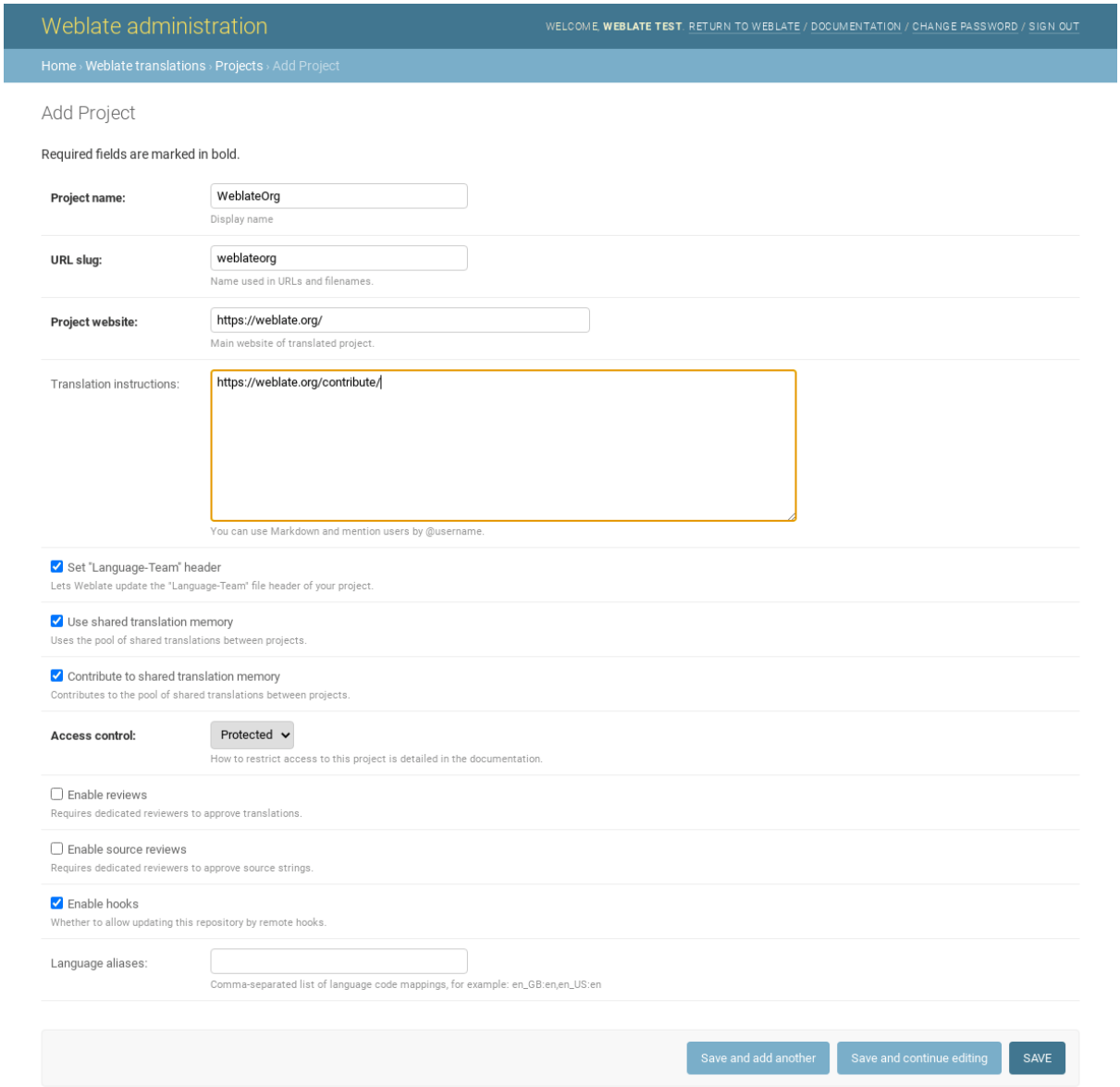
В разделе *Отчёты* вы можете проверить статус вашего сайта, настроить его для *рабочей среды*, или управлять ключами SSH, используемыми для *доступа к репозиториям*.

Управление объектами базы данных под любым разделом. Самым интересным из них является, наверное, *Переводы Weblate'а*, в котором можно управлять переводимыми проектами, смотрите разделы *Настройки проекта* и *Настройки компонента*.

Раздел *Языки Weblate* содержит определения языков, более подробное объяснение смотрите в разделе *Определение языков*.

Добавление проекта

Добавление проекта служит контейнером для всех компонентов. Обычно вы создаёте один проект для одной части программного обеспечения или книги (для справки по параметрам смотрите раздел *Настройки проекта*):



См.также:

Настройки проекта

Двуязычные компоненты

После добавления проекта в него можно добавить компоненты перевода (для справки по параметрам смотрите раздел *Настройки компонента*):

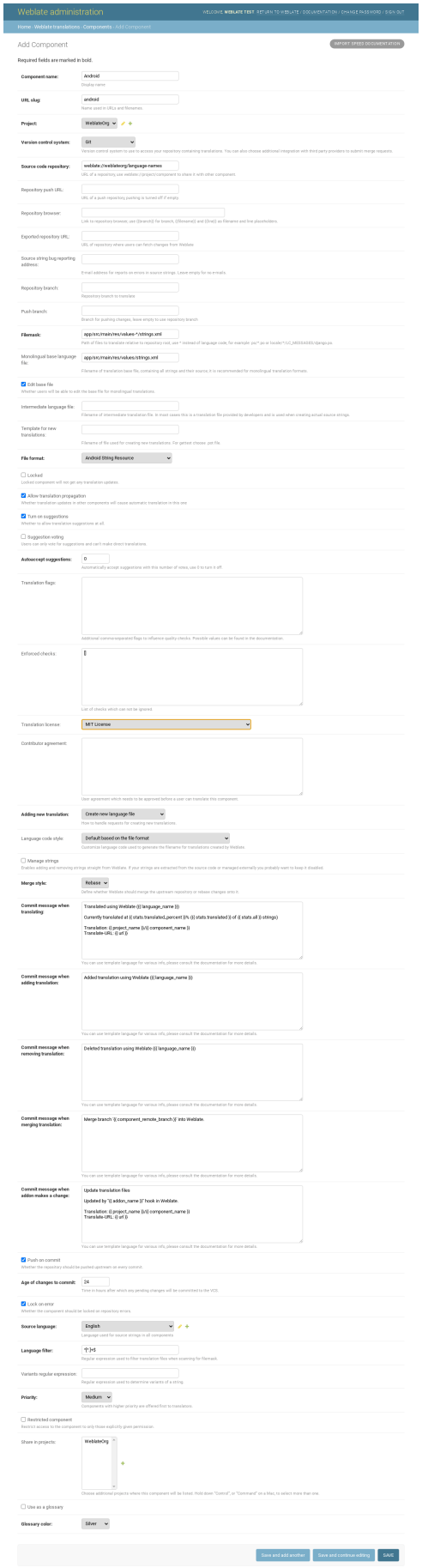
Weblate administration
WELCOME WEBLATE TEST RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Home Weblate translations Components Add Component
Add Component
IMPORT SPEED DOCUMENTATION
Required fields are marked in bold.
Component name:
Language names
Display name
URL slug:
language-names
Name used in URLs and filenames.
Project:
WeblateOrg
Version control system:
Git
Version control system to use to access your repository 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 push URL:
URL of a push repository, pushing is turned off if empty.
Repository browser:
https://github.com/WeblateOrg/demo/blob/{branch}/{filename}{URL}
Link to repository browser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders.
Exported repository URL:
URL of repository where users can fetch changes from Weblate
Source string bug reporting address:
E-mail 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
Filemask:
weblate/langdata/locale/*/LC_MESSAGES/djs
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:
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.
File format:
gettext PO file
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 quality checks. Possible values can be found in the documentation.
Enforced checks:
[]
List of checks which can not be ignored.
Translation license:
GNU General Public License v3.0 or later
Contributor agreement:
User agreement which needs to be approved before a user can translate this component.
Adding new translation:
Create new language file
How to handle requests for creating new translations.
Language code style:
Default based on the file format
Customize language code used to generate the filename for translations created by Weblate.
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 addon 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.
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:
^(cs|fwd|hu|zl)
Regular expression used to filter translation files when scanning for filemask.
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
Save and add another
Save and continue editing
SAVE

См.также:

Настройки компонента, *Двуязычные и одноязычные форматы*

Одноязычные компоненты

Для облегчения их перевода предоставьте шаблон файла, содержащий отображение идентификаторов сообщений на соответствующие строки на языке оригинала (обычно это английский) (для справки по параметрам смотрите раздел *Настройки компонента*):


Weblate administration
WELCOME, WEBLATE TEST. RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Home › Weblate translations › Components › Add Component
Add Component
IMPORT SPEED DOCUMENTATION
Required fields are marked in bold.
Component name:
Display name
URL slug:
Name used in URLs and filenames.
Project:
Version control system:
Source code repository:
Repository push URL:
Repository browser:
Exported repository URL:
Source string bug reporting address:
Repository branch:
Push branch:
Filemask:
Monolingual base language file:
Edit base file
Intermediate language file:
Template for new translations:
File format:
Locked
Allow translation propagation
Turn on suggestions
Suggestion voting
Autoaccept suggestions:
Translation flags:
Enforced checks:
Translation license:
Contributor agreement:
Adding new translation:
Language code style:
Manage strings
Merge style:
Commit message when translating:
Commit message when adding translation:
Commit message when removing translation:
Commit message when merging translation:
Commit message when addon makes a change:
Push on commit
Age of changes to commit:
Lock on error
Source language:
Language filter:
Variants regular expression:
Priority:
Restricted component
Share in projects:
Use as a glossary
Glossary color:
Save and add another
Save and continue editing
SAVE

См.также:

Настройки компонента, *Двуязычные и одноязычные форматы*

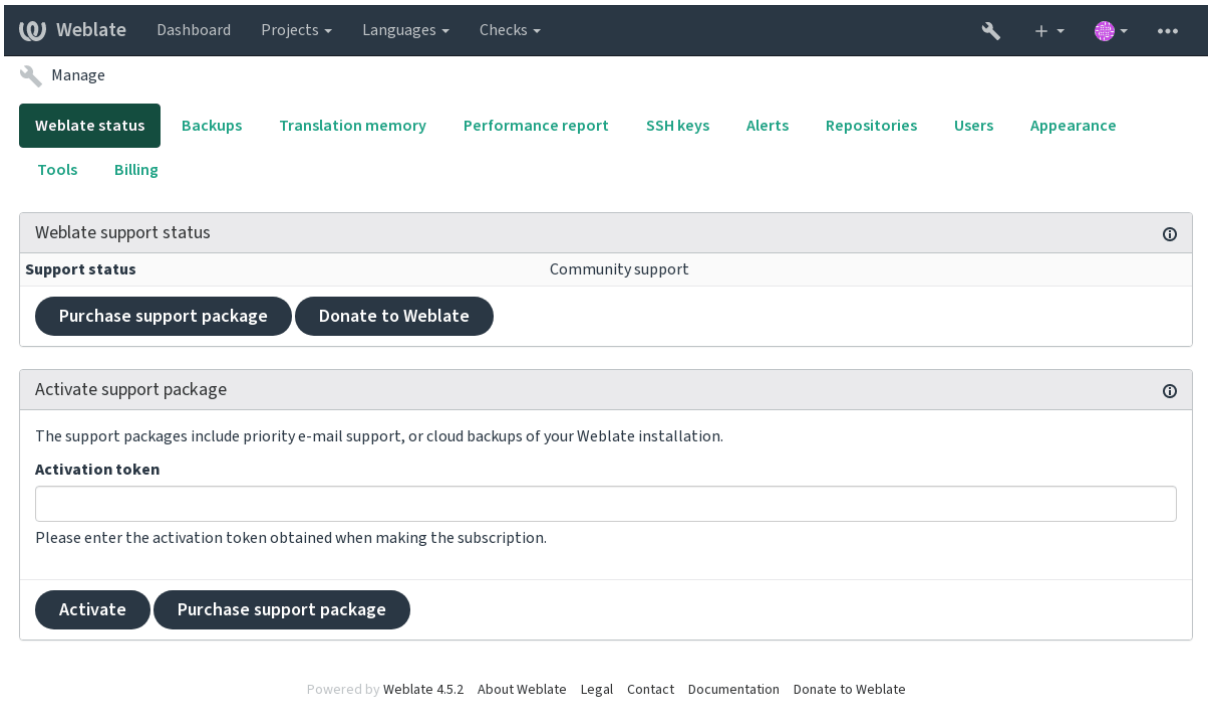
2.24 Получение поддержки по Weblate

Weblate — это свободное программное обеспечение с поддержкой сообщества, выпускаемое на условиях авторского лева. Подписчики получают приоритетную поддержку без дополнительной оплаты. Предоплаченные пакеты помощи доступны для всех. Более подробную информацию о текущих предложениях поддержки вы можете найти по адресу <https://weblate.org/support/>.

2.24.1 Интегрированная поддержка

Добавлено в версии 3.8.

Приобретённый пакет платной поддержки может быть, по вашему желанию, интегрирован в ваш интерфейс управления подпиской Weblate, где вы найдёте на него ссылку. В этом случае основные сведенья о вашей установке будут переданы Weblate.



2.24.2 Данные передаваемые Weblate

- URL, на котором настроен ваш экземпляр Weblate
- Заголовок вашего сайта
- Ваша версия Weblate
- Колличество некоторых объектов в вашей базе данных Wablate (проектов, компонентов, языков, исходных строк и пользователей)
- Открытый SSH-ключ

Additionally, when *Discover Weblate* is turned on:

- List of public projects (name, URL and website)

Ни какая другая информация не передаётся.

2.24.3 Доступные сервисы при включении интеграции

- Просмотр, действителен ли ещё ваш пакет поддержки
- *Предоставляемое Weblate'ом хранилище резервных копий*
- *Discover Weblate*

Подсказка: Платные пакеты поддержки уже активирован сразу после приобретения. Их можно использовать и без интеграции.

2.24.4 Discover Weblate

Добавлено в версии 4.5.2.

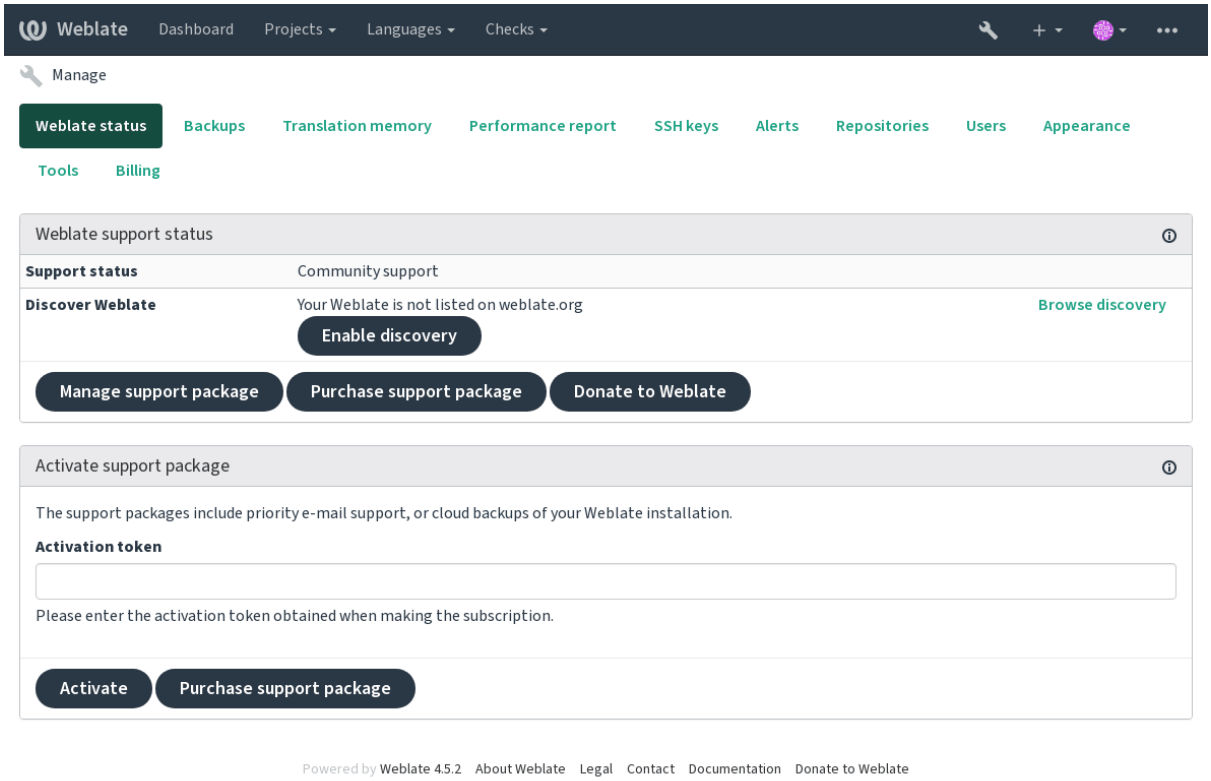
Примечание: This feature is currently in early beta.

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.

Getting listed

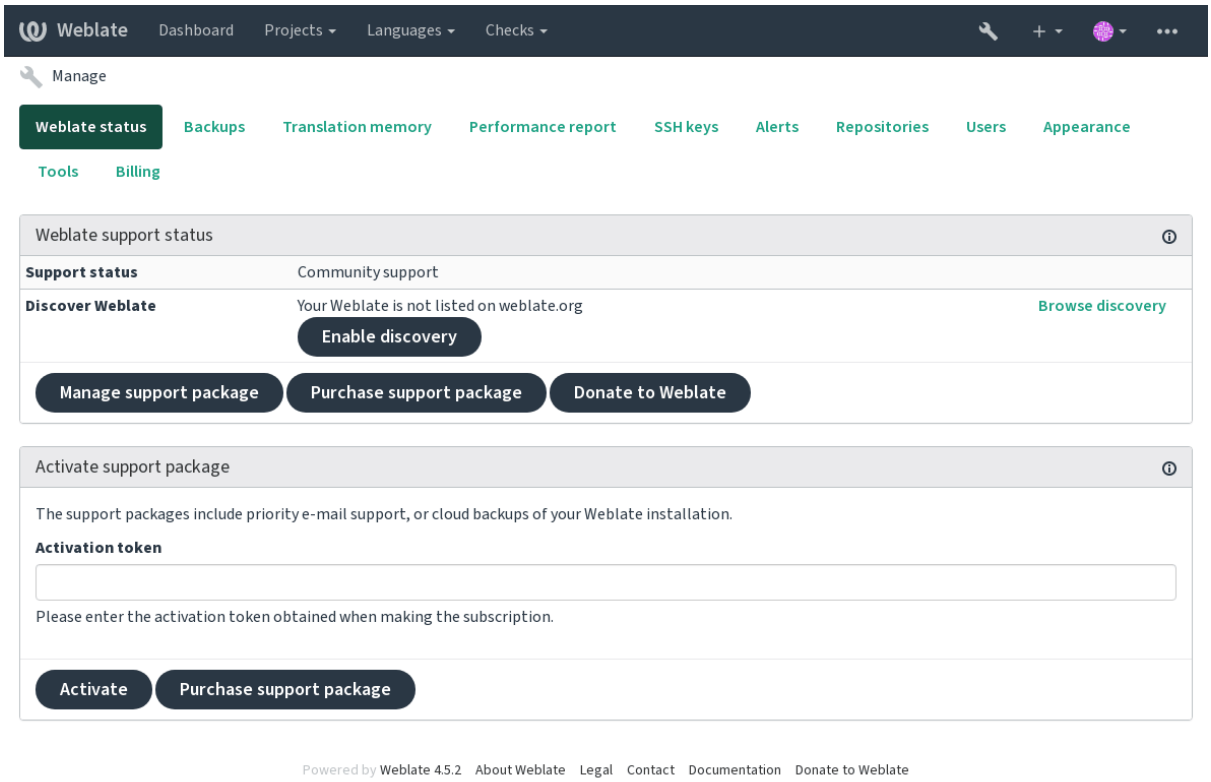
Подсказка: Participating in Discover Weblate makes Weblate submit some information about your server, please see *Данные передаваемые Weblate*.

To list your server with an active support subscription (see *Интегрированная поддержка*) in Discover Weblate all you need to do is turn this on in the management panel:



Listing your server without a support subsription in Discover Weblate:

1. Register yourself at <https://weblate.org/user/>
2. Register your Weblate server in the discovery database at <https://weblate.org/subscription/discovery/>
3. Confirm the service activation in your Weblate and turn on the discovery listing in your Weblate management page using *Enable discovery* button:



Customizing listing

You can customize the listing by providing a text and image (570 x 260 pixels) at <https://weblate.org/user/>.

2.25 Правовые документы

Примечание: Здесь вы найдёте информацию по различным правовым вопросам, которая может вам потребоваться при работе с Weblate в определённых юрисдикциях. Она приведена исключительно в качестве рекомендации без каких-либо гарантий точности или корректности оной. В конце концов, обеспечение того, чтобы то, как вы используете Weblate, соответствовало всем применимым законам и нормам, является исключительно вышей ответственностью.

2.25.1 ITAR и другие меры контроля экспорта

Weblate может быть запущен в вашем собственном датацентре или виртуальном частном облаке. В этом случае, он также может быть использован для хранения информации, подпадающей под ограничения, наложенные ITAR (*International Traffic in Arms Regulations*, *Правилами международной торговли оборонной продукцией* США) или какими-либо другими правилами; однако, в таком случае ответственность за соответствие всей информации данным правилам несут сами конечные пользователи.

Облачный хостинг Weblate не проверялся на соответствие требованиям ITAR или каким-либо другим правилам контроля экспорта, а также на данный момент не предоставляет каких-либо возможностей для ограничения доступа переводчиков в зависимости от их страны.

2.25.2 Ограничения на шифрование в США

Weblate сам по себе не содержит какой-либо криптографический код, хотя и может быть предметом ограничений на шифрование, так как он использует другие сторонние компоненты, использующие криптографию для аутентификации, а также контроля целостности и конфиденциальности данных.

Скорей всего Weblate был бы классифицирован, как ECCN (*Export Control Classification Number*, *Номер классификации контролируемого экспорта*) *5D002* или *5D992* и, так как он является публично-доступным свободным ПО, он не должен подпадать под действие EAR (*Export Administration Regulations*, *административные ограничения на экспорт*, см. Элементы шифрования НЕ подпадающие под действие EAR).

Программные компоненты, используемые Weblate (перечислены только компоненты связанные с криптографическим функционалом):

Python Смотрите https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

GnuPG Используется Weblate в качестве опции

Git Используется Weblate в качестве опции

curl Используется Git

OpenSSL Используется Python и cURL

Стойкость ключей шифрования зависит от настроек Weblate и сторонних компонентов, с которыми он взаимодействует, но в любой приличной конфигурации он будет включать различную функциональность, на которую накладываются ограничения связанные с криптографией:

- Симметричные алгоритмы с ключом превышающем 56 бит
- Факторизация целых чисел превышающих 512 бит (в асимметричных алгоритмах)
- Вычисление дискретных логарифмов в мультипликативной группе конечного поля размерностью больше 512 бит для асимметричных алгоритмов

- Дискретные логарифмы в группах отличных от вышеперечисленных и превышающие 112 бит для асимметричных алгоритмов

В Weblate нет ни каких механизмов дезактивации криптографии, но он может быть настроен таким образом, чтобы никакой криптографический код не вовлекался. Криптографические функции включают:

- Доступ к удалённым серверам по безопасным протоколам (HTTPS)
- Создание PGP подписей для коммитов

См.также:

Ограничения экспорта (EAR) свободного ПО

ГЛАВА 3

Руководство участника

3.1 Как внести свой вклад в 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:

- *Сообщение о проблемах в Weblate'е*
- *Начало участия в разработке Weblate*
- *Перевод Weblate*
- *Contribute to Weblate documentation*
- *Weblate discussions*
- *Финансирование разработки Weblate*

3.1.1 Перевод Weblate

Weblate is continually being translated using Weblate itself. Feel free to take your part in the effort of making Weblate available in as many human languages as possible. It brings Weblate closer to its users!

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 Contribute to Weblate documentation

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 affraid 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.

Подсказка: You can translate the docs.

3.1.3 Weblate discussions

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.

3.1.4 Финансирование разработки 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.

Те, кто уже оказал финансовую поддержку Weblate

Список спонсоров Weblate:

- Yashiro Ccs
- Cheng-Chia Tseng
- Timon Reinhard
- Cassidy James
- Loic Dachary
- Marozed
- https://freedombox.org/
- GNU Solidario (GNU Health)
- BallotReady
- Richard Nespithal

Хотите оказаться в этом списке? Тогда посмотрите свои опции на странице «Пожертвовать для Weblate».

3.2 Начало участия в разработке Weblate

Understand the Weblate source code by going through *Исходный код Weblate*, *Фронтэнд Weblate* and *Внутреннее устройство Weblate*.

3.2.1 Starting with the codebase

Familiarize yourself with the Weblate codebase, by having a go at the bugs labelled good first issue.

3.2.2 Запуск Weblate локально

Самый удобный подход к разработке Weblate — это создать собственное виртуальное окружение (*virtualenv*) для него с редактируемой версией исходного кода, как описано в *Установка из исходников*.

1. Clone the Weblate source code:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Create a virtualenv:

```
virtualenv .venv
.venv/bin/activate
```

3. Install Weblate (for this you need some system dependencies, see *Установка из исходников*):

```
pip install -e .
```

3. Установите все зависимости, полезные для разработки:

```
pip install -r requirements-dev.txt
```

4. Запустите сервер разработки:

```
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 *Запуск тестов локально* for more details):

```
. scripts/test-database
./manage.py test
```

См.также:

Установка из исходников

3.2.3 Запуск Weblate локально внутри 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 *Добавление проектов и компонентов перевода*.

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:

```
./rundev.sh test --failfast weblate.trans
```

Примечание: Убедитесь, что ваш контейнер Docker запущен и работает до того как запустить тесты. Вы можете проверить это, выполнив `docker ps`.

Чтобы вывести логи:

```
./rundev.sh logs
```

To stop the background containers, run:

```
./rundev.sh stop
```

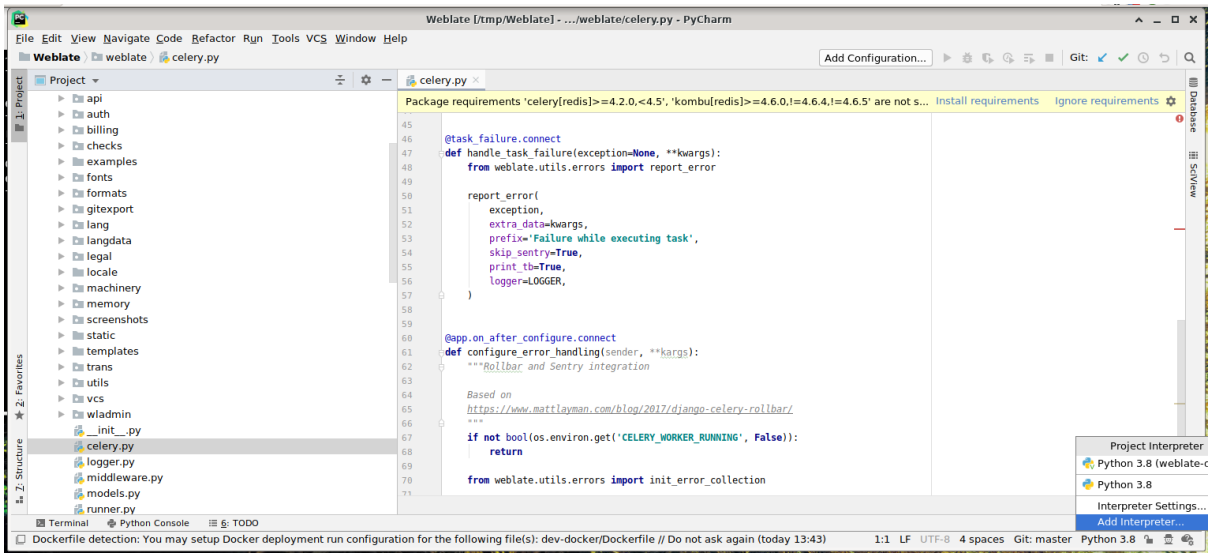
Running the script without arguments will re-create the Docker container and restart it.

Примечание: This is not a suitable setup for production, as it includes several hacks which are insecure, but they make development easier.

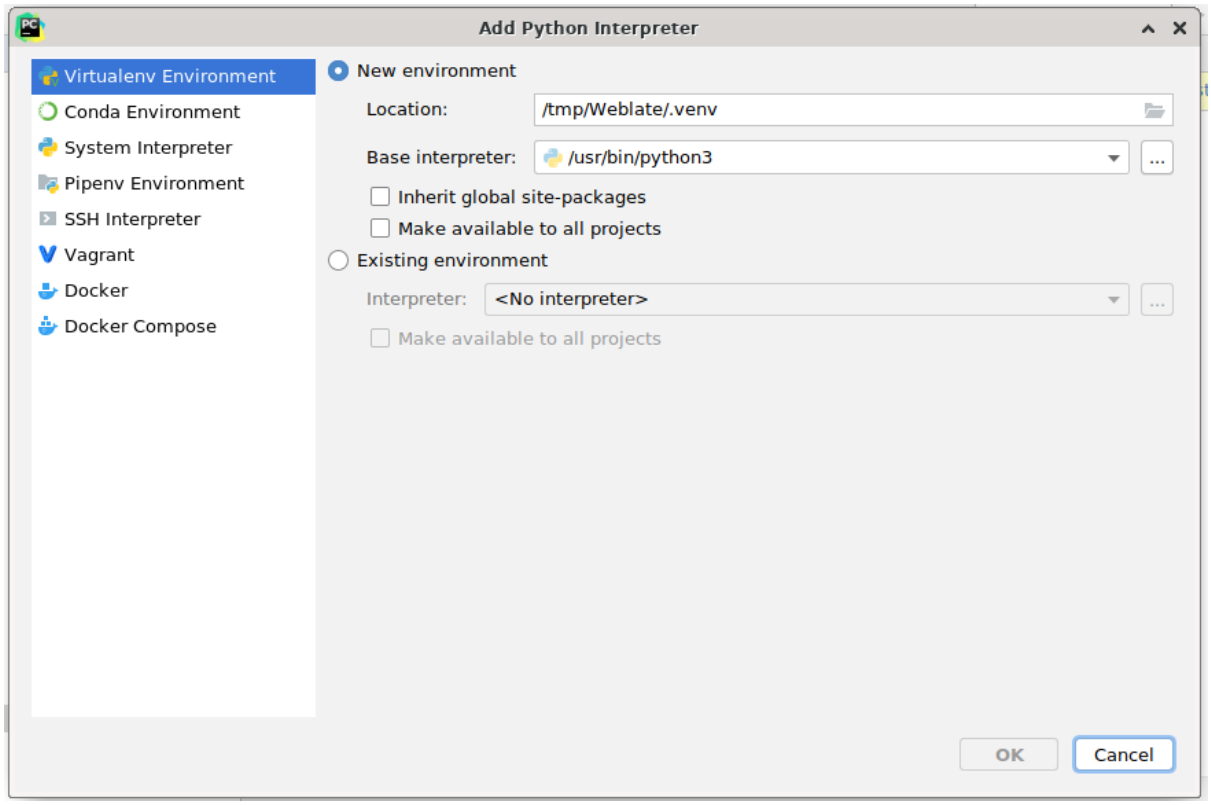
3.2.4 Разработка Weblate с помощью PyCharm

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:

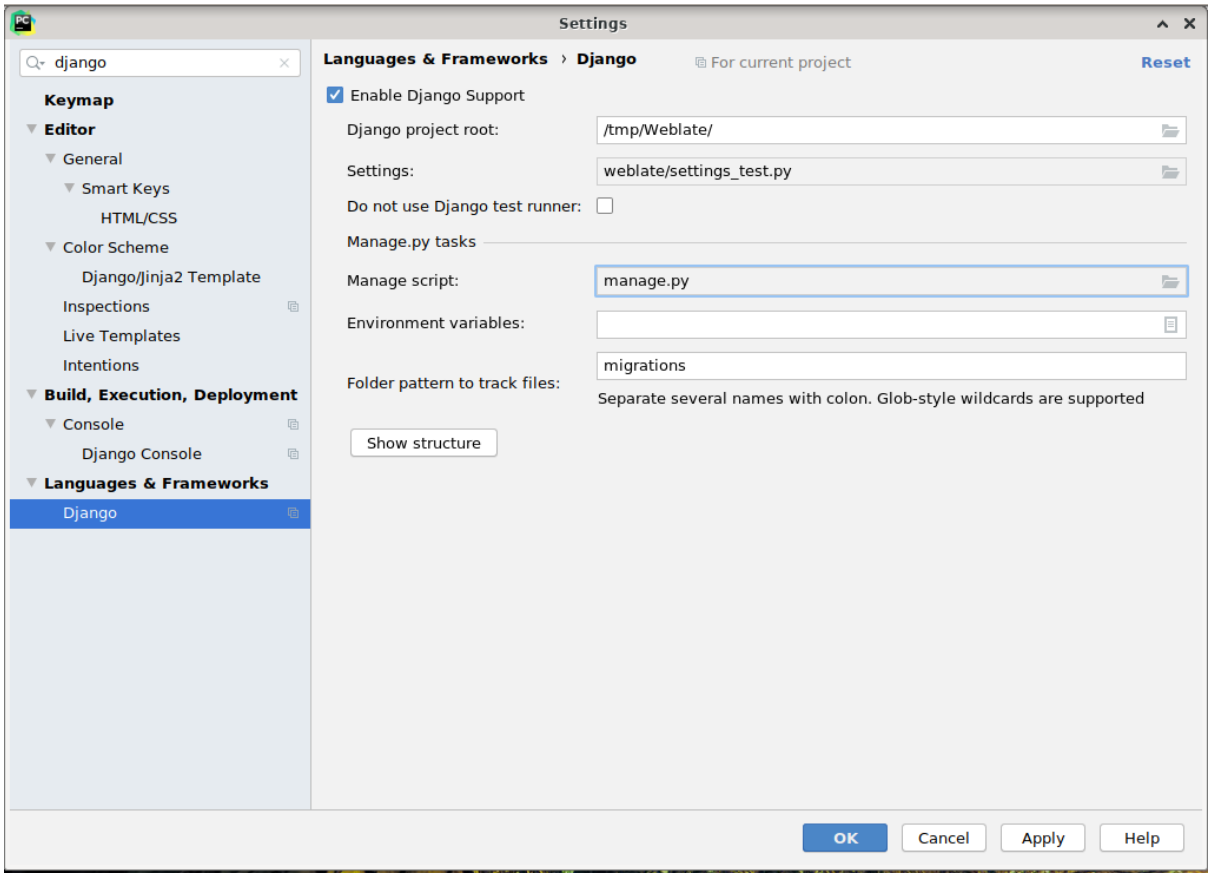


Вы можете либо позволить PyCharm создать виртуальное окружение за вас, либо выбрать уже существующее:



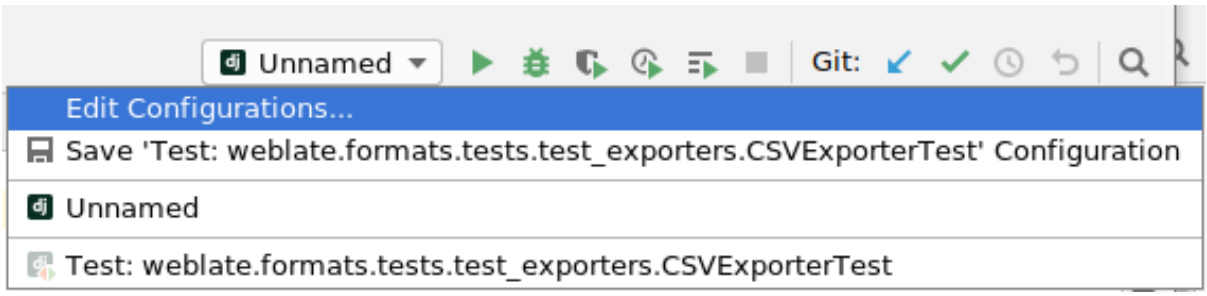
Don't forget to install the dependencies once the interpreter is set: Either through the console (the console from the IDE will directly use your virtualenv by default), or through the interface when you get a warning about missing dependencies.

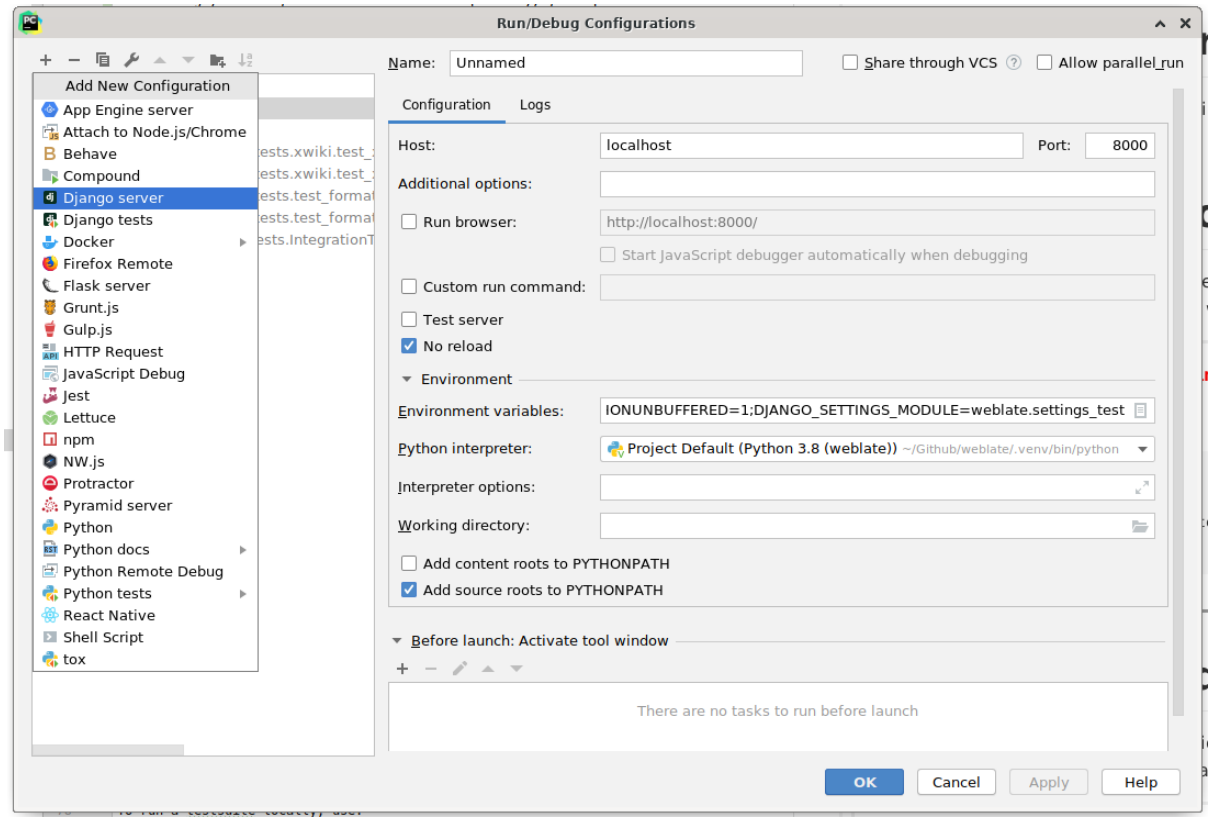
The second step is to set the right info to use Django natively inside PyCharm: The idea is to be able to immediately trigger the unit tests in the IDE. For that you need to specify the root path of the Django project and the path to its settings:



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:





Подсказка: 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 Исходный код Weblate

Weblate разрабатывается на GitHub'е. Вы можете создавать форки и открывать запросы на извлечение. Патчи в любой другой форме также приветствуются.

См.также:

Чтобы понять, как Weblate устроен изнутри, посмотрите раздел *Внутреннее устройство Weblate*.

3.3.1 Принципы «безопасности по проектированию»

Любой код для Weblate'а должен писаться с оглядкой на *Принципы «безопасности по проектированию»* (*Security by Design*).

3.3.2 Соглашения по оформлению кода

Код должен соответствовать рекомендациям по кодированию PEP-8 и должен быть отформатирован с использованием инструмента форматирования кода **black**.

Для проверки качества кода можно использовать **flake8**, рекомендуемые плагины перечислены в :file:``.pre-commit-config.yaml`, а их настройки находятся в `setup.cfg`.

Самый простой способ принудить себя к выполнению всех этих соглашений — установить pre-commit. Репозиторий Weblate содержит настройки для него, в соответствии с которыми он будет проверять, что закоммиченные файлы минимально вменяемы. После его установки (он уже есть в списке зависимостей в `requirements-lint.txt`) включите его, запустив `pre-commit install` в каталоге с рабочей копией Weblate. Таким образом все ваши изменения будут проверяться автоматически.

Также вы можете запустить эту проверку вручную, чтобы проверить все файлы, запустите:

```
pre-commit run --all
```

3.4 Отладка Weblate

Баги могут проявлять себя в том, что приложение падает или ведёт себя некорректно. Мы будем рады, если вы соберёте информацию о таких проблемах и передадите её в нашу систему отслеживания проблем.

3.4.1 Режим отладки

Включение режима отладки приведёт к тому, что исключения будут отображаться прямо в браузере. Это полезно при отладке проблем в веб-интерфейсе, но для рабочей системы применять данный режим не следует, поскольку он отражается на производительности не лучшим образом и может привести к утечке конфиденциальных данных.

См.также:

Отключение отладочного режима

3.4.2 Журналы Weblate

Weblate может вести детализированные журналы того, что происходит в фоне. С настройками по умолчанию он использует syslog в результате чего файл журнала появляется в `/var/log/messages` или в `/var/log/syslog` (в зависимости от настроек вашего демона syslog).

Процесс Celery (см. *Фоновые задачи с использованием Celery*) обычно также ведёт свои собственные журналы. Скажем, в примере с установкой Celery в качестве системного сервиса журналы будут записываться в несколько файлов в каталоге `/var/log/celery/`.

Контейнеры Docker записывают журналы в свой поток вывода (как это обычно и бывает в Docker-мире), так что вы можете просмотреть эти журналы с помощью `docker-compose logs`.

См.также:

`LOGGING` в *примере файла настроек*.

3.4.3 Не обрабатываются фоновые задачи

Многие задачи выполняются в фоновых обработчиках Celery. Если такие вещи, как отправка электронной почты или удаление компонентов не работают, то, вероятно, с этими обработчиками какие-то проблемы.

В таком случае следует:

- Проверить, что Celery запущен, см. *Фоновые задачи с использованием Celery*
- Проверить статус очереди Celery или через *интерфейс управления* или с помощью `celery_queues`
- Просмотреть журналы Celery на наличие ошибок (см. *Журналы Weblate*)

3.4.4 Не доходит электронная почта от Weblate

Проверить, правильно ли работает исходящая почта. Это можно сделать с помощью команды управления `sendtestemail` (подробности, как вызывать её в различных окружениях, смотрите в разделе «*Вызов команд управления*») или с помощью *интерфейса управления* на вкладке *Инструменты*.

Это действие отправит почту напрямую, так что таким образом можно проверить, что SMTP настроен корректно (см. *Настройка исходящей почты*). Однако большую часть почты Weblate отправляет в фоновых задачах, так что всё это может быть вызвано некоторыми проблемами с Celery. Как отладить Celery смотрите в *соответствующем разделе*.

3.4.5 Анализ падений программы

Если приложение падает, то будет полезно собрать об этом падении столько информации, сколько возможно. Самый простой сделать это — использовать сторонние сервисы, которые собирают эту информацию автоматически. Вы можете почитать о том, как это сделать, в разделе «*Сбор отчётов об ошибках*».

3.4.6 Ошибки без оповещения

Многие задачи выгружаются в Celery для фоновой обработки. Их падения в пользовательском интерфейсе не отображаются, а появляются только в журналах Celery. Настройка *сбора ошибок* поможет вам легче замечать такие сбои.

3.4.7 Проблемы с производительностью

Если в каких-то ситуациях Weblate показывает недостаточную производительность, то соберите журналы, которые демонстрируют проблему, а также всё, что поможет выяснить, как и в каком месте можно улучшить код.

Если некоторые запросы выполняются слишком долго без какой-либо индикации, то вы можете установить dogslow и использовать его вместе со *сбором отчётов об ошибках*, что позволит получить точечные и подробные трассировки стеков вызова (traceback) в этих отчётах.

3.5 Внутреннее устройство Weblate

Примечание: Эта глава даст вам базовое представление о внутреннем устройстве Weblate.

Большую часть структуры своего кода Weblate наследует от Django, так как основывается на оном.

3.5.1 Структура каталогов

Быстрый обзор структуры каталогов главного репозитория Weblate:

docs Исходный код данной документации, который можно собрать с помощью Sphinx.

dev-docker Код Docker для запуска сервера разработки, см. *Запуск Weblate локально внутри Docker*.

weblate Исходный код Weblate (точнее самого Django-приложения), см. *Внутреннее устройство Weblate*.

weblate/static Клиентские файлы (CSS, Javascript, изображения), см. *Фронтэнд Weblate*.

3.5.2 Модули

Weblate состоит из нескольких Django-приложений (некоторые из них опциональны, см. *Дополнительные модули Weblate*):

`accounts`

Аккаунты пользователей, профили и уведомления.

`addons`

Надстройки для изменения поведения Weblate, см. *addons*.

`api`

API на основе Django REST framework.

`auth`

Аутентификация и разрешения.

`billing`

Дополнительный модуль *Тарифный план*.

`checks`

Модуль *Проверки качества* строки перевода.

`fonts`

Модуль проверки отрисовки шрифтов.

`formats`

Слой абстракции от форматов файлов на основе translate-toolkit.

`gitexport`

Дополнительный модуль *Экспортер Git*.

`lang`

Модуль определения языка и моделей множественного числа.

`legal`

Дополнительный модуль *Правовые вопросы*.

`machinery`

Интеграция сервисов машинного перевода.

`memory`

Встроенная память переводов, смотрите раздел *Память переводов*.

`screenshots`

Модуль управления снимками экрана и распознаванием текста.

`trans`

Основной модуль обработки переводов.

`utils`

Различные вспомогательные утилиты.

`vcs`

Абстракция системы контроля версий.

`wladmin`

Настройка интерфейса администра Django.

3.6 Разработка надстроек

Надстройки — это способ внести изменений в ваш рабочий процесс локализации с помощью Weblate.

class `weblate.addons.base.`**BaseAddon**(*storage=None*)

classmethod can_install(*component*, *user*)
Проверяет, совместима ли надстройка с указанным компонентом.

configure(*settings*)
Сохранение конфигурации.

daily(*component*)
Обработчики, которые будут вызываться ежедневно.

classmethod get_add_form(*user*, *component*, ***kwargs*)
Возвращает форму с пользовательскими настройками. Вызывается при добавлении новой надстройки к компоненту.

get_settings_form(*user*, ***kwargs*)
Возвращает форму с пользовательскими настройками этой надстройки.

post_add(*translation*)
Обработчик, вызываемый после добавления нового перевода.

post_commit(*component*)
Обработчик, вызываемый после того, как изменения будут закоммиченны в репозиторий.

post_push(*component*)
Обработчик, вызываемый после того как изменения в будут отправлены в вышестоящий репозиторий.

post_update(*component*, *previous_head: str*, *skip_push: bool*)
Обработчик, вызываемый после того, как локальный репозиторий будет обновлён из вышестоящего.

Параметры

- **previous_head** (`str`) – Текущее состояние репозитория (HEAD) перед обновлением; может быть пустым, если этот репозиторий только что склонирован.

- **skip_push** (*bool*) – Определяет, должна ли надстройка пропустить собственно отправку изменений в вышестоящий репозиторий. Обычно этот параметр просто передаётся в конкретные методы реализации, как commit_and_push или commit_pending.

pre_commit(*translation*, *author*)
Обработчик, вызываемый перед тем, как изменения будут закоммиченны в репозиторий.

pre_push(*component*)
Обработчик, вызываемый перед тем как изменения в будут отправлены в вышестоящий репозиторий.

pre_update(*component*)
Обработчик, вызываемый перед тем, как локальный репозиторий будет обновлён из вышестоящего.

save_state()
Сохранить информацию о состоянии надстройки.

stay_on_create = False
Базовый класс для всех надстроек Weblate.

store_post_load(*translation*, *store*)
Обработчик, вызываемый после разбора файла.

В качестве аргумента ему передаётся объект класса формата файла.

Это может быть полезно для модификации параметров класса формата файла, например, изменить , как именно данный файл будет сохраняться.

unit_pre_create(*unit*)
Обработчик, вызываемый перед созданием новой единицы перевода.

Вот пример надстройки:

```
#
# Copyright © 2012 - 2021 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"}}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    # List of events addon should receive
    events = (EVENT_PRE_COMMIT,)
    # Addon unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example addon")
    # Detailed addon description
    description = _("This addon does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

3.7 Фронтэнд Weblate

Фронтэнд в данный момент написан на Bootstrap, jQuery и паре других сторонних библиотек.

3.7.1 Поддерживаемые браузеры

Weblate поддерживает последние стабильные релизы всех основных браузеров и платформ.

Альтернативные браузеры, которые используют последние версии WebKit, Blink, или Gecko (не важно, напрямую ли или опосредованно через API платформы) в явном виде не поддерживаются. Однако Weblate должен (в большинстве случаев) отображаться и функционировать корректно и в них.

Более старые браузеры, хотя и могут всё ещё работать, но некоторый функционал в них может быть ограничен.

3.7.2 Управление зависимостями

Для обновления сторонних библиотек используется менеджер пакетов yarn. Его настройки находятся в файле `scripts/yarn`, а для обновления, сборки и копирования библиотек в корректное место в каталоге `weblate/static/vendor`, где располагаются все сторонние библиотеки фронтэнда, есть скрипт-обёртка `scripts/yarn-update`.

Добавление новой сторонней библиотеки обычно заключается в:

```
# 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 Оформление кода

Weblate полагается на Prettier как для форматирования кода JavaScript, так и CSS-файлов.

Мы также используем ESLint для проверки кода на JavaScript.

3.7.4 Локализация

Если вам надо добавить какой-либо текст, который будет показываться пользователю, то он должен быть локализуемым. В большинстве случаев будет достаточно просто обернуть свой текст в вызов `gettext`, но для более сложных случаев есть и более интересные возможности:

```
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'
```

См.также:

Раздел про перевод в документации Django

3.7.5 Иконки

В данный момент Weblate использует иконки в стиле Material Design. Если вам нужны какие-то дополнительные эмблемки, то обратите внимание на Material Design Icons или на Material Design Resources.

В дополнение к этому есть скрипт `scripts/optimize-svg`, который уменьшает размер этих SVG-иконок, так как большинство из них встраиваются внутрь HTML, чтобы возможно было задавать стиль контурам.

3.8 Сообщение о проблемах в Weblate'е

Weblate issue tracker is hosted at GitHub.

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 *Проблемы безопасности* section below.

If you are not sure about your bug report or feature request, you can try *Weblate discussions*.

3.8.1 Проблемы безопасности

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.

В качестве альтернативы вы можете сообщать о проблемах на security@weblate.org. Эти сообщения в итоге также окажутся на HackerOne.

Если вы по каким-либо причинам не хотите использовать HackerOne, то вы можете сообщить об ошибке по электронной почте на michal@cihar.com. Вы также можете зашифровать письмо с помощью PGP-ключа *3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D*. Этот ключ также доступен в Keybase.

Примечание: 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.

Некоторые из них:

- Django
- Django REST framework
- Python Social Auth

3.9 Система автоматического тестирования и непрерывной интеграции Weblate

Тестами покрыта большая часть современного кода. Увеличьте это покрытия, добавляя новые тесты при добавлении какой-либо новой функциональности, дабы проверить, что она работает корректно.

3.9.1 Непрерывная интеграция

Текущие результаты тестов можно найти в GitHub Actions, а отчёты о проценте покрытия тестами на Codecov.

Есть несколько задач, которые проверяют разные аспекты:

- Модульное тестирование
- Сборка и внешние ссылки в документации
- Проверка миграции со всех поддерживаемых релизов
- Проверка исходного кода (линтинг)
- Проверка установки (гарантирует, что в сгенерированных файлах всё есть и они могут быть протестированы)

Настройки для системы непрерывной интеграции расположены в каталоге `.github/workflows`. Она полагается на скрипты, хранящиеся в каталоге `ci`. Скрипты можно также выполнить вручную, но для этого нужно установить несколько переменных среды (в основном связанных с соединением с базой данных, те, что определены в файле настроек Django). Пример, как определить эти переменные, находится в `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
```

Простой запуск может выглядеть как:

```
. scripts/test-database
./ci/run-migrate
./ci/run-test
./ci/run-docs
```

3.9.2 Запуск тестов локально

Чтобы запустить тесты локально, выполните:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

Подсказка: Вам потребуется сервер базы данных (PostgreSQL), который будет использоваться для тестов. По умолчанию Django создаёт отдельную базу данных для запуска тестов (с префиксом `test_`). Например, если в ваших настройках используется база данных `weblate`, то тесты будут запускаться с базой данных `test_weblate`. Смотрите подробности по настройке в главе *Настройка базы данных для Weblate*.

При конфигурации среды непрерывной интеграции также используется `weblate/settings_test.py` (см. *Непрерывная интеграция*), его параметры можно поменять через переменные среды:

```
# 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
```

До того как запустить тесты, надо сначала собрать статические файлы, так как они нужны для некоторых из них:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

Вы также можете запустить только определённые тесты:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

Подсказка: Тесты можно также запустить внутри docker-контейнера для разработки, см. *Запуск Weblate локально внутри Docker*.

См.также:

Смотрите подробности о том, как писать и запускать тесты в документации Django: Testing in Django.

3.10 Схемы данных

Для определения структуры внешних JSON-файлов Weblate использует JSON-схему.

3.10.1 Схема памяти переводов Weblate'а

https://weblate.org/schemas/weblate-memory.schema.json			
тип	*массив*		
элементы	*Элемент памяти переводов*		
	тип	*объект*	
	свойства		
	• **category**	*Категория строки*	
		1 — глобальная, 2 — общая, 10000000+ — специфичная для проекта, 20000000+ — специфичная для пользователя	
		тип	*целое число*
		примеры	1
		минимум	0
		значение по умолчанию	1
	• **origin**	*Происхождение строки*	
		Имя файла или название компонента	
		тип	*строка*
		примеры	test.tmx
			project/component
		значение по умолчанию	
	• **source**	*Исходная строка*	
		тип	*строка*
		примеры	Hello
		мин. длина	1
		значение по умолчанию	
	• **source_language**	*Исходный язык*	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		тип	*строка*
		примеры	en
		шаблон	^[^]+$
		значение по умолчанию	
	• **target**	*Целевая строка*	
		тип	*строка*
		примеры	Привет
		мин. длина	1
		значение по умолчанию	
	• **target_language**	*Целевой язык*	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		тип	*строка*
		примеры	ru
		шаблон	^[^]+$
		значение по умолчанию	
	дополнительные свойства	False	
определения			

См.также:

Память переводов, `dump_memory`, `import_memory`

3.10.2 Файл экспорта данных пользователя Weblate'а

https://weblate.org/schemas/weblate-userdata.schema.json			
тип	*объект*		
свойства			
• **basic**	*Основные данные*		
	тип	*объект*	
	свойства		
	• **username**	*Имя пользователя*	
		тип	*строка*
		примеры	admin
		значение по умолчанию	
	• **full_name**	*Полное имя*	
		тип	*строка*
		примеры	Администратор Weblate
		значение по умолчанию	
	• **email**	*Адрес электронной почты*	
		тип	*строка*
		примеры	noreply@example.com
		значение по умолчанию	
	• **date_joined**	*Дата присоединения*	
		тип	*строка*
		примеры	2019-11-18T18:53:54.862Z
		значение по умолчанию	
• **profile**	*Профиль*		
	тип	*объект*	
	свойства		
	• **language**	*Язык*	
		тип	*строка*
		примеры	ru
		шаблон	^.*$
		значение по умолчанию	
	• **suggested**	*Количество предложенных строк*	
		тип	*целое число*
		примеры	1
		значение по умолчанию	0
	• **translated**	*Количество переведённых строк*	
		тип	*целое число*
		примеры	24
		значение по умолчанию	0
	• **uploaded**	*Количество загруженных снимков экрана*	
		тип	*целое число*
		примеры	1
		значение по умолчанию	0
	• **hide_completed**	*Скрывать завершённые переводы с панели управления*	
		тип	*логический*
		примеры	False

continues on next page

Таблица 2 – продолжение с предыдущей страницы

		значение по умолчанию	True	
	• **secondary_in_zen**	*Выводить ли вспомогательные переводы в дзен-режиме*		
		тип	*логический*	
		примеры	True	
		значение по умолчанию	True	
	• **hide_source_secondary**	*Скрывать ли источник, если есть вспомогательный перевод*		
		тип	*логический*	
		примеры	False	
		значение по умолчанию	True	
	• **editor_link**	*Ссылка на редактор*		
		тип	*строка*	
		примеры		
		шаблон	^.*$	
		значение по умолчанию		
	• **translate_mode**	*Режим редактора переводов*		
		тип	*целое число*	
		примеры	0	
		значение по умолчанию	0	
	• **zen_mode**	*Дзен-режим редактора*		
		тип	*целое число*	
		примеры	0	
		значение по умолчанию	0	
	• **special_chars**	*Специальные символы*		
		тип	*строка*	
		примеры		
		шаблон	^.*$	
		значение по умолчанию		
	• **dashboard_view**	*Вид панели управления по умолчанию*		
		тип	*целое число*	
		примеры	1	
		значение по умолчанию	0	
	• **dashboard_component_list**	*Список компонентов по умолчанию*		
		значение по умолчанию	null	
		любое из	тип	*null*
			тип	*целое число*
	• **languages**	*Языки перевода*		
		тип	*массив*	
		значение по умолчанию		
		элементы	*Код языка*	
			тип	*строка*
			примеры	ru
			шаблон	^.*$
			значение по умолчанию	
	• **secondary_languages**	*Вспомогательные языки*		
		тип	*массив*	

continues on next page

Таблица 2 – продолжение с предыдущей страницы

		значение по умолчанию		
		элементы	*Код языка*	
			тип	*строка*
			примеры	uk
			шаблон	^.*$
			значение по умолчанию	
	• **watched**	*Отслеживаемые проекты*		
		тип	*массив*	
		значение по умолчанию		
		элементы	*Плашка проекта*	
			тип	*строка*
			примеры	weblate
			шаблон	^.*$
			значение по умолчанию	
• **auditlog**	*Журнал аудита*			
	тип	*массив*		
	значение по умолчанию			
	элементы	*Записи*		
		тип	*объект*	
		свойства		
		• **address**	*IP-адрес*	
			тип	*строка*
			примеры	127.0.0.1
			шаблон	^.*$
			значение по умолчанию	
		• **user_agent**	*Агент пользователя* (*User agent*)	
			тип	*строка*
			примеры	PC / Linux / Firefox 70.0
			шаблон	^.*$
			значение по умолчанию	
		• **timestamp**	*Отметка времени*	
			тип	*строка*
			примеры	2019-11-18T18:58:30.845Z
			шаблон	^.*$
			значение по умолчанию	
		• **activity**	*Действие*	
			тип	*строка*
			примеры	login
			шаблон	^.*$
			значение по умолчанию	
определения				

См.также:

Профиль пользователя, `dumpuserdata`

3.11 Выпуск новых версий Weblate

3.11.1 Расписание выпусков

Weblate Придерживается двухмесячного цикла основных выпусков (x.y). Между ними обычно происходят выпуски с исправлениями ошибок (x.y.z), которые исправляют проблемы, проскочившие в основные выпуски.

Изменение мажорной версии означает, что при обновлении эту версию нельзя пропускать. В таких случаях всегда необходимо сначала обновиться до версии *x.0* перед тем, как обновляться до старших выпусков *x.y*.

См.также:

Обновление Weblate

3.11.2 Планирование выпусков

Список исправленных ошибок и новых возможностей для предстоящего выпуска собирается с помощью целей выпуска (milestones) на GitHub; вы можете посмотреть нашу дорожную карту на <https://github.com/WeblateOrg/weblate/milestones>.

3.11.3 Процесс выпуска

Что нужно проверить перед выпуском:

1. Проверить новые языки, на которые был сделан перевод командой **./scripts/list-translated-languages**.
2. Установить финальную версию командой **./scripts/prepare-release**.
3. Удостовериться, что скриншоты находятся в актуальном состоянии с помощью команды **make -C docs update-screenshots**.
4. Merge any possibly pending translations **wlc push; git remote update; git merge origin/weblate**

Выполнить выпуск:

5. Произвести выпуск: **./scripts/create-release --tag** (смотрите, что для этого необходимо ниже).

Что нужно сделать вручную после выпуска:

6. Обновить образ Docker.
7. Закрыть целевую версию (milestone) на GitHub.
8. Как только образ докер будет проверен, добавить метку и загрузить его.
9. Обновить схему Helm до новой версии.
10. Добавить новую версию в `.github/workflows/migrations.yml`, чтобы миграция на неё покрывалась тестами.
11. Increase version in the website download links.
12. Увеличить версию в репозитории с помощью команды **./scripts/set-version**.

Чтобы создать метки с помощью скрипта **./scripts/create-release**. Вам понадобится:

- GnuPG с закрытым ключом, используемым для подписи выпусков
- Доступ на отправку изменений в git-репозиторий Weblate (он отправляет метки)
- Настроенная утилита **hub** и права на создание выпусков в репозитории Weblate

- SSH-доступ к серверу скачивания Weblate (файлы для скачивания с Web-сайта копируются туда)

3.12 Безопасность и конфиденциальность

Совет: В Weblate, безопасность поддерживает такую среду, которая защищает конфиденциальность наших пользователей.

Development of Weblate adheres to the Best Practices of the Linux Foundation's Core Infrastructure Initiative.

3.12.1 Отслеживание зависимостей для реагирования на уязвимости

Security issues in our dependencies are monitored using Dependabot. This covers the Python and JavaScript libraries, and the latest stable release has its dependencies updated to avoid vulnerabilities.

Подсказка: 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 Docker container security

Контейнеры Docker'а сканируются сканерами Anchore и Trivy.

This allows us to detect vulnerabilities early and release improvements quickly.

You can get the results of these scans at GitHub — they are stored as artifacts on our CI in the SARIF format (Static Analysis Results Interchange Format).

См.также:

Непрерывная интеграция

3.13 О Weblate

3.13.1 Цели проекта

Инструменты непрерывной локализации, основанные на web-технологиях, с плотной *интеграцией с системами контроля версий* и поддерживающими широкий набор *форматов файлов*, дают возможность переводчикам проще вносить свой вклад в различные проекты.

3.13.2 Название проекта

«Weblate» образовано слиянием слов «web» и «translate» (переводить).

3.13.3 Веб-сайт проекта

The landing page is https://weblate.org and a cloud hosted service at https://hosted.weblate.org. This documentation can be found on https://docs.weblate.org.

3.13.4 Логотипы проекта

Логотип проекта и другая графика доступна в репозитории: <https://github.com/WeblateOrg/graphics/>.

3.13.5 Руководство

Этот проект сопровождается Михалом Чигаржем (Michal Čihař) <michal@cihar.com>.

3.13.6 Авторы

Weblate был создан Михалом Чиигаржем (Michal Čihař) <michal@cihar.com> в 2012 году. С этого момента уже тысячи людей внесли в него свой вклад.

3.14 Лицензия

Авторское право (C) 2012–2020 Михал Чигарж (Michal Čihař) michal@cihar.com

Это свободная программа: вы можете перераспространять ее и/или изменять ее на условиях Стандартной общественной лицензии GNU в том виде, в каком она была опубликована Фондом свободного программного обеспечения; либо версии 3 лицензии, либо (по вашему выбору) любой более поздней версии.

Эта программа распространяется в надежде, что она будет полезной, но БЕЗО ВСЯКИХ ГАРАНТИЙ; даже без неявной гарантии ТОВАРНОГО ВИДА или ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННЫХ ЦЕЛЕЙ. Подробнее см. в Стандартной общественной лицензии GNU.

Вы должны были получить копию Стандартной общественной лицензии GNU вместе с этой программой. Если это не так, см. <https://www.gnu.org/licenses/>.

ГЛАВА 4

История изменений

4.1 Weblate 4.5.2

Released on March 26th 2021.

- Configurable schedule for automatic translation.
- Added Lua format check.
- Ignore format strings in the *Повторяющиеся слова* 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.
- Improved performance when adding new strings.
- Several bug fixes in quality checks.
- Several performance improvements.
- Added integration with *Discover Weblate*.
- Fixed checks behavior with read-only strings.

4.2 Weblate 4.5.1

Released on March 05th 2021.

- Fixed editing of glossary flags in some corner cases.
- Extend metrics usage to improve performance of several pages.
- Store correct source language in TMX files.
- Better handling for uploads of monolingual PO using API.
- Improved alerts behavior glossaries.
- Improved Markdown link checks.
- Indicate glossary and source language in breadcrumbs.

- Paginated component listing of huge projects.
- Improved performance of translation, component or project removal.
- Улучшена производительность массовой правки.
- Fixed preserving «Needs editing» and «Approved» states for ODF files.
- Improved interface for customizing translation-file downloads

4.3 Weblate 4.5

Выпущен 19 февраля 2021 года.

- Добавлена поддержка формата `lua-format`, используемого в PO-файлах gettext.
- Добавлена поддержка разделяемых компонентов между проектами.
- Исправлено поведение проверки «несколько переменных без названий» в случае, если использовались разные флаги форматирования.
- Из проектов удалено поле «список рассылки»; вместо него рекомендуется использовать общие инструкции для переводчиков.
- Добавлена надстройка «Генерация псевдолокалей».
- Добавлена поддержка файлов формата TermBase eXchange.
- Добавлена поддержка ручного объявления вариантов строк с помощью флага `variants`.
- Улучшена производительность проверок согласованности.
- Улучшена производительность памяти переводов при работе с очень длинными строками.
- Добавлена поддержка поиска по пояснениям.
- Строки теперь можно добавлять и удалять также и в двуязычных форматах.
- Расширен список поддерживаемых языков для машинного перевода Amazon Translate.
- Автоматическое включение проверок Java MessageFormat для файлов свойств Java.
- Добавлен новый метод загрузки для добавления новых строк к переводу.
- Добавлен простой интерфейс просмотра перевода.
- Глоссарии теперь хранятся как обычные компоненты.
- Удалён особый API для глоссариев, поскольку теперь для них используется API компонента.
- Добавлен упрощённый интерфейс для переключения некоторых флагов.
- Добавлена поддержка непереводимых и нежелательных терминов в глоссарии.
- Добавлена поддержка определения терминологии в глоссарии.
- Переключатель направления текста перемещён, чтобы освободить побольше места для визуальной клавиатуры.
- Добавлен параметр для автоматического отслеживания проектов, в которые пользователь внёс свой вклад.
- Добавлена проверка на соответствие перевода термину в глоссарии.
- Добавлена поддержка настройки цвета навигационного текста.

4.4 Weblate 4.4.2

Выпущен 14 января 2021 года.

- В дистрибутиве исправлен один повреждённый MO-файл.

4.5 Weblate 4.4.1

Выпущен 13 января 2021 года.

- Исправлена отмена изменений в строках с множественным числом.
- Исправлено отображение помощи для настроек проекта.
- Улучшено управление пользователями.
- Улучшена обработка контекста в одноязычных PO-файлах.
- Исправлено поведение надстройки «Очистка» при обработке форматов HTML, ODF, IDML и Windows RC.
- Исправлен парсинг расположения из CSV-файлов.
- Использование сжатия содержимого при скачивании файлов.
- Улучшена процедура импорта ZIP-файлов.
- Улучшено распознавание форматов файлов при загрузке.
- Теперь Weblate старается избегать дупликации запросов на слияние на Pagure.
- Улучшена производительность при отображении призрачных переводов.
- Переписана реализация поля редактирования переводов, чтобы она использовала родную область ввода браузера.
- Исправлена ошибка из-за которой надстройка «Очистка» добавляла новые строки.
- Добавлен API для запроса/изменения информации о надстройках.

4.6 Weblate 4.4

Выпущен 15 декабря 2020 года.

- Улучшена проверка значений при создании нового компонента.
- Weblate теперь требует Django 3.1.
- Добавлена поддержка настройки внешнего вида через интерфейс управления.
- Исправлена обработка строк, доступных только для чтения, при массовых правках.
- Улучшена интеграция с CodeMirror.
- Добавлена надстройка для удаления пустых строк из файлов перевода.
- Редактор CodeMirror теперь используется и для переводов.
- Подсветка синтаксиса в редакторе перевода для XML, HTML, Markdown и reStructuredText.
- Подсветка местозаполнителей в редакторе перевода.
- Улучшена поддержка нестандартных языковых кодов.
- Добавлены предупреждения при использовании неоднозначных языковых кодов.

- При добавлении нового языка перевода пользователю теперь показывается отфильтрованный список языков.
- Расширены возможности поиска изменений в истории.
- Улучшены страницы с информацией о тарифном плане и рабочий процесс на тарифном плане для свободных проектов.
- Расширен API по работе со статистикой переводов.
- Улучшена вкладка «другие языки» на страницах перевода.
- Добавлен API для задач.
- Улучшена производительность загрузки файлов.
- Улучшено отображение специальных символов, определённых пользователем.
- Улучшена производительность автоматического перевода.
- Несколько незначительные улучшения пользовательского интерфейса.
- Улучшено именование скачиваемых ZIP-файлов.
- Добавлена опциональная возможность получать уведомления от неотслеживаемых проектов.

4.7 Weblate 4.3.2

Выпущен 4 ноября 2020 года.

- Исправлено падение при использовани некоторых файловых масок компонентов.
- Улучшена точность проверки «Повторяющиеся слова».
- Добавлена поддержка запросов на извлечение Pagure.
- Улучшены сообщения об ошибках, которые выводятся при неудавшейся регистрации.
- Отменено отображение комментариев разработчиков как Markdown.
- Упрощена настройка Git-репозиториев с названием ветки по умолчанию отличным от «master».
- Вновь созданные внутренние репозитории теперь используют «main» в качестве ветки по умолчанию.
- Уменьшение количества ложно-положительных результатов для проверки «Перевод без изменения» при переводе ReST (reStructuredText).
- Исправлены проблемы с отображением CodeMirror в некоторых ситуациях.
- Группа «Шаблоны» переименованы в группу «Источники», дабы сделать её смысл более очевидным.
- Исправлены запросы на извлечения GitLab в репозиториях с более длинными путями.

4.8 Weblate 4.3.1

Выпущен 21 октября 2020 года.

- Улучшена производительность автоматического перевода.
- Исправлено истечение срока действия сессии для авторизованных пользователей.
- Добавлена возможность скрыть информацию о версии Weblate.
- Улучшена совместимость с сервером Bitbucket.
- Улучшена производительность обновления памяти переводов.
- Уменьшино использование памяти.

- Улучшена производительность просмотра в виде матрицы.
- Добавлено подтверждение перед удалением пользователей из проекта.

4.9 Weblate 4.3

Выпущен 15 октября 2020 года.

- В API включена статистика пользователей.
- Исправлена сортировка компонентов на страницах с постраничной разбивкой результатов.
- Определение исходного языка для глоссария.
- Переписана поддержка GitHub'овских и GitLab'овских запросов на извлечение.
- Исправлен счётчик статистики после удаления предложения.
- Расширен общедоступный профиль пользователя.
- Исправлены настройки принудительных проверок.
- Улучшена документация по встроенному резервному копированию.
- Атрибут исходного языка перемещён из проекта в компонент.
- Добавлена проверка форматирования локализации Vue.
- Обычная проверка заполнителей теперь поддерживает регулярные выражения.
- Улучшен внешний вид матричного режима.
- «Автоматизированный перевод» теперь называется «автоматическими предложениями».
- Добавлена поддержка взаимодействия с несколькими экземплярами GitLab'a или GitHub'a.
- API расширен для покрытия обновления проектов, а также обновления и удаления единиц перевода и глоссариев.
- API единицы перевода теперь правильно обрабатывает строки с формами множественного числа.
- Создать компонент теперь можно, загрузив ZIP-файл или документ.
- Единообразие в кодах состояний ответа API.
- Поддержка разметки Markdown в соглашении участника.
- Улучшено отслеживание исходных строк.
- Улучшена совместимость с форматами JSON, YAML и CSV.
- Добавлена поддержка удаления строк.
- Улучшена производительность скачивания файлов.
- Улучшен интерфейс управления репозиториями.
- Автоматическое включение java-формата для Android.
- Добавлена поддержка локализованных снимков экрана.
- Добавлена поддержка Python 3.9.
- Исправлен перевод HTML-файлов при некоторых условиях.

4.10 Weblate 4.2.2

Выпущен 2 сентября 2020 года.

- Исправлено сопоставление исходных строк для форматов на основе JSON.
- Исправлено перенаправление входа для некоторых конфигураций аутентификации.
- Исправлена аутентификация через LDAP с синхронизацией групп.
- Исправлено падение при сообщении о прогрессе автоматического перевода.
- Исправлено уплотнение коммитов Git'а при включённом добавлении завершителей.
- Исправлено создание компонентов из локальной системы контроля версий с помощью API.

4.11 Weblate 4.2.1

Выпущен 21 августа 2020 года.

- Исправлено сохранение форм множественного числа для некоторых локалей в ресурсах Android'а.
- Исправлено падение надстройки очистки на некоторых файлах XLIFF.
- Теперь доступно создание сети доставки содержимого (CDN) локализации внутри образа Docker.

4.12 Weblate 4.2

Выпущен 18 августа 2020 года.

- Улучшены страницы пользователей и добавлен список пользователей.
- Прекращена поддержка перехода с выпусков серии 3.x, перехода через версии 4.1 или 4.0.
- Добавлены экспорты в несколько одноязычных форматов.
- Улучшены графики деятельности.
- Количество отображаемых соседних строк теперь поддаётся настройке.
- Добавлена поддержка блокировки компонентов, в которых возникли ошибки в репозитории.
- Упрощена основная навигация (кнопки заменены на иконки).
- Улучшена обработка кода языка в интеграции с Google Translate.
- Дополнение Git для уплотнения коммитов теперь генерирует завершители `Co-authored-by:`.
- Улучшен анализатор поисковых запросов.
- Улучшена обратная связь с пользователями при проверке строк формата.
- Улучшена производительность при массовых изменениях состояния.
- Добавлены перенаправления для совместимости после переименования проекта или компонента.
- Добавлены уведомления об одобрении строк, блокировке компонентов и изменении лицензии.
- Добавлена поддержка ModernMT.
- Разрешено избегать перезаписи одобренных переводов при загрузке файлов.
- Прекращена поддержка некоторых перенаправлений URL-адресов, существовавших ради совместимости.
- Добавлена проверка для шаблонных литералов ECMAScript.
- Добавлен параметр для отслеживания компонента.

- Из ключей единиц перевода JSON удалена ведущая точка.
- Удалена отдельная очередь Celery для памяти переводов.
- Разрешено одновременно переводить все компоненты одного языка.
- Разрешено настраивать HTTP -заголовки `Content-Security-Policy`.
- Добавлена поддержка задания псевдонимов для языков на уровне проекта.
- Новая надстройка для помощи в локализации HTML или JavaScript, смотрите раздел *CDN локализации JavaScript'a*.
- Теперь домен Weblate настраивается в настройках, смотрите описание параметра `SITE_DOMAIN`.
- Добавлена поддержка поиска по компонентам и проектам.

4.13 Weblate 4.1.1

Выпущен 19 июня 2020 года.

- Исправлено изменение конфигурации автоматического исправления или надстроек в Docker.
- Исправлено возможное падение на странице «О Weblate».
- Улучшена установка скомпилированных файлов локали.
- Исправлено добавление слов в глоссарий.
- Исправлены горячие клавиши для автоматизированного перевода.
- Удалён отладочный вывод, вызывающий при некоторых настройках отбрасывание событий журнала.
- Исправлена индикация блокировки в списке проектов.
- Исправлено перечисление ключей GPG при некоторых настройках.
- Добавлен параметр для выбора используемой версии API DeepL.
- Добавлена поддержка работы в качестве провайдера SAML-сервиса, смотрите *Аутентификация через SAML*.

4.14 Weblate 4.1

Выпущен 15 июня 2020 года.

- Добавлена поддержка создания новых переводов с включённым кодом страны.
- Добавлена поддержка поиска исходных строк со снимками экрана.
- Расширен состав информации, доступный в статистике.
- Улучшено редактирование поиска на страницах перевода.
- Улучшена обработка одновременных обновлений репозитория.
- В форму создания проекта включён исходный язык.
- В список переводчиков включено количество изменений.
- Исправлена возникающая в некоторых случаях ошибка выбора языка пользовательского интерфейса.
- Разрешение внесения методов регистрации в белый список при закрытых регистрациях.
- Улучшен поиск смежных терминов в глоссарии.
- Улучшены соответствия с памятью переводов.
- Группировка одинаковых результатов автоматизированного перевода.

- Добавление прямой ссылки для редактирования снимка экрана прямо со страницы перевода.
- Улучшен диалог подтверждения удаления.
- В скачиваемый ZIP-файл включены шаблоны.
- В объявлениях добавлена поддержка Markdown и настройка уведомлений.
- Расширенная информация в списках проверок.
- Добавлена поддержка новых форматов файлов: *Строки Laravel PHP*, *Файлы HTML*, *Формат OpenDocument*, *Формат IDML*, *Файлы ресурсов Windows*, *Переводы в INI-файлах*, *Переводы в INI-файлах Inno Setup*, *Файлы свойств GWT*, *JSON-файлы go-i18n* и *Файл ARB*.
- Последовательное использование статуса «пропущена» в качестве статуса пропущенных проверок.
- Добавлена поддержка для настройки включённых надстроек по умолчанию.
- Исправлено сочетание клавиш редактора для пропуска проверок.
- Улучшен машинный перевод строк с заполнителями.
- Показ призрачного перевода для языков пользователя, чтобы упростить ему начало перевода.
- Улучшен разбор кода языка.
- Показ переводов на язык пользователя на первом месте в списке.
- Переименование «форм» на более обобщённые «варианты».
- Добавлены новые проверки качества: *Несколько переменных без названий*, *Давно не переводилась* и *Повторяющиеся слова*.
- Вновь введена поддержка стирания памяти переводов.
- Исправлен параметр для включения в исходной строке игнорирования проверок.
- Добавлена поддержка настройки различных веток, в которые следует отправлять изменения.
- API теперь сообщает об ограничении в частоте запросов в заголовках HTTP.
- Добавлена поддержка Google Translate V3 API (расширенная).
- Добавлена возможность ограничения доступа на уровне компонентов.
- Во флагах перевода добавлена поддержка пробельных и других специальных символов, смотрите раздел *Настройка поведения с помощью флагов*.
- Отображаемый текст, если он включён, показывается всегда.
- API теперь поддерживает фильтрацию изменений.
- Добавлена поддержка обмена глоссариями между проектами.

4.15 Weblate 4.0.4

Выпущен 7 мая 2020 года.

- Исправлено выполнение тестового набора в некоторых окружениях с Python'ом 3.8.
- Исправление опечаток в документации.
- Исправлена возникавшая в некоторых случаях ошибка создания компонентов с помощью API.
- Исправлены ошибки JavaScript, ломающие навигацию на мобильных устройствах.
- Исправлено падение при отображении некоторых проверок.
- Исправлен вывод списка снимков экрана.
- Исправлены ежемесячные выдержки с уведомлениями.

- Исправлено поведение промежуточного перевода с единицами перевода, не существовавшими в переводе.

4.16 Weblate 4.0.3

Выпущен 2 мая 2020 года.

- Исправлено возможное падение в отчётах.
- Упоминания пользователей в комментариях теперь не чувствительны к регистру.
- Исправлена миграция PostgreSQL для не суперпользователей.
- Исправлено изменение URL-адреса репозитория при создании компонента.
- Исправлено падение при исчезновении вышестоящего репозитория.

4.17 Weblate 4.0.2

Выпущен 27 апреля 2020 года.

- Улучшена производительность статистики перевода.
- Улучшена производительность смены меток.
- Улучшена производительность массовой правки.
- Улучшена производительность памяти переводов.
- Исправлено возможное падение при удалении компонента.
- Исправлено отображение изменений перевода в некоторых пограничных случаях.
- Улучшено предупреждение о слишком длинной очереди Celery.
- Исправлены возможные ложные срабатывания при проверке согласованности.
- Исправлена взаимоблокировка при изменении репозитория связанного компонента.
- В список изменений, CSV и отчёты включено расстояние редактирования.
- Обход ложных срабатываний проверки интервала между знаками препинания для канадского французского языка.
- Исправлен экспорт в формат XLIFF с заполнителями.
- Исправлено ложное срабатывание с проверкой на символ нулевой ширины.
- Улучшены отчёты об ошибках конфигурации.
- Исправлена загрузка двуязычных исходных текстов.
- Автоматическое определение поддерживаемых языков для машинного перевода DeepL.
- Исправлено отображение индикатора выполнения в некоторых пограничных случаях.
- Исправлено срабатывание некоторых проверок на непереведённых строках.

4.18 Weblate 4.0.1

Выпущен 16 апреля 2020 года.

- Исправлена установка пакета из PyPI.

4.19 Weblate 4.0

Выпущен 16 апреля 2020 года.

- Weblate теперь требует Python 3.6 или новее.
- Добавлен обзор управления предупреждений компонентов.
- Добавлено предупреждение компонента о сломанных URL-адресах браузера репозитория.
- Улучшены страницы входа и регистрации.
- Управление доступом к проекту и настройки рабочего процесса интегрированы в настройки проекта.
- Добавлена проверка и подсветка для интерполяций и вложенных строк i18next.
- Добавлена проверка и подсветка для процентных заполнителей.
- Отображение неудачных проверок в предложениях.
- Запись изменений исходной строки в историю.
- Обновление переводчика Microsoft Translator на третью версию его API.
- Повторная реализация серверной части памяти переводов.
- В *Поиск* добавлена поддержка нескольких `is:` поисков.
- Разрешено делать так, чтобы проверка *Перевод без изменения* не попадала во внутренний чёрный список.
- Улучшено извлечение комментариев из одноязычных po файлов.
- «Доски сообщений» переименованы в «объявления».
- Исправлены случайные проблемы с регистрационными письмами.
- Улучшена надстройка обновления LINGUAS-файла: теперь она обрабатывает большее количество синтаксических вариантов.
- Исправлено редактирование одноязычного исходного файла XLIFF.
- В *Поиск* добавлена поддержка точного соответствия.
- Расширенный API, охватывающий снимки экрана, пользователей, группы, списки компонентов и расширенное создание проектов.
- Добавлена поддержка загрузки исходных текстов в двуязычные переводы.
- Добавлена поддержка промежуточного языка от разработчиков.
- Добавлена поддержка рецензирования исходных строк.
- Расширенные параметры скачивания памяти переводов со всей платформы.

4.20 Серия Weblate 3.x

4.20.1 Weblate 3.11.3

Выпущен 11 марта 2020 года.

- Исправлен поиск полей с определённым приоритетом.
- Исправлен предопределённый запрос для недавно добавленных строк.
- Исправлен поиск, возвращающий повторяющиеся совпадения.
- Исправлена отрисовка уведомлений в Gmail.
- Исправлен возврат изменений из истории.
- Добавлены ссылки на события в выдержках с уведомлениями.
- Исправлено электронное письмо для подтверждения удаления учётной записи.
- Добавлена поддержка аутентификации через Slack в контейнере Docker.
- Устранение отправки уведомлений для языков, на которые пользователь не подписан.
- Включение в обзор производительности очередей Celery.
- Исправлены ссылки на документацию для надстроек.
- Уменьшение количества ложно-отрицательных результатов для проверки «Перевод без изменения».
- Повышение версии зависимости bleach для закрытия уязвимости CVE-2020-6802.
- Исправлено перечисление изменений в истории на уровне проекта.
- Исправлено аннулирование статистики в некоторых граничных случаях.
- Исправлен поиск определённых состояний строки.
- Улучшено поведение проверки строк формата при пропущенном проценте.
- Исправлена аутентификация с использованием некоторых сторонних провайдеров.

4.20.2 Weblate 3.11.2

Выпущен 22 февраля 2020 года.

- Исправлена отрисовка предложений.
- Исправлены некоторые строки, для которых ошибочно сообщалось, что в нет слов.

4.20.3 Weblate 3.11.1

Выпущен 20 февраля 2020 года.

- Документированы изменения в настройке Celery.
- Улучшена проверка имени файла при создании компонентов.
- Исправлены минимальные версии некоторых зависимостей.
- Исправлено добавление групп в определённых версиях Django.
- Исправлена ручная отправка изменений в вышестоящий репозиторий.
- Улучшен поиск соответствия в словаре.

4.20.4 Weblate 3.11

Выпущен 17 февраля 2020 года.

- Разрешено использование URL-адреса для отправки в систему контроля версий во время создания компонента через API.
- Проверка ширины отрисованного текста теперь показывает изображение с отрисованным текстом.
- Исправлены ссылки в электронных письмах с уведомлениями.
- Улучшен внешний вид простых текстовых электронных писем.
- Отображение проигнорированных проверок и разрешение сделать их снова активными.
- Отображение соседних ключей в одноязычных переводах.
- Добавлена поддержка группировки строковых форм.
- Рекомендация обновления до новой версии Weblate в проверках системы.
- Предоставление более подробного анализа для предупреждения о повторяющемся языке.
- Включение более подробной информации о лицензиях на страницах проекта.
- Автоматическое преобразование поверхностных локальных копий в полные, если это необходимо.
- Исправлено скачивание строк, требующих действий.
- Новое предупреждение о повторном использовании одной и той же маски файлов.
- Улучшено извлечение мест подстановки в XML.
- Теперь параметр `SINGLE_PROJECT` может включать принудительное перенаправление на выбранный проект.
- Добавлена возможность указания, что описанная в комментарии проблема решена.
- Добавлена массовая правка флагов.
- Добавлена поддержка меток.
- Добавлена надстройка «Массовая правка».
- Добавлен параметр для *Принудительные проверки*.
- Увеличен умолчательный срок действия ссылок подтверждения.
- Улучшена интеграция с Matomo.
- Исправлена проверка *Уже переведено* для корректной обработки изменений в исходных строках.
- Расширение настроек автоматического обновления, контролируемых параметром `AUTO_UPDATE`.
- Надстройки LINGUAS теперь выполняют полную синхронизацию переводов с Weblate.

4.20.5 Weblate 3.10.3

Выпущен 18 января 2020 года.

- Поддержка translate-toolkit 2.5.0.

4.20.6 Weblate 3.10.2

Выпущен 18 января 2020 года.

- К проектам добавлена индикация блокировки.
- Исправлена ошибка в CSS, вызывавшая мерцание в некоторых веб-браузерах.
- Исправлен поиск в системах с неанглийскими локалями.
- Улучшено сопоставление репозитория для обработчиков GitHub и Bitbucket.
- Исправлена миграция данных на некоторых установках Python 2.7.
- Разрешена настройка поверхностного клонирования в Git'е.
- Улучшена обработка фоновых уведомлений.
- Исправлена ошибка отправки сломанных форм при использовании браузерной кнопки «назад».
- Новая надстройка для настройки форматирования YAML.
- Исправлена проверка «Одинаковые формы множественного числа»: теперь она не срабатывает в языках с единственной формой множественного числа.
- Исправлен поиск с использованием регулярных выражений по некоторым полям.

4.20.7 Weblate 3.10.1

Выпущен 9 января 2020 года.

- Расширенный API с созданием перевода.
- Исправлено несколько граничных случаев при миграции данных.
- Совместимость с Django 3.0.
- Улучшена производительность очистки данных.
- Добавлена поддержка настройки security.txt.
- Улучшены хлебные крошки в журнале изменений.
- Улучшен список переводов в панели управления.
- Улучшены HTTP-ответы для веб-обработчиков.
- Добавлена поддержка запросов на слияние GitLab в контейнере Docker.

4.20.8 Weblate 3.10

Выпущен 20 декабря 2019 года.

- Улучшен пользовательский интерфейс приложения.
- Добавлена проверка на двойной пробел.
- Исправлено создание новых языков.
- Избежание отправки уведомлений из журнала аудита на удалённые адреса электронной почты.
- Добавлена поддержка строк только для чтения.
- Добавлена поддержка Markdown'а в комментариях.
- Разрешение размещения текста инструкции по переводу в информации о проекте.
- Добавлено копирование в буфер обмена для дополнительных языков.
- Улучшена поддержка Mercurial.

- Улучшена производительность извлечения из Git-репозитория.
- Добавлен поиск по возрасту строки.
- Показ исходного языка для всех переводов.
- Показ контекста для соседних строк.
- Добавлена поддержка уведомлений об операциях с репозиторием.
- Улучшены списки перевода.
- Расширены возможности поиска.
- Добавлена поддержка автоматического перевода строк, помеченных на правку.
- Избежание отправки повторных уведомлений для предупреждений в связанных компонентах.
- Улучшение умолчательного сообщения о запросе на слияние.
- Улучшение индикации состояния строки в Дзен-режиме.
- Добавлена поддержка большего количества языков в Яндекс.Переводчике.
- Улучшен внешний вид электронных писем уведомлений.
- Предоставлена возможность выбора лицензии на перевод.

4.20.9 Weblate 3.9.1

Выпущен 28 октября 2019 года.

- Из резервных копий удалены кое-какие ненужные файлы.
- Исправлено возможное падение в отчётах.
- Исправлена падение при миграции из одной базы данных в другую.
- Добавлена поддержка принудительной отправки изменений в Git-репозитории.
- Снижение риска аннулирования токена регистрации.
- Исправлено удаление учётной записи, упёршейся в ограничение на частоту запросов.
- Добавлен поиск по приоритету.
- Исправлено возможное падение при добавлении строк в JSON-файл.
- Проверка и исправление небезопасного HTML теперь учитывает разметку исходной строки.
- Избегание отправки уведомлений приглашённым, но в то же время удалённым пользователям.
- Исправлено SSL-соединение с redis в Celery, находящимся в контейнере Docker'а.

4.20.10 Weblate 3.9

Выпущен 15 октября 2019 года.

- Включение в скачиваемые файлы метаданных Weblate.
- Улучшен пользовательский интерфейс для неудачных проверок.
- Указание недостающих строк в проверках формата.
- Отдельная проверка на пробелы между знаками препинания французского языка.
- Добавлена поддержка исправления некоторых ошибок проверки качества.
- Добавлено отдельное разрешение на создание новых проектов.
- Расширена статистика по количеству символов.

- Улучшение поддержки языковых кодов в стиле Java.
- Добавлена новая обобщённая проверка на заполнители.
- Добавлена поддержка заполнителей WebExtension JSON.
- Добавлена поддержка формата плоского XML-файла.
- API расширен методами удаления и создания проекта, компонента и перевода.
- Добавлена поддержка веб-обработчиков Gitea и Gitee.
- Добавлена новая пользовательская проверка на основе регулярных выражений.
- Разрешение настройки внесения вклада в общую память переводов.
- Скачивание ZIP-архива добавлено для большего количества файлов перевода.
- Разбор атрибутов maxwidth и font в формате XLIFF сделан совместимым со стандартом.
- Добавлена новая проверка и исправление для безопасной HTML-разметки при переводе веб-приложений.
- В компоненте добавлено предупреждение при обнаружении неподдерживаемой конфигурации.
- Добавлена надстройка автоматического перевода для начального перевода.
- Автоматический перевод расширен для добавления предложений.
- Отображение в обзоре параметров надстройки.
- Sentry теперь поддерживается через современный Sentry SDK вместо Raven.
- Изменены примеры настроек, чтобы они лучше соответствовали окружению рабочей среды.
- Добавлено автоматическое резервное копирование с помощью BorgBackup.
- Разбиение надстройки очистки для RESX на две надстройки очистки и обновления, чтобы избежать нежелательных обновлений файлов.
- Добавлены расширенные возможности поиска.
- Пользователям разрешено скачивать свои собственные отчёты.
- Для помощи в настройке компонентов добавлено руководство по локализации.
- Добавлена поддержка запросов на слияние GitLab.
- Улучшено отображение статуса репозитория.
- Выполнение автоматического перевода в фоновом режиме.

4.20.11 Weblate 3.8

Выпущен 15 августа 2019 года.

- Добавлена поддержка упрощённого создания похожих компонентов.
- Добавлена поддержка разбора флагов перевода из форматов файлов, основанных на XML.
- Журналирование исключений в журнал Celery.
- Улучшена производительность надстроек, работающих с репозиторием.
- Улучшен внешний вид электронных писем уведомлений.
- Исправлено поведение при сбросе пароля.
- Улучшена производительность на большинстве страниц перевода.
- Исправлен список языков, неизвестных Weblate'у.
- Добавлена поддержка клонирования надстроек в обнаруженные компоненты.

- Добавлена поддержка замены содержимого файла загруженным файлом.
- Добавлена поддержка перевода содержимого не из системы контроля версий.
- Добавлен виджет изображения OpenGraph для его использования в социальных сетях.
- Добавлена поддержка анимированных снимков экрана.
- Улучшена обработка одноязычных файлов XLIFF.
- Избегание отправки нескольких уведомлений на одно событие.
- Добавлена поддержка фильтра по изменениям.
- Расширенные предопределённые периоды отчётности.
- Добавлена поддержка веб-обработчиков для репозиториев Azure Repos.
- Новые необязательные уведомления об ожидающих предложениях или непереведённых строках.
- В письма с уведомлениями добавлена ссылка для отписки от них в одно нажатие.
- Исправлены ложные срабатывания в проверке «Уже переведено».
- Новый интерфейс управления для администраторов.
- Приоритет строки теперь можно задать с помощью флагов.
- Добавлены представления управления языком.
- Добавлены проверки на формат строк для библиотеки Qt и Ruby.
- Добавлена конфигурация, лучше подходящая для установок с единственным проектом.
- Уведомление о появлении новой строки при изменении исходной в одноязычном переводе.
- Добавлено отдельное представление для памяти переводов с возможностью поиска.

4.20.12 Weblate 3.7.1

Выпущен 28 июня 2019 года.

- Обновление документации.
- Исправлены некоторые ограничения для версий зависимостей.
- Обновлена база данных языков.
- Обновления локализации.
- Различные мелкие исправления пользовательского интерфейса.
- Улучшена обработка неподдерживаемых, но обнаруженных файлов перевода.
- Более подробный отчёт о не удовлетворённых требованиях к формату файлов.

4.20.13 Weblate 3.7

Выпущен 21 июня 2019 года.

- Добавлена отдельная очередь Celery для уведомлений.
- Использование согласованного с приложением вида для просмотра API.
- В отчёты включена статистика по утверждённым переводам.
- Сообщение о прогрессе при обновлении компонента перевода.
- Разрешено прерывать запущенное фоновое обновление компонентов.
- Расширение языка шаблонов для манипуляций именами файлов.

- Использование шаблонов для ссылки на редактор и адреса браузера репозитория.
- Показ максимальной длины и текущего количества символов при редактировании перевода.
- В проверке «Перевод без изменения» улучшена обработка аббревиатур.
- Обновлена страница входа новых участников.
- Добавлена поддержка настройки надстройки msgmerge.
- При отправке уведомлений добавлена задержка открытия SMTP-соединения.
- Улучшено журналирование ошибок.
- В надстройке генерирования файлов MO разрешено использование пользовательских расположений.
- Добавлены надстройки для очистки старых предложений или комментариев.
- Добавлен параметр для включения горизонтального режима в Дзен-редакторе.
- Повышена производительность импорта множества взаимосвязанных компонентов.
- Исправлены примеры установки, бывшие в некоторых случаях неправильными.
- Улучшено отображение предупреждений в изменениях.
- Добавлен новый виджет горизонтальной статистики.
- Улучшена проверка строк формата для форм множественного числа.
- Добавлен инструмент управления шрифтами.
- Новая проверка размеров отрисовываемого текста.
- Добавлена поддержка форматов субтитров.
- Включена общая статистика по языкам.
- Добавлены отчёты на уровне проекта и на глобальном уровне.
- Улучшен пользовательский интерфейс при отображении статуса перевода.
- Новая цветовая схема и логотип Weblate'а.
- Новый внешний вид растровых плашек.

4.20.14 Weblate 3.6.1

Выпущен 26 апреля 2019 года.

- Улучшена обработка одноязычных файлов XLIFF.
- Исправлены выдержки с уведомлениями в некоторых граничных случаях.
- Исправлено предупреждение об ошибке скрипта надстройки.
- Исправлено создание файлов MO для одноязычных файлов PO.
- Исправлено отображение не установленных проверок.
- Обозначение управляемых проектов в списке проектов.
- Разрешено обновление для восстановления из отсутствующего репозитория системы контроля версий.

4.20.15 Weblate 3.6

Выпущен 20 апреля 2019 года.

- Добавлена поддержка скачивания данных пользователя.
- Теперь надстройки автоматически запускаются при установке.
- Улучшены инструкции по разрешению конфликтов слияния.
- Надстройка очистки теперь совместима с переводами метаданных App store.
- Настраиваемый синтаксис кода языка при добавлении новых переводов.
- Предупреждение об использовании Python'а 2 в связи с запланированным прекращением его поддержки в апреле 2020 года.
- Извлечение специальных символов из исходной строки для их показа и быстрой вставке при переводе.
- Расширенная статистика по участникам, отражающая как количество источников, так и количество переводов.
- Администраторы и надстройка проверки согласованности теперь могут добавлять переводы, даже если они отключены для пользователей.
- Исправлено описание переключателя отключения манипуляции заголовком `Language-Team`.
- Уведомление пользователей, упомянутых в комментариях.
- Из настройки компонента удалено автоопределение формата файла.
- Исправлено создание файлов MO для одноязычных файлов PO.
- Добавлены выдержки с уведомлениями.
- Добавлена поддержка заглушения уведомлений компонента.
- Добавлены уведомления о новых предупреждениях, сообщениях на досках сообщений или компонентах.
- Теперь можно настроить уведомления для управляемых проектов.
- Улучшена обработка трёхбуквенных кодов языков.

4.20.16 Weblate 3.5.1

Выпущен 10 марта 2019 года.

- Исправлен пример Celery-модуля systemd.
- Исправлены уведомление из HTTP-репозиториев с логинами.
- Исправлены состояния гонки при редактировании исходных строк для одноязычных переводов.
- При неудачном выполнении надстроек их вывод включается в логи.
- Улучшена проверка настроек при добавлении нового языка.
- Разрешено редактирование формата в настройках компонента.
- Обновлены инструкции по установки: теперь предпочтительная версия Python 3.
- Улучшения производительности и последовательности при загрузке переводов.
- Терминологическая служба Майкрософт сделана совместимой с текущими выпусками Zeep.
- Обновления локализации.

4.20.17 Weblate 3.5

Выпущен 3 марта 2019 года.

- Улучшена производительность встроенной памяти переводов.
- Добавлен интерфейс управления глобальной памятью переводов.
- Улучшена посылка предупреждений при плохом состоянии компонента.
- Добавлен интерфейс управления совместной доской сообщений.
- Теперь можно настраивать сообщения коммитов в надстройках.
- Уменьшено количество коммитов при обновлении вышестоящего репозитория.
- Исправлены возможные потери метаданных во время перемещения компонента между проектами.
- Улучшена навигация в Дзен-режиме.
- Добавлены несколько новых проверок качества (для Markdown и URL).
- Добавлена поддержка файлов метаданных app store.
- Добавлена поддержка переключения между интеграцией с GitHub или Gerrit.
- Добавлена проверка на буквы кашиды.
- Добавлен параметр для уплотнения коммитов на основе их авторства.
- Улучшена поддержка формата файла XLSX.
- Совместимость с Tesseract 4.0.
- Надстройка «Тарифный план» теперь удаляет проекты неоплачиваемые в течении 45 дней.

4.20.18 Weblate 3.4

Выпущен 22 января 2019 года.

- Добавлена поддержке заполнителей XLIFF.
- Celery теперь может поддерживать множество очередей заданий.
- Добавлена поддержка переименования и перемещения проектов и компонентов.
- В отчёты включено количество символов.
- Добавлена возможность управляемого добавления компонентов перевода при использовании автоматического обнаружения файлов перевода.
- Настраиваемые сообщения merge-коммитов для Git.
- Добавлена визуальная индикация предупреждений в компонентах при навигации.
- Улучшена производительность загрузки файлов перевода.
- Новая надстройка для уплотнения коммитов перед их отправкой.
- Улучшено отображение изменений перевода.
- Стиль слияния по умолчанию изменён на «rebase». Также теперь его можно настроить.
- Улучшена обработка нестандартных подтегов в кодах языка.
- Улучшена производительность обновления полнотекстовых индексов.
- Расширен API загрузки файлов. Теперь он поддерживает больше параметров.

4.20.19 Weblate 3.3

Выпущен 30 ноября 2018 года.

- Добавлена поддержка удаления компонентов и проектов.
- Улучшена производительность для некоторых одноязычных переводов.
- Добавлены предупреждения для компонентов перевода, чтобы подсвечивать проблемы с переводом.
- Показывать имя ресурса строки XLIFF в качестве контекста (если оно доступно) .
- Добавлена поддержка состояний XLIFF.
- Добавлена поддержка файлов защищённых от записи в DATA_DIR.
- Улучшен экспорт изменений в CSV.

4.20.20 Weblate 3.2.2

Выпущен 20 октября 2018 года.

- Удаление зависимости от Babel, т.к. она более не нужна.
- Обновлены определения языков.
- Улучшена документация по надстройкам, LDAP и Celery.
- Исправлено включение флагов «конец строк в формате DOS» и «автоматический формат сообщений java».
- Исправлен запуск тестов setup.py из пакета PyPi.
- Улучшена обработка строк со множественным числом.
- Исправлены проблемы с API загрузки переводов в некоторых крайних случаях.
- Исправлено обновление настроек Git, если они не были изменеы вручную.

4.20.21 Weblate 3.2.1

Выпущен 10 октября 2018 года.

- Задокументирована зависимость от backports.csv при работе на Python 2.7.
- Исправлен запуск тестов от root.
- Улучшена обработка ошибок в модуле gitexport.
- Исправлено отображение прогресса для вновь добавленных языков.
- Корректно сообщать об ошибках в рабочих процессах Celery в Sentry.
- Исправлено создание новых переводов с помощью Qt Linguist.
- Исправлены периодические ошибки обновления полнотекстовых индексов.
- Улучшена валидация при создании новых компонентов.
- Добавлена поддержка подчистки старых предложений.

4.20.22 Weblate 3.2

Выпущен 6 октября 2018 года.

- Добавлена команда управления install_addon для автоматизации установки надстроек.
- Сделана доступной более тонкая настройка ограничения частоты запросов.
- Добавлена поддержка экспорта и импорта файлов Excel.
- Улучшена очистка компонентов, если надстройка обнаруживает множество компонентов.
- Переписан бэкэнд машинного перевода для Microsoft Terminology.
- Weblate теперь использует Celery и разгружает на него часть обработки.
- Улучшены возможности поиска и добавлен поиск по регулярным выражениям.
- Добавлена поддержка машинного перевода с помощью Youdao Zhiyun API.
- Добавлена поддержка машинного перевода с помощью Baidu API.
- Интегрированы задачи по поддержке работоспособности и очистки с помощью Celery.
- Улучшена производительность загрузки переводов почти на 25%.
- Удалена поддержка слияния заголовков при загрузках.
- Удалена поддержка настраиваемых сообщений коммитов.
- Настраиваемые режимы редактирования (дзен/полный).
- Добавлена поддержка сообщений об ошидках с помощью Sentry.
- Добавлена поддержка автоматических ежедневных обновлений репозиториев.
- Добавлена возможность создания проектов и компонентов пользователями.
- Встроенная память переводов теперь автоматически сохраняет сделанные переводы.
- Пользователи и проекты могут импортировать свои уже существующие памяти переводов.
- Улучшено управление связанными строками скриншотов.
- Добавлена поддержка проверки Java MessageFormat.

См. подробный список исправленных проблем в описании целей версии 3.2 на GitHub.

4.20.23 Weblate 3.1.1

Выпущен 27 июля 2018 года.

- Исправлены ошибки в тестах, выдаваемых на некоторых установках.

4.20.24 Weblate 3.1

Выпущен 27 июля 2018 года.

- Обновления с версий более старых, чем 3.0.1, не поддерживаются.
- Разрешено переопределение сообщений коммитов через настройки.
- Улучшена совместимость web-обработчиков с Weblate, устанавливаемом на собственном хостинге.
- Добавлена поддержка Amazon Translate.
- Совместимость с Django 2.1.
- Проверки системы Django теперь используются для диагностики проблем с установкой.
- Удалена поддержка сервиса libravatar, который скоро прекращает работу.

- Новая надстройка, позволяющая помечать не изменённые переводы «на правку».
- Добавлена поддержка перепрыгивания в указанное место при переводе.
- Скачанные переводы теперь могут быть настроены.
- Улучшен алгоритм вычисления похожести строк при сопоставлении с памятью переводов.
- Добавлена поддержка подписи git-коммитов с помощью GnuPG.

4.20.25 Weblate 3.0.1

Выпущен 10 июня 2018 года.

- Исправлены возможные проблемы перехода с версии 2.20.
- Обновления локализации.
- Удалены устаревшие примеры обработчиков (hooks).
- Улучшено кеширование документации.
- Исправлено отображение документации для администраторов.
- Исправлена обработка длинных названий языков.

4.20.26 Weblate 3.0

Выпущен 1 июня 2018 года.

- Переписан контроль доступа.
- Подчистка кода в некоторых местах, что привело к перемещению и переименованию модулей.
- Новая надстройка для автоматического обнаружения компонентов.
- Команда управления import_project теперь принимает несколько иные параметры.
- Добавлена базовая поддержка файлов ресурсов Windows.
- Новая надстройка, которая позволяет сохранять имена участников в заголовках PO-файлов.
- Удалены покомпонентные скрипты-обработчики, используйте вместо этого надстройки.
- Добавлена поддержка сбора соглашений участников с правилами/лицензиями.
- Изменения прав доступа теперь отслеживаются в истории.
- Новая надстройка, которая гарантирует, что у всех компонентов и проекте будут одинаковые переводы.
- Поддержка большего количества переменных в шаблонах сообщений коммитов.
- Добавлена поддержка добавления дополнительного текстового содержимого.

4.21 Серия Weblate 2.x

4.21.1 Weblate 2.20

Выпущен 4 апреля 2018 года.

- Улучшена скорость клонирования репозиториев subversion.
- Изменена блокировка репозиториев — теперь используются сторонние библиотеки.
- Добавлена поддержка скачивания только строк, для которых требуются действия.
- Добавлена поддержка поиска по нескольким языковым кодам одновременно.

- Новая надстройка, позволяющая настроить правила переноса строк в выходных gettext-файлах.
- Новая надстройка для настройки форматирования JSON.
- Добавлена поддержка аутентификации «на предъявителя» для API, совместимая с RFC 6750.
- Добавлена поддержка автоматического перевода с помощью различных сервисов машинного перевода.
- Добавлена поддержка HTML-разметки в сообщениях на совместной доске.
- Добавлена поддержка массового изменения состояния строк.
- Теперь необходим Translate-toolkit 2.3.0 или новее; старые версии более не поддерживаются.
- Добавлена встроенная память переводов.
- Добавлен обзор списка компонентов и страницы списков покомпонентного обзора.
- Добавлена поддержка машинного перевода с помощью DeepL.
- Результаты машинного перевода теперь кешируются внутри Weblate.
- Добавлена поддержка переупорядочивания зафиксированных изменений.

4.21.2 Weblate 2.19.1

Выпущен 20 февраля 2018 года.

- Исправлены проблемы с миграцией при обновлении с 2.18.
- Улучшена валидация API загрузки файлов.

4.21.3 Weblate 2.19

Выпущен 15 февраля 2018 года.

- Исправлен импорт некоторых форматов файлов.
- Отображать информацию о браузере в журнале аудита в формате удобным для человека.
- Добавлен файловый экспорт в TMX.
- Различные улучшения производительности загрузки файлов перевода.
- Добавлена возможность отключить управление доступом в самос Weblate в пользу управления доступом в Django.
- Улучшена производительность глоссария на больших строках.
- Совместимость с django_auth_ldap 1.3.0.
- Ошибки в конфигурации теперь сохраняются и сообщения о них поступают постоянно.
- Флаги игнорирования теперь уважаются при автоматических исправлениях пробелов.
- Улучшена совместимость с некоторыми установками Subversion.
- Улучшен встроенный сервис машинного перевода.
- Добавлена поддержка SAP Translation Hub.
- Добавлена поддержка терминологической службы Майкрософт.
- Удалена поддержка рекламы в уведомлениях по e-mail.
- Улучшено составление отчётов о прогрессе переводов на уровне языков.
- Улучшена поддержка различных форм множественного числа.
- Добавлена поддержка репозиториев Subversion, которые не используют stdlayout.
- Добавлены надстройки для изменения рабочего процесса переводчиков.

4.21.4 Weblate 2.18

Выпущен 15 декабря 2017 года.

- Улучшен сбор и отображение статистики участников.
- Улучшена настройка виртуальной клавиатуры со специальными символами.
- Добавлена поддержка формата файлов DTD.
- Изменены горячие клавиши так, чтобы они с меньшей вероятностью конфликтовали с горячими клавишами браузера/ОС.
- Улучшена поддержка флага «Одобрено» в файлах XLIFF.
- Добавлена возможность не переносить длинные строки в PO-файлах gettext.
- Добавлена кнопка для копирования постоянной ссылки на текущую переводимую строку.
- Прекращена поддержка Django 1.10 и добавлена поддерживается Django 2.0.
- Удалена блокировка переводимых строк во время перевода.
- Добавлена поддержка добавления новых строк к одноязычным переводам.
- Добавлена поддержка рабочего процесса перевода с выделенными рецензентами.

4.21.5 Weblate 2.17.1

Выпущен 13 октября 2017 года.

- Исправлен запуск тестов в некоторых специфических ситуациях.
- Обновления локалей.

4.21.6 Weblate 2.17

Выпущен 13 октября 2017 года.

- Теперь Weblate по умолчанию делает поверхностные (shallow) Git-клоны.
- Улучшена производительность при обновлении больших файлов перевода.
- Добавлена поддержки блокировки регистрации с определённых адресов e-mail.
- Пользователи теперь могут удалять свои собственные комментарии.
- Добавлена возможность предварительтельного просмотра при использовании «найти и заменить».
- Настройки форм поиска и загрузки теперь постоянны для каждого клиента.
- Расширены возможности поиска.
- Более тонкая попроектная настройка ACL.
- Изменено значение по умолчанию для BASE_DIR.
- Добавлен второй, дополнительный, шаг при удалении аккаунта, дабы предупредить случайные удаления.
- Теперь можно редактировать настройки доступа к проекту.
- Добавлена опционально защита от спама в предложениях с помощью Akismet.

4.21.7 Weblate 2.16

Выпущен 11 августа 2017 года.

- Различные улучшения производительности.
- Добавлена поддержка вложенного формата JSON.
- Добавлена поддержка формата WebExtension JSON.
- Исправлена аутентификация при экспорте в git.
- Улучшен импорт CSV в некоторых случаях.
- Улучшен внешний вид виджета «Другие переводы».
- Проверки максимальной длины текста теперь принуждают соблюдать её прямо в форме.
- Время ожидания коммита теперь можно настроить покомпонентно.
- Различные подчистки пользовательского интерфейса.
- Исправлен поиск по компоненту/проекту/сайту.

4.21.8 Weblate 2.15

Выпущен 30 июня 2017 года.

- Теперь показывается большее количество связанных переводов в «других переводах».
- Добавлена возможность просматривать переводы текущей строки на другие языки.
- Теперь для литовского языка по умолчанию используются 4 формы множественного числа.
- Исправлена загрузка различных форматов одноязычных файлов перевода.
- Улучшены сообщения об ошибках при неудачной аутентификации.
- При удалении слов из глоссария состояние страницы теперь сохраняется.
- Добавлена прямая ссылка на редактирования вторичных языков перевода.
- Добавлены проверки качества для форматирования строк Perl.
- Добавлена поддержка отклонения повторно используемых паролей.
- Расширена панель инструментов для редактирования RTL-языков (с письмом справа налево).

4.21.9 Weblate 2.14.1

Выпущен 24 мая 2017 года.

- Исправлена возможная ошибка при пагинации результатов поиска.
- Исправлена миграция со старых версий в некоторых крайних случаях.
- Исправлена возможная межсайтовая подделка запроса (CSRF) при отметке проектов для отслеживания или снятия этой метки.
- Смена пароля более не аутентификация пользователя.
- Исправлена возможный обход CAPTCHA при забытом пароле.

4.21.10 Weblate 2.14

Выпущен 17 мая 2017 года.

- Добавление записей в глоссарий через AJAX-запросы.
- Разлогинивание теперь делается через POST-запрос, дабы избежать межсайтовой подделки запросов (CSRF).
- Сброс API-токена теперь делается через POST-запрос, дабы избежать межсайтовой подделки запросов (CSRF).
- Weblate теперь устанавливает политику безопасности контента (Content-Security-Policy) по умолчанию.
- URL локального редактора теперь проверяется, дабы предотвратить self-XSS атаку.
- Пароль теперь по умолчанию проверяется по списку обычных уязвимостей.
- Уведомления пользователей о важных действиях с их аккаунтами, как смена пароля.
- Экспорт в CSV теперь экранирует потенциальные формулы.
- Различные незначительные улучшения безопасности.
- Попытки аутентификации теперь ограничены по частоте.
- Содержимое предложений теперь сохраняется в истории.
- Сохранение важных действий с аккаунтов в журнале аудита.
- Запрос подтверждения пароля при удалении аккаунта или добавлении новых ассоциаций.
- Теперь показывается время, когда было сделано какое-либо предложение.
- Добавлена новая проверка качества для завершающей точки с запятой.
- Теперь гарантируется, что ссылками на поиск можно делиться.
- В API добавлена информация об исходных строках и снимках экрана.
- Теперь разрешена перезапись переводов при загрузке через API.

4.21.11 Weblate 2.13.1

Выпущен 12 апреля 2017 года.

- Исправлено формирование списка управляемых проектов в профиле.
- Исправлены проблемы с миграцией, из-за которых пропадали некоторые разрешения.
- Исправлено формирование списка текущих форматов файлов на странице скачивания переводов.
- Теперь возвращается ошибка HTTP 404, когда пользователь пытается получить доступ к проекту, на который у него нет достаточных привилегий.

4.21.12 Weblate 2.13

Выпущен 12 апреля 2017 года.

- Исправлены проверки качества в шаблонах перевода.
- Добавлены проверки качества, которые запускаются при потере переводов.
- Добавлена возможность просматривать текущие предложения конкретного пользователя.
- Добавлена возможность автоматически создавать списки компонентов.
- Теперь можно настроить панель управления по умолчанию для неаутентифицировавшихся пользователей.

- Добавлена возможность просмотреть 25 случайных строк требующих рецензирования.
- История теперь отражает изменения строк.
- Улучшено отображение ошибок при добавлении новых переводов.
- Добавлен поиск по конкретному языку внутри одного проекта.
- Групповые списки контроля доступа (Group ACL) теперь можгут быть ограничены рамками конкретного проекта.
- Попроектные ACL теперь реализованы через групповые списки контроля доступа (Group ACL) .
- Добавлена возможность более тонкого контроля привилегий.
- Различные мелкие улучшения пользовательского интерфейса.

4.21.13 Weblate 2.12

Выпущен 3 марта 2017 года.

- Улучшен интерфейс администратора для групп.
- Добавлена поддержка API Яндекс Переводчика.
- Улучшена скорость поиска по всему сайту.
- Добавлен поиск по проекту и по компоненту.
- Добавлена операция «поиск и замена» по проекту и по компоненту.
- Улучшено отображение противоречивых переводов.
- Добавлена поддержка открытия файлов с исходным кодом в локальном редакторе.
- Добавлена поддержка добавления специальных символов на виртуальную клавиатуру.
- Улучшено управление снимками экрана: добавлена поддержка распознавания текста для сопоставления их с исходными строками.
- В сообщения коммитов по умолчанию теперь включается ссылка и информация о переводе.
- Добавлена поддержка форматов переводов Joomla.
- Улучшена надёжность при импорте разных форматов.

4.21.14 Weblate 2.11

Выпущен 31 января 2017 года.

- Теперь на странице языка располагается детальная информация о языке.
- Улучшена реализация поддержки Mercurial.
- Добавлена возможность задавать приоритет перевода компонентов.
- Более единообразное использование групповых списков контроля доступа (Group ACL) даже с более редкоиспользуемыми разрешениями.
- Добавлена переменная WL_BRANCH для скриптов-обработчиков.
- Улучшена документация для разработчиков.
- Улучшена совместимость с разными версиями Git в надстройке для экспорта в Git.
- Добавлена попроектная и покомпонентная статистика.
- Добавлены отображения кодов языков для улучшения поддержки API перевода Microsoft.
- Работа по полнотекстовому удалению перемещена в фоновую задачи ради ускорения удаления переводов.

- Исправлено отображение исходных строк с формами множественного числа для языков с единственной формой.
- Улучшена обработка ошибок при импорте проекта.
- Различные улучшения производительности.

4.21.15 Weblate 2.10.1

Выпущен 20 января 2017 года.

- Исправлена утечка информации о существовании аккаунта в форме сброса пароля (CVE-2017-5537).

4.21.16 Weblate 2.10

Выпущен 15 декабря 2016 года.

- Добавлена проверка качества, которая проверяет, что формы множественного числа переведены по-разному.
- Исправлены обработчики для GitHub для репозиториев, требующих аутентификацией.
- Добавлен опциональный модуль для экспорта в Git.
- Поддержка API переводчика Microsoft Cognitive Services.
- Упрощён пользовательский интерфейс проектов и компонентов.
- Добавлен автоматический корректор для удаления управляющих символов.
- В проекты добавлен обзор каждого отдельного языка.
- Добавлена поддержка экспорта в CSV.
- Добавлено скачивание статичтики в формате CSV.
- Добавлен «просмотр в виде матрицы», для быстрого обзора всех переводов.
- Добавлен базовый API для изменений и строк.
- Добавлена поддержка машинного перевода через сервера Apertium APy.

4.21.17 Weblate 2.9

Выпущен 4 ноября 2016 года.

- Расширены параметры для команды управления createadmin.
- Расширена команда import_json так, чтобы она могла корректно обрабатывать существующие компоненты.
- Добавлена поддержка YAML-файлов.
- Владельцы проектов теперь могут настраивать компоненты перевода и изменять детали проекта.
- Термин «Подписка» (на проекты) заменён на «Отслеживание» (проектов).
- Проекты теперь можно отмечать для отслеживания прямо с их страницы.
- Добавлен многоязычный виджет статуса.
- Дополнительные языки теперь подсвечиваются, если не показывается исходная строка.
- Удаление предложений теперь сохраняется в истории.
- Улучшен интерфейс выбора языка в профиле.
- Исправлен показ сообщений совместной доски для компонентов.

- Теперь вкладка настроек остаётся открытой после их сохранения.
- Теперь коментарии к исходным строкам отображаются более заметно.
- Теперь вспомогательный модуль для слияния PO-файлов Gettext в git репозиториях устанавливается автоматически.
- Добавлен инструмент «поиск и замена».
- Добавлена поддержка загрузки визуального контента (снимков экрана) связанного с переводами.

4.21.18 Weblate 2.8

Выпущен 31 августа 2016 года.

- Улучшения в документации.
- Переводы.
- Обновлены библиотеки javascript идущие в комплекте с Weblate.
- Добавлена команда управления list_translators.
- Django 1.8 более не поддерживается.
- Исправлена совместимость с Django 1.10.
- Добавлена поддержка Subversion.
- Проверка корректности XML отделена от проверки на то что теги XML не совпадают.
- API теперь принимает во внимание параметр HIDE_REPO_CREDENTIALS.
- Теперь в Дзен-режиме показываются изменения в исходной строке.
- Alt+PageUp/PageDown/Home/End теперь работают и в Дзен-режиме.
- Добавлена всплывающая подсказка, отображающая точное время изменений.
- Добавлена возможность выбирать фильтры и начинать поиск прямо со страницы перевода.
- Добавлено удаление переводов через пользовательский интерфейс.
- Улучшено поведение при вставке маркеров-заполнителей.
- Исправлены проблемы с автоматической блокировкой в Дзен-режиме.

4.21.19 Weblate 2.7

Выпущен 10 июля 2016 года.

- Удалена возможность машинного перевода с помощью Google web translate.
- Улучшены сообщения коммитов при добавлении перевода.
- Исправлен API Google Translate для иврита.
- Совместимость с Mercurial 3.8.
- Добавлена команда управления import_json.
- Корректное упорядочивание новых переводов.
- Теперь также показывается полный текст предложения.
- Расширен API (добавлена информация по детальному статусу репозиториев, статистике, …).
- Тесты более не требуют доступ к сети, чтобы проверить, корректность работы с репозиториями.

4.21.20 Weblate 2.6

Выпущен 28 апреля 2016 года.

- Исправлена проверка компонентов с языковым фильтром.
- Улучшена поддержка файлов XLIFF.
- Исправлен машинный перевод для неанглоязычных исходных текстов.
- Добавлен REST API.
- Совместимость с Django 1.10.
- К сообщениям совместной доски добавлены категории.

4.21.21 Weblate 2.5

Выпущен 10 марта 2016 года.

- Исправлен автоматический перевод для владельцев проектов.
- Улучшенная производительность выполнения коммитов и отправки в репозиторий.
- Новая команда управления для добавления предложений из командной строки.
- Добавлена поддержка объединения комментариев при загрузке файлов.
- Добавлена поддержка некоторых расширений GNU к формату функции printf C.
- Улучшения в документации.
- Добавлена поддержка генерации списка переводчиков.
- Добавлена поддержка генерации статистики участников.
- Поиск по всему сайту теперь можно вести только по одному языку.
- Улучшение проверок качества для армянского языка.
- Добавлена поддержка создания компонентов перевода без существующих переводов.
- Поддержка добавления новых переводов при работе с Qt TS.
- Улучшена поддержка перевода PHP-файлов.
- Улучшена производительность для проверок качества.
- Исправлен поиск неудачных проверок по всему сайту.
- Добавлена возможность задать исходный язык.
- Улучшена поддержка файлов XLIFF.
- Расширен список параметров команды import_project.
- Улучшено таргетирование сообщений на совместной доске.
- Поддержка автоматического перевода строк уже переведённых в другом проекте.
- Оптимизирован полнотекстовый поисковый индекс.
- Добавлена команда управления для автоматического перевода.
- Добавлена подсветка маркеров-заполнителей.
- Добавлены горячие клавиши для заполнителей, проверок и машинных переводов.
- Улучшена блокировка переводов.
- Добавлена проверка качества интерполяций AngularJS.
- Добавлена многообещающая система списков контроля доступа, основанная на группах.

- Прояснение терминологии для строк «на правку» (ранее известных как «неточные», «fuzzy»).
- Прояснение терминологии для строк «требующих действий» и «непереведённых».
- Поддержка Python 3.
- Удалена поддержка Django 1.7.
- Удалена зависимость от msginit при создании новых PO-файлов gettext.
- Добавлена возможность настраивать вид панели управления.
- Улучшены уведомления при ушибках разбора.
- К команде import_project добавлен параметр, который позволяет импортировать компоненты с дублирующимися именами.
- Улучшена поддержка перевода PHP-файлов.
- Добавлен экспорт в XLIFF для словарей.
- Добавлен экспорт в XLIFF и gettext PO для всех переводов.
- Улучшения в документации.
- Добавлена поддержка настраиваемых автоматических групповых назначений.
- Улучшено добавление новых переводов.

4.21.22 Weblate 2.4

Выпущен 20 сентября 2015 года.

- Улучшена поддержка PHP-файлов.
- Возможность добавления списков контроля доступа (ACL) для анонимных пользователей.
- Улучшена настраиваемость команды import_project.
- Добавлена возможность делать дамп истории в CSV.
- Добавлены меры, чтобы предупредить ошибочное копирование/вставку графических отображений пробельных символов.
- Добавлена поддержка веб-обработчиков Bitbucket.
- Более жёсткий контроль неточных (fuzzy) строк при загрузке переводов.
- Некоторые адреса URL поменялись, возможно, вам придётся обновить свои закладки.
- Скрипты обработчики теперь выполняются из корневого каталога системы контроля версий.
- Скрипты-обработчики теперь имеют доступ к переменным среды, указывающим на текущий компонент.
- Добавлена команда управления для оптимизации полнотекстового индекса.
- Добавлена поддержка отправки отчётов об ошибках в Rollbar.
- У проектов теперь может быть несколько владельцев.
- Владельцы проектов теперь сами могут управлять списком владельцев проектов.
- Добавлена поддержка формата `javascript-format`, используемого в PO-файлах gettext.
- Поддержка добавления новых переводов в XLIFF.
- Улучшено автоопределения формата файлов.
- Расширены возможности горячих клавиш.
- Для некоторых языков улучшено сопоставление строк словарю.
- Улучшена вёрстка большинства страниц.

- Поддержка добавления слов в словарь во время перевода.
- Добавлена поддержка фильтрации того, какие языки будут управляться Weblate.
- Добавлена поддержка перевода и импорта CSV-файлов.
- Переписана обработка статических файлов.
- Прямые ссылки на логи/регистрацию в сторонних сервисах, если такой есть только один.
- Ожидающие изменения теперь будут коммититься при удалении аккаунта.
- Добавлена команда управления для изменения имени сайта.
- Добавлен параметр, в котором можно настроить коммитера по умолчанию.
- Добавлены обработчики, вызываемые после добавления нового перевода.
- Добавлена возможность задать несколько файлов, которые будут добавлены к коммиту.

4.21.23 Weblate 2.3

Выпущен 22 мая 2015 года.

- Прекращена поддержка Django 1.6 и миграция с помощью South.
- Поддержка добавления новых переводов при использовании файлов свойств Java.
- Добавлена возможность принимать предложения без редактирования.
- Улучшена поддержка Google OAuth 2.0.
- Добавлена поддержка файлов Microsoft .resx.
- Подправлен файл robots.txt, предлагаемый по умолчанию, так чтобы он не разрешал значительное расползание переводов.
- Упрощено принятие предложений.
- Добавлены владельцы проектов — пользователи, которые будут получать важные уведомления.
- Добавлена возможность отключить редактирование шаблонов одноязычных переводов.
- Более детализированный вид статуса репозитория.
- Прямые ссылки на редактирование шаблонов при изменении переводов.
- Теперь разрешено назначать больше привилегий для владельцев проектов.
- Теперь возможно показывать второй язык в Дзен-режиме.
- Возможность прятать исходную строку в пользу вторичного языка.

4.21.24 Weblate 2.2

Выпущен 19 февраля 2015 года.

- Улучшения производительности.
- Полнотекстовый поиск по полям «расположение» и «комментарии».
- Новый график активности на основе SVG/javascript.
- Поддержка Django 1.8.
- Поддержка удаления комментариев.
- Добавлена собственная SVG-плашка.
- Добавлена поддержка Google Analytics.
- Улучшено управление именами файлов перевода.

- Добавлена поддержка одноязычных JSON-переводов.
- Блокировка компонентов теперь сохраняется в истории.
- Поддержка редактирования исходного (шаблонного) языка одноязычных переводов.
- Добавлена базовая поддержка Gerrit.

4.21.25 Weblate 2.1

Выпущен 5 декабря 2014 года.

- Добавлена поддержка репозиториев Mercurial.
- Шрифт Glyphicon заменён на Awesome.
- Добавлены иконки для сервисов аутентификации через социальные сети.
- Цвета кнопок и иконки теперь более последовательны и согласованы.
- Улучшения в документации.
- Исправления различных ошибок.
- Автоматическое скрытие дополнительных столбцов в списке переводов на маленьких экранах.
- Изменена конфигурация путей файлов.
- Улучшено управление и хранение SSH-ключей.
- Улучшена блокировка репозиториев.
- Добавлены настраиваемые проверки качества для исходных строк.
- Добавлена возможность прятать завершённые переводы с панели управления.

4.21.26 Weblate 2.0

Выпущен 6 ноября 2014 года.

- Новый отзывчивый пользовательский интерфейс, созданный с помощью Bootstrap.
- Переписан модуль работы с системами контроля версий.
- Улучшения в документации.
- Добавлена совместная доска для публикации сообщений для всего сайта.
- Добавлена возможность задавать приоритеты строк.
- Добавлена поддержка форматов файлов JSON.
- Исправлены проблемы с генерацией mo-файлов в определённых случаях.
- Добавлена поддержка уведомлений для GitLab.
- Добавлена поддержка отключения предложений для переводов.
- Поддержка Django 1.7.
- У проектов с ACL теперь есть управление пользователями.
- Расширены возможности поиска.
- Предоставление большего количества подсказок переводчикам о множественном числе.
- Исправлена блокировка Git-репозиториев.
- Совместимость с более старыми версиями Git.
- Улучшена поддержка списков контроля доступа (ACL).

- Добавлены кнопки кавычек специфичные для конкретных языков и другие специальные символы.
- Добавлена поддержка экспорта статистики в JSONP.

4.22 Серия Weblate 1.x

4.22.1 Weblate 1.9

Выпущен 6 мая 2014 года.

- Совместимость с Django 1.6.
- Совместимость с Django 1.4 более не поддерживается.
- Команды управления для блокировки/разблокировки переводов.
- Улучшена поддержка TS-файлов (Qt).
- Пользователи теперь могут удалять свои аккаунты.
- Аватары теперь могут быть отключены.
- Атрибуты «первое имя» и «второе имя» объединены в один.
- Аватары теперь загружаются и кешируются на стороне сервера.
- Добавлена поддержка плашки shields.io.

4.22.2 Weblate 1.8

Выпущен 7 ноября 2013 года.

- Проверьте инструкции по обновлению в руководстве.
- Более приятный вид сводки проектов.
- Улучшен внешний вид ссылок на социальные сети, по которым можно поделиться переводом.
- Больше контроля над анонимными пользователями.
- Поддержка входа через сторонние сервисы, см. подробности в руководстве.
- Теперь пользователи могут совершать вход и по адресу электронной почты вместо имени пользователя.
- Улучшения в документации.
- Улучшен процесс рецензирования исходных строк.
- Поиск по всем строкам.
- Улучшено отслеживание исходных строк.
- Защита регистрации с помощью каптчи.

4.22.3 Weblate 1.7

Выпущен 7 октября 2013 года.

- Проверьте инструкции по обновлению в руководстве.
- Проверка форматирования строк для Python с фигурными скобками.
- Покомпонентная настройка проверок качества.
- Детальная статистика по каждому переводу.
- Изменён способ связывания предложений, проверок и комментариев со строками.

- Пользователи теперь могут добавлять свой текст к сообщениям коммитов.
- Поддержка подписки на запросы на добавление новых языков.
- Поддержка добавления новых переводов.
- Виджеты и диаграммы теперь отрисовываются с помощью Pillow вместо Pango + Cairo.
- Добавлен виджет плашки статуса.
- Удалены неверные проверки направления письма в тексте.
- Изменения в словаре теперь логируются в истории.
- Улучшения производительности страницы перевода.

4.22.4 Weblate 1.6

Выпущен 25 июля 2013 года.

- Улучшена обработка ошибок при регистрации.
- Просмотр изменений.
- Исправлена сортировка предложений машинного перевода.
- Улучшена поддержка машинного перевода MyMemory.
- Добавлена поддержка машинного перевода Amagama.
- Различные оптимизации часто используемых страницах.
- Выделение искомых фраз в результатах поиска.
- Поддержка автоматических исправлений при сохранении сообщения.
- Отслеживание истории переводов и возможность откатиться к более старой версии.
- Добавлена поддержка Google Translate API.
- Добавлена поддержка управления SSH-ключами серверов.
- Разнообразные улучшения валидации форм.
- Разнообразные улучшения проверок качества.
- Улучшена производительность при импорте.
- Добавлена поддержка голосований за предложения.
- Подчищен интерфейс администратора.

4.22.5 Weblate 1.5

Выпущен 16 апреля 2013 года.

- Проверьте инструкции по обновлению в руководстве.
- Добавлены общедоступные страницы пользователей.
- Улучшено описания в формах редактирования строк со множественным числом.
- Добавлена поддержка экспорта глоссария в TBX.
- Добавлена поддержка уведомлений Bitbucket.
- Графики активности теперь доступны для каждого отдельного перевода, языка и пользователя.
- Расширены параметры команды администратора import_project.
- Совместимость с Django 1.5.

- Аватары теперь показываются с помощью libravatar.
- Добавлена возможность экспорта в JSON в формате удобном для человека.
- Различные улучшения производительности.
- Отображение количества неудачных проверок и неточных строк в индикаторах готовности перевода также и для проектов и языков.
- Добавлена поддержка добавления своих предкоммитных обработчиков и коммита дополнительных файлов.
- Переписан поиск ради лучшей производительности и удобства пользователя.
- Новый интерфейс машинного перевода.
- Добавлена поддержка перевода одноязычных PO-файлов.
- Расширено количество кешируемых метаданных ради увеличения скорости разнообразного поиска.
- Теперь отображается также и количество слов.

4.22.6 Weblate 1.4

Выпущен 23 января 2013 года.

- Исправлено удаление комментариев/строк при удалении соответствующих строк.
- Добавлен параметр для отключения автоматического распространения переводов.
- Добавлена возможность подписываться на уведомления о неудачном слиянии (в системах контроля версий).
- Теперь проекты, которым нужен особый ttkit-загрузчик, импортируются корректно.
- Добавлена карта сайта, чтобы облегчить поисковым ботам доступ ко всем страницам.
- Добавлены прямые ссылки на строки в уведомления по электронной почте и ленты.
- Различные улучшения в интерфейсе администратора.
- В интерфейс администратора добавлены подсказки по настройке Weblate для рабочей среды.
- Добавлены виджеты и страница взаимодействия для отдельных языков.
- Улучшена обработка блокировки переводов.
- Теперь отображается больше отрывков кода для вставки виджетов на сторонние сайты.
- Отображение количества неудачных проверок и неточных строк в индикаторах готовности перевода.
- Больше опций для форматирования сообщений коммитов.
- Исправлена обработка ошибок сервисов машинного перевода.
- Улучшено поведение блокировки при автоматическом переводе.
- Поддержка показа изменений относительно предыдущей исходной строки.
- Добавлена поддержка поиска подстрок.
- Разнообразные улучшения проверок качества.
- Добавлена поддержка попроектных списков контроля доступа (ACL).
- Базовое покрытие кодовой базы юнит-тестами.

4.22.7 Weblate 1.3

Выпущен 16 ноября 2012 года.

- Возможность работы поверх PostgreSQL.
- Удаление языков, удалённых в вышестоящем git-репозитории.
- Улучшена обработка проверок качества.
- Добавлены новые проверки (для BB-кода, разметки XML и переводов строк).
- Поддержка возможности смены базы (rebase) вместо слияния (merge).
- Возможность переместить Weblate (запускать его установленным по другому пути, например, */weblate*).
- Возможность выбрать тип файла вручную, если автоопределение с этим не справилось.
- Улучшена поддержка ресурсов Android.
- Поддержка генерации SSH-ключей из веб-интерфейса.
- Экспорт данных сделан более заметным.
- Новые кнопки для ввода некоторых специальных символов.
- Поддержка экспорта словарей.
- Поддержка блокирования всей установки Weblate.
- Проверки для исходных строк и поддержка их рецензирования.
- Поддержка комментариев пользователей и для переводов и для исходных строк.
- Улучшено отслеживание журнала изменений.
- За изменениями теперь можно следить с помощью RSS.
- Улучшена поддержка языков с письмом с право налево.

4.22.8 Weblate 1.2

Выпущен 14 августа 2012 года.

- Weblate теперь использует South для миграции базы данных, посмотрите инструкции, если вы обновляетесь.
- Исправлены мелкие проблемы со связанными git-репозиториями.
- Новая страница введения для вовлечения людей в процесс перевода с помощью Weblate.
- Добавлены виджеты, которые можно использовать для продвижения в массы перевода своего проекта.
- Добавлена возможность отката (*reset*) репозитория к его состоянию в удалённом источнике (*origin*).
- Проекты или компоненты теперь могут быть заблокированы от перевода.
- Возможность отключить некоторые переводы.
- Теперь параметры добавления новых переводов можно настроить.
- Настройка git-коммитов для каждого отдельного проекта.
- Простая защита от спама.
- Улучшена вёрстка главной страницы.
- Поддержка автоматической отправки изменений при каждом коммите.
- Поддержка уведомлений по e-mail для переводчиков.
- Теперь в настройках перечисляются только используемые языки.
- Улучшена обработка неизвестных языков при импорте проекта.

- Поддержка блокировки перевода переводчиком.
- Возможность поддержки в актуальном состоянии заголовка `Language-Team` в PO-файле.
- Добавление некоторой статистики на страницу «О Weblate».
- Поддержка (и требование) django-registration 0.8.
- Кеширование количества строк с неудачными проверками.
- Проверка зависимостей во время установки.
- Улучшения в документации.

4.22.9 Weblate 1.1

Выпущен 4 июля 2012 года.

- Улучшение нескольких переводов.
- Улучшены проверки при создании компонентов.
- Добавлена поддержка совместного использования git-репозитория несколькими компонентами.
- Теперь более необязательно делается коммит при каждой попытке извлечения (*pull*) из удалённого репозитория.
- Добавлена поддержка выделения индекса в отдельный процесс.

4.22.10 Weblate 1.0

Выпущен 10 мая 2012 года.

- Улучшены проверки при добавлении/сохранении компонентов.
- Экспериментальная поддержка файлов компонентов Android (требуется патченный ttkit).
- Обновления вызванные обработчиками теперь запускаются в фоне.
- Улучшены инструкции установки.
- Улучшена навигация по словарю.

4.23 Серия Weblate 0.x

4.23.1 Weblate 0.9

Выпущен 18 апреля 2012 года.

- Исправлен импорт неизвестных языков.
- Улучшен список соседних сообщений.
- Улучшено несколько проверок.
- Обновление документации.
- Добавлены определения для ещё нескольких языков.
- Всякая разная чистка кода.
- Улучшения в документации.
- Изменение макета файла.
- Обновление вспомогательных скриптов до версии Django 1.4.

- Улучшение навигации во время перевода.
- Улучшение обработки переименования po-файлов.
- Улучшены проверки при создании компонентов.
- Полная интеграция установки в команду syncdb.
- На всех страницах перевода добавлен список последних изменений.
- Проверка не переведённых строк теперь игнорирует сообщения, содержащие только строку формата.

4.23.2 Weblate 0.8

Выпущен 3 апреля 2012 года.

- Собственный полнотекстовый поиск заменён на Whoosh.
- Различные исправления и улучшения проверок.
- Новая команда updatechecks.
- Множество обновлений переводов.
- Добавлен словарь для хранения наиболее часто используемых терминов.
- Добавлена конечная точка /admin/report/ для обзора состояния репозиториев.
- Сервисы машинного перевода больше не блокируют загрузку страницы.
- Интерфейс управления теперь также содержит полезные действия для обновления данных.
- Запись в журнал сделанных пользователями изменений.
- Возможность откладывания коммита в Git, чтобы генерировать поменьше коммитов от одного пользователя.
- Возможность просмотра неудачных проверок.
- Автоматический перевод с использованием уже переведённых строк.
- Новая страница «О Weblate» теперь показывает использованные версии программного обеспечения.
- Совместимость с Django 1.4.
- Возможность отправки изменений в удалённый репозиторий из веб-интерфейса.
- Добавлено рецензирование переводов, выполненных другими участниками.

4.23.3 Weblate 0.7

Выпущен 16 февраля 2012 года.

- Прямая поддержка GitHub'овских уведомлений.
- Добавлена поддержка очистки осиротевших проверок и переводов.
- Отображение соседних строк во время перевода.
- Отображение похожих строк во время перевода.
- Улучшен поиск строк.

4.23.4 Weblate 0.6

Выпущен 14 февраля 2012 года.

- Добавлены различные проверки для переведённых сообщений.
- Настраиваемый контроль доступа.
- Улучшена обработка переводов с новыми строками.
- Добавлена сортировка таблиц на стороне клиента.
- Если вы обновляетесь, пожалуйста, ознакомьтесь с инструкциями по обновлению.

4.23.5 Weblate 0.5

Выпущен 12 февраля 2012 года.

- **Поддержка машинного перевода с использованием следующих онлайн-сервисов:**
  - Apertium
  - Microsoft Translator
  - MyMemory
- Несколько новых переводов.
- Улучшено слияние изменений из вышестоящего репозитория.
- Улучшенная обработка одновременных git pull и перевода.
- Распространение теперь работает и для неточных изменений.
- Распространение теперь работает и для загрузок файлов.
- Исправлено скачивание файлов при использовании FastCGI (и, возможно, других протоколов).

4.23.6 Weblate 0.4

Выпущен 8 февраля 2012 года.

- В документацию добавлено руководство по использованию.
- Исправлены обработчики API, которые не требовали CSRF-защиты.

4.23.7 Weblate 0.3

Выпущен 8 февраля 2012 года.

- Улучшенное отображение исходной строки для переводов форм множественного числа.
- Новая документация в формате Sphinx.
- Отображение при переводе дополнительных языков.
- Улучшенная страница ошибок со списком существующих проектов.
- Новая статистика по языкам.

4.23.8 Weblate 0.2

Выпущен 7 февраля 2012 года.

- Улучшена проверка некоторых форм.
- Предупреждение пользователей при обновлении профиля.
- Запоминание URL-адреса для входа в систему.
- Именование областей текста при вводе форм множественного числа.
- Автоматическое расширение области перевода.

4.23.9 Weblate 0.1

Выпущен 6 февраля 2012 года.

- Начальный выпуск.

Содержание модулей Python

W

wlc, 137
wlc.config, 138
wlc.main, 138

HTTP Routing Table

/

ANY /, 88

/api

GET /api/, 90

/api/addons

GET /api/addons/, 126
GET /api/addons/(int:id)/, 126
PUT /api/addons/(int:id)/, 126
DELETE /api/addons/(int:id)/, 127
PATCH /api/addons/(int:id)/, 126

/api/changes

GET /api/changes/, 123
GET /api/changes/(int:id)/, 123

/api/component-lists

GET /api/component-lists/, 127
GET /api/component-lists/(str:slug)/, 127
POST /api/component-lists/(str:slug)/components/, 128
PUT /api/component-lists/(str:slug)/, 127
DELETE /api/component-lists/(str:slug)/, 127
DELETE /api/component-lists/(str:slug)/components/(str:component_slug), 128
PATCH /api/component-lists/(str:slug)/, 127

/api/components

GET /api/components/, 106
GET /api/components/(string:project)/(string:compo, 106
GET /api/components/(string:project)/(string:compo, 110
GET /api/components/(string:project)/(string:compo, 114
GET /api/components/(string:project)/(string:compo, 110
GET /api/components/(string:project)/(string:compo, 112
GET /api/components/(string:project)/(string:compo, 113
GET /api/components/(string:project)/(string:compo, 111
GET /api/components/(string:project)/(string:compo, 110
GET /api/components/(string:project)/(string:compo, 114
GET /api/components/(string:project)/(string:compo, 113
POST /api/components/(string:project)/(string:comp, 126
POST /api/components/(string:project)/(string:comp, 115
POST /api/components/(string:project)/(string:comp, 110
POST /api/components/(string:project)/(string:comp, 112
POST /api/components/(string:project)/(string:comp, 113
PUT /api/components/(string:project)/(string:compo, 109
DELETE /api/components/(string:project)/(string:c, 110
DELETE /api/components/(string:project)/(string:c, 115
PATCH /api/components/(string:project)/(string:com, 108

/api/groups

GET /api/groups/, 94
GET /api/groups/(int:id)/, 94
POST /api/groups/, 94
POST /api/groups/(int:id)/componentlists/, 96
POST /api/groups/(int:id)/components/, 95
POST /api/groups/(int:id)/languages/, 96

POST /api/groups/(int:id)/projects/, 96
POST /api/groups/(int:id)/roles/, 95
PUT /api/groups/(int:id)/, 95
DELETE /api/groups/(int:id)/, 95
DELETE /api/groups/(int:id)/componentlists/(int:component_list_id), 96
DELETE /api/groups/(int:id)/components/(int:component_id), 96
DELETE /api/groups/(int:id)/languages/(string:language_code), 96
DELETE /api/groups/(int:id)/projects/(int:project_id), 96
PATCH /api/groups/(int:id)/, 95

/api/languages

GET /api/languages/, 98
GET /api/languages/(string:language)/, 98
GET /api/languages/(string:language)/statistics/, 99
POST /api/languages/, 98
PUT /api/languages/(string:language)/, 99
DELETE /api/languages/(string:language)/, 99
PATCH /api/languages/(string:language)/, 99

/api/projects

GET /api/projects/, 100
GET /api/projects/(string:project)/, 100
GET /api/projects/(string:project)/changes/, 101
GET /api/projects/(string:project)/components/, 102
GET /api/projects/(string:project)/languages/, 105
GET /api/projects/(string:project)/repository/, 101
GET /api/projects/(string:project)/statistics/, 105
POST /api/projects/, 100
POST /api/projects/(string:project)/components/, 102
POST /api/projects/(string:project)/repository/, 102
PUT /api/projects/(string:project)/, 101
DELETE /api/projects/(string:project)/, 101
PATCH /api/projects/(string:project)/, 101

/api/roles

GET /api/roles/, 97
GET /api/roles/(int:id)/, 97
POST /api/roles/, 97
PUT /api/roles/(int:id)/, 97
DELETE /api/roles/(int:id)/, 98
PATCH /api/roles/(int:id)/, 97

/api/screenshots

GET /api/screenshots/, 124
GET /api/screenshots/(int:id)/, 124
GET /api/screenshots/(int:id)/file/, 124
POST /api/screenshots/, 125
POST /api/screenshots/(int:id)/file/, 124
POST /api/screenshots/(int:id)/units/, 124
PUT /api/screenshots/(int:id)/, 125
DELETE /api/screenshots/(int:id)/, 126
DELETE /api/screenshots/(int:id)/units/(int:unit_i 125
PATCH /api/screenshots/(int:id)/, 125

/api/tasks

GET /api/tasks/, 128
GET /api/tasks/(str:uuid)/, 128

/api/translations

GET /api/translations/, 115
GET /api/translations/(string:project)/(string:co 115
GET /api/translations/(string:project)/(string:co 118
GET /api/translations/(string:project)/(string:co 119
GET /api/translations/(string:project)/(string:co 119
GET /api/translations/(string:project)/(string:co 120
GET /api/translations/(string:project)/(string:co 118
POST /api/translations/(string:project)/(string:c 118
POST /api/translations/(string:project)/(string:c 119
POST /api/translations/(string:project)/(string:c 120
POST /api/translations/(string:project)/(string:c 118
DELETE /api/translations/(string:project)/(string 117

/api/units

GET /api/units/, 121
GET /api/units/(int:id)/, 121
PUT /api/units/(int:id)/, 122
DELETE /api/units/(int:id)/, 122
PATCH /api/units/(int:id)/, 122

/api/users

GET /api/users/, 91
GET /api/users/(str:username)/, 91
GET /api/users/(str:username)/notifications/, 93
GET /api/users/(str:username)/notifications/(int:subscription_id)/, 93
GET /api/users/(str:username)/statistics/, 92
POST /api/users/, 91
POST /api/users/(str:username)/groups/, 92
POST /api/users/(str:username)/notifications/, 93
PUT /api/users/(str:username)/, 92
PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 93
DELETE /api/users/(str:username)/, 92
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 94
PATCH /api/users/(str:username)/, 92
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 93

/exports

GET /exports/rss/, 132
GET /exports/rss/(string:project)/, 132
GET /exports/rss/(string:project)/(string:component)/, 132
GET /exports/rss/(string:project)/(string:component)/(string:language)/, 132
GET /exports/rss/language/(string:language)/, 132
GET /exports/stats/(string:project)/(string:component)/, 131

/hooks

GET /hooks/update/(string:project)/, 129
GET /hooks/update/(string:project)/(string:component)/, 129
POST /hooks/azure/, 130
POST /hooks/bitbucket/, 129
POST /hooks/gitea/, 130
POST /hooks/gitee/, 130
POST /hooks/github/, 129
POST /hooks/gitlab/, 129
POST /hooks/pagure/, 130

Алфавитный указатель

Символы

.XML resource file
file format, 74
--add
auto_translate command line option, 347
--addon ADDON
install_addon command line option, 353
--age HOURS
commit_pending command line option, 348
--author USER@EXAMPLE.COM
add_suggestions command line option, 347
--base-file-template TEMPLATE
import_project command line option, 351
--check
importusers command line option, 353
--config PATH
wlc command line option, 134
--config-section SECTION
wlc command line option, 134
--configuration CONFIG
install_addon command line option, 353
--convert
wlc command line option, 135
--email USER@EXAMPLE.COM
createadmin command line option, 349
--file-format FORMAT
import_project command line option, 351
--force
loadpo command line option, 354
--force-commit
pushgit command line option, 355
--format {csv,json,text,html}
wlc command line option, 134
--ignore
import_json command line option, 350
--inconsistent
auto_translate command line option, 347
--input
wlc command line option, 135
--key KEY
wlc command line option, 134
--lang LANGUAGE
loadpo command line option, 354
--language-code
list_translators command line option, 354
--language-map LANGMAP
import_memory command line option, 350
--language-regex REGEX
import_project command line option, 351
--license NAME
import_project command line option, 351
--license-url URL
import_project command line option, 351
--main-component
import_project command line option, 351
--main-component COMPONENT
import_json command line option, 350
--mt MT
auto_translate command line option, 347
--name
createadmin command line option, 349
--name-template TEMPLATE
import_project command line option, 351
--new-base-template TEMPLATE
import_project command line option, 351
--no-password
createadmin command line option, 349
--no-privs-update
setupgroups command line option, 356
--no-projects-update

setupgroups command line option, 356
--no-update
setuplang command line option, 356
--output
wlc command line option, 135
--overwrite
auto_translate command line option, 347
wlc command line option, 135
--password PASSWORD
createadmin command line option, 349
--project PROJECT
import_json command line option, 350
--source PROJECT/COMPONENT
auto_translate command line option, 347
--threshold THRESHOLD
auto_translate command line option, 347
--update
createadmin command line option, 349
import_json command line option, 350
install_addon command line option, 353
--url URL
wlc command line option, 134
--user USERNAME
auto_translate command line option, 347
--username USERNAME
createadmin command line option, 349
--vcs NAME
import_project command line option, 351

A

add_suggestions
weblate admin command, 347
add_suggestions command line option
--author USER@EXAMPLE.COM, 347
ADMINS
setting, 182
AKISMET_API_KEY
setting, 301
ALLOWED_HOSTS
setting, 182
Android
file format, 69
ANONYMOUS_USER_NAME
setting, 301
API, 87, 132, 137
Apple strings
file format, 70
ARB
file format, 73
AUDITLOG_EXPIRY
setting, 301
AUTH_LOCK_ATTEMPTS
setting, 301
AUTH_TOKEN_VALID
setting, 302
auto_translate
weblate admin command, 347
auto_translate command line option
--add, 347
--inconsistent, 347
--mt MT, 347
--overwrite, 347
--source PROJECT/COMPONENT, 347
--threshold THRESHOLD, 347
--user USERNAME, 347
AUTO_UPDATE
setting, 301
AUTOFIX_LIST
setting, 302
AVATAR_URL_PREFIX
setting, 302

B

BACKGROUND_TASKS
setting, 303
BASE_DIR
setting, 303
BaseAddon (*класс в weblate.addons.base*), 391
BASIC_LANGUAGES
setting, 304

C

can_install() (*метод класса weblate.addons.base.BaseAddon*), 391
celery_queues
weblate admin command, 348
changes
wlc command line option, 135
CHECK_LIST
setting, 304
checkgit
weblate admin command, 348
cleanup
wlc command line option, 135
cleanuptrans
weblate admin command, 348
Comma separated values
file format, 74
Command (*класс в wlc.main*), 139
COMMENT_CLEANUP_DAYS
setting, 305
commit
wlc command line option, 134
commit_pending
weblate admin command, 348
commit_pending command line option
--age HOURS, 348
COMMIT_PENDING_HOURS
setting, 305
commitgit

weblate admin command, 348
configure() (*метод weblate.addons.base.BaseAddon*), 391
createadmin
weblate admin command, 349
createadmin command line option
--email USER@EXAMPLE.COM, 349
--name, 349
--no-password, 349
--password PASSWORD, 349
--update, 349
--username USERNAME, 349
CSP_CONNECT_SRC
setting, 304
CSP_FONT_SRC
setting, 304
CSP_IMG_SRC
setting, 304
CSP_SCRIPT_SRC
setting, 304
CSP_STYLE_SRC
setting, 304
CSV
file format, 74

D

daily() (*метод weblate.addons.base.BaseAddon*), 391
DATA_DIR
setting, 305
DATABASE_BACKUP
setting, 306
DATABASES
setting, 183
DEBUG
setting, 183
DEFAULT_ACCESS_CONTROL
setting, 306
DEFAULT_ADD_MESSAGE
setting, 307
DEFAULT_ADDON_MESSAGE
setting, 307
DEFAULT_ADDONS
setting, 307
DEFAULT_AUTO_WATCH
setting, 306
DEFAULT_COMMIT_MESSAGE
setting, 307
DEFAULT_COMMITER_EMAIL
setting, 307
DEFAULT_COMMITER_NAME
setting, 308
DEFAULT_DELETE_MESSAGE
setting, 307
DEFAULT_FROM_EMAIL
setting, 183
DEFAULT_LANGUAGE
setting, 308
DEFAULT_MERGE_MESSAGE
setting, 307
DEFAULT_MERGE_STYLE
setting, 308
DEFAULT_PULL_MESSAGE
setting, 309
DEFAULT_RESTRICTED_COMPONENT
setting, 307
DEFAULT_SHARED_TM
setting, 308
DEFAULT_TRANSLATION_PROPAGATION
setting, 308
download
wlc command line option, 135
DTD
file format, 76
dump_memory
weblate admin command, 349
dumpuserdata
weblate admin command, 349

E

ENABLE_AVATARS
setting, 309
ENABLE_HOOKS
setting, 309
ENABLE_HTTPS
setting, 309
ENABLE_SHARING
setting, 309

F

file format
.XML resource file, 74
Android, 69
Apple strings, 70
ARB, 73
Comma separated values, 74
CSV, 74
DTD, 76
gettext, 63
go-i18n, 72
GWT properties, 67
i18next, 72
INI translations, 67, 68
Java properties, 66
Joomla translations, 68
JSON, 71
PHP strings, 70
PO, 63
Qt, 68
RC, 77
RESX, 74
Ruby YAML, 76
Ruby YAML Ain't Markup Language, 76
string resources, 69
TS, 68
XLIFF, 65

XML, 76
YAML, 75
YAML Ain't Markup Language, 75

G

get() (*метод wlc.Weblate*), 138
get_add_form() (*метод класса weblate.addons.base.BaseAddon*), 391
GET_HELP_URL
setting, 310
get_settings_form() (*метод weblate.addons.base.BaseAddon*), 391
gettext
file format, 63
GITHUB_CREDENTIALS
setting, 310
GITHUB_TOKEN
setting, 311
GITHUB_USERNAME
setting, 311
GITLAB_CREDENTIALS
setting, 310
GITLAB_TOKEN
setting, 310
GITLAB_USERNAME
setting, 310
go-i18n
file format, 72
GOOGLE_ANALYTICS_ID
setting, 311
GWT properties
file format, 67

H

HIDE_REPO_CREDENTIALS
setting, 311
HIDE_VERSION
setting, 311

I

i18next
file format, 72
import_demo
weblate admin command, 349
import_json
weblate admin command, 350
import_json command line option
--ignore, 350
--main-component COMPONENT, 350
--project PROJECT, 350
--update, 350
import_memory
weblate admin command, 350
import_memory command line option
--language-map LANGMAP, 350
import_project
weblate admin command, 351
import_project command line option
--base-file-template TEMPLATE, 351
--file-format FORMAT, 351
--language-regex REGEX, 351
--license NAME, 351
--license-url URL, 351
--main-component, 351
--name-template TEMPLATE, 351
--new-base-template TEMPLATE, 351
--vcs NAME, 351
importuserdata
weblate admin command, 353
importusers
weblate admin command, 353
importusers command line option
--check, 353
INI translations
file format, 67, 68
install_addon
weblate admin command, 353
install_addon command line option
--addon ADDON, 353
--configuration CONFIG, 353
--update, 353
IP_BEHIND_REVERSE_PROXY
setting, 312
IP_PROXY_HEADER
setting, 312
IP_PROXY_OFFSET
setting, 312
iPad
перевод, 70
iPhone
перевод, 70

J

Java properties
file format, 66
Joomla translations
file format, 68
JSON
file format, 71

L

LEGAL_URL
setting, 313
LICENSE_EXTRA
setting, 313
LICENSE_FILTER
setting, 313
LICENSE_REQUIRED
setting, 314
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
setting, 314
list_languages
weblate admin command, 354
list_translators
weblate admin command, 354
list_translators command line option

--language-code, 354
list_versions
weblate admin command, 354
list-components
wlc command line option, 134
list-languages
wlc command line option, 134
list-projects
wlc command line option, 134
list-translations
wlc command line option, 134
load() (*метод wlc.config.WeblateConfig*), 138
loadpo
weblate admin command, 354
loadpo command line option
--force, 354
--lang LANGUAGE, 354
LOCALIZE_CDN_PATH
setting, 314
LOCALIZE_CDN_URL
setting, 314
lock
wlc command line option, 135
lock_translation
weblate admin command, 355
lock-status
wlc command line option, 135
LOGIN_REQUIRED_URLS
setting, 314
LOGIN_REQUIRED_URLS_EXCEPTIONS
setting, 315
ls
wlc command line option, 134

M

MACHINE_TRANSLATION_SERVICES
setting, 316
main() (*в модуле wlc.main*), 138
MATOMO_SITE_ID
setting, 315
MATOMO_URL
setting, 315
move_language
weblate admin command, 355
MT_APERTIUM_APY
setting, 316
MT_AWS_ACCESS_KEY_ID
setting, 317
MT_AWS_REGION
setting, 317
MT_AWS_SECRET_ACCESS_KEY
setting, 317
MT_BAIDU_ID
setting, 317
MT_BAIDU_SECRET
setting, 317
MT_DEEPL_API_VERSION
setting, 317
MT_DEEPL_KEY
setting, 318
MT_GOOGLE_CREDENTIALS
setting, 318
MT_GOOGLE_KEY
setting, 318
MT_GOOGLE_LOCATION
setting, 318
MT_GOOGLE_PROJECT
setting, 318
MT_MICROSOFT_BASE_URL
setting, 319
MT_MICROSOFT_COGNITIVE_KEY
setting, 319
MT_MICROSOFT_ENDPOINT_URL
setting, 319
MT_MICROSOFT_REGION
setting, 319
MT_MODERNMT_KEY
setting, 319
MT_MODERNMT_URL
setting, 319
MT_MYMEMORY_EMAIL
setting, 320
MT_MYMEMORY_KEY
setting, 320
MT_MYMEMORY_USER
setting, 320
MT_NETEASE_KEY
setting, 320
MT_NETEASE_SECRET
setting, 320
MT_SAP_BASE_URL
setting, 321
MT_SAP_PASSWORD
setting, 322
MT_SAP_SANDBOX_APIKEY
setting, 321
MT_SAP_USE_MT
setting, 322
MT_SAP_USERNAME
setting, 321
MT_SERVICES
setting, 316
MT_TMSERVER
setting, 320
MT_YANDEX_KEY
setting, 321
MT_YOUDAO_ID
setting, 321
MT_YOUDAO_SECRET
setting, 321

N

NEARBY_MESSAGES
setting, 322

P

PAGURE_CREDENTIALS
    setting, 322
PAGURE_TOKEN
    setting, 323
PAGURE_USERNAME
    setting, 322
PHP strings
    file format, 70
PIWIK_SITE_ID
    setting, 315
PIWIK_URL
    setting, 315
PO
    file format, 63
post() (*метод wlc.Weblate*), 138
post_add() (*метод weblate.addons.base.BaseAddon*), 391
post_commit() (*метод weblate.addons.base.BaseAddon*), 391
post_push() (*метод weblate.addons.base.BaseAddon*), 391
post_update() (*метод weblate.addons.base.BaseAddon*), 391
pre_commit() (*метод weblate.addons.base.BaseAddon*), 392
pre_push() (*метод weblate.addons.base.BaseAddon*), 392
pre_update() (*метод weblate.addons.base.BaseAddon*), 392
pull
    wlc command line option, 134
push
    wlc command line option, 135
pushgit
    weblate admin command, 355
pushgit command line option
    --force-commit, 355
Python, 137

Q

Qt
    file format, 68

R

RATELIMIT_ATTEMPTS
    setting, 323
RATELIMIT_LOCKOUT
    setting, 323
RATELIMIT_WINDOW
    setting, 323
RC
    file format, 77
register_command() (*в модуле wlc.main*), 138
REGISTRATION_ALLOW_BACKENDS
    setting, 323
REGISTRATION_CAPTCHA
    setting, 324
REGISTRATION_EMAIL_MATCH
    setting, 324
REGISTRATION_OPEN
    setting, 324
repo
    wlc command line option, 135
REPOSITORY_ALERT_THRESHOLD
    setting, 325
REQUIRE_LOGIN
    setting, 325
reset
    wlc command line option, 135
REST, 87
RESX
    file format, 74
RFC
    RFC 4646, 60
Ruby YAML
    file format, 76
Ruby YAML Ain't Markup Language
    file format, 76

S

save_state() (*метод weblate.addons.base.BaseAddon*), 392
SECRET_KEY
    setting, 183
SENTRY_DSN
    setting, 325
SERVER_EMAIL
    setting, 183
SESSION_COOKIE_AGE_AUTHENTICATED
    setting, 325
SESSION_ENGINE
    setting, 183
setting
    ADMINS, 182
    AKISMET_API_KEY, 301
    ALLOWED_HOSTS, 182
    ANONYMOUS_USER_NAME, 301
    AUDITLOG_EXPIRY, 301
    AUTH_LOCK_ATTEMPTS, 301
    AUTH_TOKEN_VALID, 302
    AUTO_UPDATE, 301
    AUTOFIX_LIST, 302
    AVATAR_URL_PREFIX, 302
    BACKGROUND_TASKS, 303
    BASE_DIR, 303
    BASIC_LANGUAGES, 304
    CHECK_LIST, 304
    COMMENT_CLEANUP_DAYS, 305
    COMMIT_PENDING_HOURS, 305
    CSP_CONNECT_SRC, 304
    CSP_FONT_SRC, 304
    CSP_IMG_SRC, 304
    CSP_SCRIPT_SRC, 304
    CSP_STYLE_SRC, 304
    DATA_DIR, 305

DATABASE_BACKUP, 306
DATABASES, 183
DEBUG, 183
DEFAULT_ACCESS_CONTROL, 306
DEFAULT_ADD_MESSAGE, 307
DEFAULT_ADDON_MESSAGE, 307
DEFAULT_ADDONS, 307
DEFAULT_AUTO_WATCH, 306
DEFAULT_COMMIT_MESSAGE, 307
DEFAULT_COMMITER_EMAIL, 307
DEFAULT_COMMITER_NAME, 308
DEFAULT_DELETE_MESSAGE, 307
DEFAULT_FROM_EMAIL, 183
DEFAULT_LANGUAGE, 308
DEFAULT_MERGE_MESSAGE, 307
DEFAULT_MERGE_STYLE, 308
DEFAULT_PULL_MESSAGE, 309
DEFAULT_RESTRICTED_COMPONENT, 307
DEFAULT_SHARED_TM, 308
DEFAULT_TRANSLATION_PROPAGATION, 308
ENABLE_AVATARS, 309
ENABLE_HOOKS, 309
ENABLE_HTTPS, 309
ENABLE_SHARING, 309
GET_HELP_URL, 310
GITHUB_CREDENTIALS, 310
GITHUB_TOKEN, 311
GITHUB_USERNAME, 311
GITLAB_CREDENTIALS, 310
GITLAB_TOKEN, 310
GITLAB_USERNAME, 310
GOOGLE_ANALYTICS_ID, 311
HIDE_REPO_CREDENTIALS, 311
HIDE_VERSION, 311
IP_BEHIND_REVERSE_PROXY, 312
IP_PROXY_HEADER, 312
IP_PROXY_OFFSET, 312
LEGAL_URL, 313
LICENSE_EXTRA, 313
LICENSE_FILTER, 313
LICENSE_REQUIRED, 314
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 314
LOCALIZE_CDN_PATH, 314
LOCALIZE_CDN_URL, 314
LOGIN_REQUIRED_URLS, 314
LOGIN_REQUIRED_URLS_EXCEPTIONS, 315
MACHINE_TRANSLATION_SERVICES, 316
MATOMO_SITE_ID, 315
MATOMO_URL, 315
MT_APERTIUM_APY, 316
MT_AWS_ACCESS_KEY_ID, 317
MT_AWS_REGION, 317
MT_AWS_SECRET_ACCESS_KEY, 317
MT_BAIDU_ID, 317
MT_BAIDU_SECRET, 317
MT_DEEPL_API_VERSION, 317
MT_DEEPL_KEY, 318
MT_GOOGLE_CREDENTIALS, 318
MT_GOOGLE_KEY, 318
MT_GOOGLE_LOCATION, 318
MT_GOOGLE_PROJECT, 318
MT_MICROSOFT_BASE_URL, 319
MT_MICROSOFT_COGNITIVE_KEY, 319
MT_MICROSOFT_ENDPOINT_URL, 319
MT_MICROSOFT_REGION, 319
MT_MODERNMT_KEY, 319
MT_MODERNMT_URL, 319
MT_MYMEMORY_EMAIL, 320
MT_MYMEMORY_KEY, 320
MT_MYMEMORY_USER, 320
MT_NETEASE_KEY, 320
MT_NETEASE_SECRET, 320
MT_SAP_BASE_URL, 321
MT_SAP_PASSWORD, 322
MT_SAP_SANDBOX_APIKEY, 321
MT_SAP_USE_MT, 322
MT_SAP_USERNAME, 321
MT_SERVICES, 316
MT_TMSERVER, 320
MT_YANDEX_KEY, 321
MT_YOUDAO_ID, 321
MT_YOUDAO_SECRET, 321
NEARBY_MESSAGES, 322
PAGURE_CREDENTIALS, 322
PAGURE_TOKEN, 323
PAGURE_USERNAME, 322
PIWIK_SITE_ID, 315
PIWIK_URL, 315
RATELIMIT_ATTEMPTS, 323
RATELIMIT_LOCKOUT, 323
RATELIMIT_WINDOW, 323
REGISTRATION_ALLOW_BACKENDS, 323
REGISTRATION_CAPTCHA, 324
REGISTRATION_EMAIL_MATCH, 324
REGISTRATION_OPEN, 324
REPOSITORY_ALERT_THRESHOLD, 325
REQUIRE_LOGIN, 325
SECRET_KEY, 183
SENTRY_DSN, 325
SERVER_EMAIL, 183
SESSION_COOKIE_AGE_AUTHENTICATED, 325
SESSION_ENGINE, 183
SIMPLIFY_LANGUAGES, 326
SINGLE_PROJECT, 327
SITE_DOMAIN, 326
SITE_TITLE, 326
SPECIAL_CHARS, 327
STATUS_URL, 327
SUGGESTION_CLEANUP_DAYS, 327
UPDATE_LANGUAGES, 327
URL_PREFIX, 327
VCS_BACKENDS, 328
VCS_CLONE_DEPTH, 328

WEBLATE_ADDONS, 329
WEBLATE_EXPORTERS, 329
WEBLATE_FORMATS, 330
WEBLATE_GPG_IDENTITY, 330
WEBSITE_REQUIRED, 330
setupgroups
weblate admin command, 356
setupgroups command line option
--no-privs-update, 356
--no-projects-update, 356
setuplang
weblate admin command, 356
setuplang command line option
--no-update, 356
show
wlc command line option, 134
SIMPLIFY_LANGUAGES
setting, 326
SINGLE_PROJECT
setting, 327
SITE_DOMAIN
setting, 326
SITE_TITLE
setting, 326
SPECIAL_CHARS
setting, 327
statistics
wlc command line option, 135
STATUS_URL
setting, 327
stay_on_create (*атрибут weblate.addons.base.BaseAddon*), 392
store_post_load() (*метод weblate.addons.base.BaseAddon*), 392
string resources
file format, 69
SUGGESTION_CLEANUP_DAYS
setting, 327

T

TS
file format, 68

U

unit_pre_create() (*метод weblate.addons.base.BaseAddon*), 392
unlock
wlc command line option, 135
unlock_translation
weblate admin command, 356
UPDATE_LANGUAGES
setting, 327
updatechecks
weblate admin command, 356
updategit
weblate admin command, 357
upload
wlc command line option, 135
URL_PREFIX
setting, 327

V

VCS_BACKENDS
setting, 328
VCS_CLONE_DEPTH
setting, 328
version
wlc command line option, 134

W

Weblate (*класс в wlc*), 138
weblate admin command
add_suggestions, 347
auto_translate, 347
celery_queues, 348
checkgit, 348
cleanuptrans, 348
commit_pending, 348
commitgit, 348
createadmin, 349
dump_memory, 349
dumpuserdata, 349
import_demo, 349
import_json, 350
import_memory, 350
import_project, 351
importuserdata, 353
importusers, 353
install_addon, 353
list_languages, 354
list_translators, 354
list_versions, 354
loadpo, 354
lock_translation, 355
move_language, 355
pushgit, 355
setupgroups, 356
setuplang, 356
unlock_translation, 356
updatechecks, 356
updategit, 357
WEBLATE_ADDONS
setting, 329
WEBLATE_ADMIN_EMAIL, 144, 145, 149
WEBLATE_ADMIN_NAME, 144, 145
WEBLATE_ADMIN_PASSWORD, 141, 144, 145
WEBLATE_AKISMET_API_KEY, 364
WEBLATE_ALLOWED_HOSTS, 183, 187, 326
WEBLATE_EMAIL_PORT, 155
WEBLATE_EMAIL_USE_SSL, 155
WEBLATE_EMAIL_USE_TLS, 155
WEBLATE_EXPORTERS
setting, 329
WEBLATE_FORMATS
setting, 330
WEBLATE_GPG_IDENTITY

setting, 330
WEBLATE_LOCALIZE_CDN_PATH, 156
WEBLATE_REQUIRE_LOGIN, 325
WEBLATE_SECURE_PROXY_SSL_HEADER, 146
WEBLATE_SILENCED_SYSTEM_CHECKS, 212
WEBLATE_SITE_DOMAIN, 185, 204, 326
WeblateConfig (*класс в wlc.config*), 138
WeblateException, 137
WEBSITE_REQUIRED
setting, 330
wlc, 132
модуль, 137
wlc command line option
--config PATH, 134
--config-section SECTION, 134
--convert, 135
--format {csv,json,text,html}, 134
--input, 135
--key KEY, 134
--output, 135
--overwrite, 135
--url URL, 134
changes, 135
cleanup, 135
commit, 134
download, 135
list-components, 134
list-languages, 134
list-projects, 134
list-translations, 134
lock, 135
lock-status, 135
ls, 134
pull, 134
push, 135
repo, 135
reset, 135
show, 134
statistics, 135
unlock, 135
upload, 135
version, 134
wlc.config
модуль, 138
wlc.main
модуль, 138

X

XLIFF
file format, 65
XML
file format, 76

Y

YAML
file format, 75
YAML Ain't Markup Language
file format, 75

⁉

двуязычный формат
перевод, 61
модуль
wlc, 137
wlc.config, 138
wlc.main, 138
одноязычный формат
перевод, 61
перевод
iPad, 70
iPhone, 70
двуязычный формат, 61
одноязычный формат, 61
переменная окружения
CELERY_BACKUP_OPTIONS, 157
CELERY_BEAT_OPTIONS, 157
CELERY_MAIN_OPTIONS, 157
CELERY_MEMORY_OPTIONS, 157
CELERY_NOTIFY_OPTIONS, 157
CELERY_TRANSLATE_OPTIONS, 157
POSTGRES_ALTER_ROLE, 154
POSTGRES_DATABASE, 153
POSTGRES_HOST, 153
POSTGRES_PASSWORD, 153
POSTGRES_PORT, 153
POSTGRES_SSL_MODE, 153
POSTGRES_USER, 153
REDIS_DB, 154
REDIS_HOST, 154
REDIS_PASSWORD, 154
REDIS_PORT, 154
REDIS_TLS, 154
REDIS_VERIFY_SSL, 154
ROLLBAR_ENVIRONMENT, 156
ROLLBAR_KEY, 156
SENTRY_DSN, 156
SENTRY_ENVIRONMENT, 156
SOCIAL_AUTH_SLACK_SECRET, 153
UWSGI_WORKERS, 157
WEBLATE_ADD_ADDONS, 157
WEBLATE_ADD_APPS, 157
WEBLATE_ADD_AUTOFIX, 157
WEBLATE_ADD_CHECK, 157
WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 147
WEBLATE_ADMIN_EMAIL, 144, 145, 149
WEBLATE_ADMIN_NAME, 144, 145
WEBLATE_ADMIN_PASSWORD, 141, 144, 145
WEBLATE_AKISMET_API_KEY, 148, 364
WEBLATE_ALLOWED_HOSTS, 145, 183, 187, 326
WEBLATE_AUTH_LDAP_BIND_DN, 150
WEBLATE_AUTH_LDAP_BIND_PASSWORD, 150
WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS, 150
WEBLATE_AUTH_LDAP_SERVER_URI, 150

WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 150
WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 150
WEBLATE_AUTH_LDAP_USER_SEARCH, 150
WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 150
WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 150
WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 150
WEBLATE_BASIC_LANGUAGES, 149
WEBLATE_CSP_CONNECT_SRC, 148
WEBLATE_CSP_FONT_SRC, 148
WEBLATE_CSP_IMG_SRC, 148
WEBLATE_CSP_SCRIPT_SRC, 148
WEBLATE_CSP_STYLE_SRC, 148
WEBLATE_DATABASE_BACKUP, 154
WEBLATE_DEBUG, 144
WEBLATE_DEFAULT_ACCESS_CONTROL, 148
WEBLATE_DEFAULT_AUTO_WATCH, 149
WEBLATE_DEFAULT_COMMITER_EMAIL, 148
WEBLATE_DEFAULT_COMMITER_NAME, 148
WEBLATE_DEFAULT_FROM_EMAIL, 145
WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 148
WEBLATE_DEFAULT_SHARED_TM, 148
WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 148
WEBLATE_EMAIL_BACKEND, 155
WEBLATE_EMAIL_HOST, 155
WEBLATE_EMAIL_HOST_PASSWORD, 155
WEBLATE_EMAIL_HOST_USER, 155
WEBLATE_EMAIL_PORT, 155
WEBLATE_EMAIL_USE_SSL, 155
WEBLATE_EMAIL_USE_TLS, 155
WEBLATE_ENABLE_HTTPS, 146
WEBLATE_GET_HELP_URL, 156
WEBLATE_GITHUB_TOKEN, 147
WEBLATE_GITHUB_USERNAME, 147
WEBLATE_GITLAB_TOKEN, 147
WEBLATE_GITLAB_USERNAME, 147
WEBLATE_GOOGLE_ANALYTICS_ID, 147
WEBLATE_GPG_IDENTITY, 148
WEBLATE_HIDE_VERSION, 149
WEBLATE_IP_PROXY_HEADER, 146
WEBLATE_LEGAL_URL, 156
WEBLATE_LICENSE_FILTER, 148
WEBLATE_LICENSE_REQUIRED, 149
WEBLATE_LOCALIZE_CDN_PATH, 156
WEBLATE_LOCALIZE_CDN_URL, 156
WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 147
WEBLATE_LOGLEVEL, 144
WEBLATE_MT_APERTIUM_APY, 149
WEBLATE_MT_AWS_ACCESS_KEY_ID, 149
WEBLATE_MT_AWS_REGION, 149
WEBLATE_MT_AWS_SECRET_ACCESS_KEY, 149
WEBLATE_MT_DEEPL_API_VERSION, 149
WEBLATE_MT_DEEPL_KEY, 149
WEBLATE_MT_GLOSBE_ENABLED, 150
WEBLATE_MT_GOOGLE_KEY, 149
WEBLATE_MT_MICROSOFT_BASE_URL, 149
WEBLATE_MT_MICROSOFT_COGNITIVE_KEY, 149
WEBLATE_MT_MICROSOFT_ENDPOINT_URL, 149
WEBLATE_MT_MICROSOFT_REGION, 149
WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED, 150
WEBLATE_MT_MODERNMT_KEY, 149
WEBLATE_MT_MYMEMORY_ENABLED, 149
WEBLATE_MT_SAP_BASE_URL, 150
WEBLATE_MT_SAP_PASSWORD, 150
WEBLATE_MT_SAP_SANDBOX_APIKEY, 150
WEBLATE_MT_SAP_USE_MT, 150
WEBLATE_MT_SAP_USERNAME, 150
WEBLATE_NO_EMAIL_AUTH, 153
WEBLATE_PAGURE_TOKEN, 148
WEBLATE_PAGURE_USERNAME, 147
WEBLATE_REGISTRATION_ALLOW_BACKENDS, 145
WEBLATE_REGISTRATION_OPEN, 145
WEBLATE_REMOVE_ADDONS, 157
WEBLATE_REMOVE_APPS, 157
WEBLATE_REMOVE_AUTOFIX, 157
WEBLATE_REMOVE_CHECK, 157
WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 147
WEBLATE_REQUIRE_LOGIN, 147, 325
WEBLATE_SAML_IDP_ENTITY_ID, 153
WEBLATE_SAML_IDP_URL, 153
WEBLATE_SAML_IDP_X509CERT, 153
WEBLATE_SECURE_PROXY_SSL_HEADER, 146
WEBLATE_SERVER_EMAIL, 145
WEBLATE_SILENCED_SYSTEM_CHECKS, 148, 212
WEBLATE_SIMPLIFY_LANGUAGES, 148
WEBLATE_SITE_DOMAIN, 144, 185, 204, 326
WEBLATE_SITE_TITLE, 144
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 152
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 152
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 152
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECR 152
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENA 152
WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 151
WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET,

151
WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 152
WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 152
WEBLATE_SOCIAL_AUTH_FEDORA, 153
WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 151
WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 151
WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, 152
WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 152
WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, 152
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, 152
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, 152
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS, 152
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, 152
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, 152
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM, 152
WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL, 152
WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY, 152
WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY, 152
WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET, 152
WEBLATE_SOCIAL_AUTH_OPENSUSE, 153
WEBLATE_SOCIAL_AUTH_SLACK_KEY, 153
WEBLATE_SOCIAL_AUTH_UBUNTU, 153
WEBLATE_STATUS_URL, 156
WEBLATE_TIME_ZONE, 145
WEBLATE_URL_PREFIX, 148
WEBLATE_WEBSITE_REQUIRED, 149
WL_BRANCH, 297
WL_COMPONENT_NAME, 298
WL_COMPONENT_SLUG, 298
WL_COMPONENT_URL, 298
WL_ENGAGE_URL, 298
WL_FILE_FORMAT, 297
WL_FILEMASK, 297
WL_LANGUAGE, 297
WL_NEW_BASE, 297
WL_PATH, 297
WL_PREVIOUS_HEAD, 297
WL_PROJECT_NAME, 298
WL_PROJECT_SLUG, 298
WL_REPO, 297
WL_TEMPLATE, 297
WL_VCS, 297