Weblate

```
Weblate

1. GNU gettext
2. Android
```

```
1. ../devel/integration
```

```
Weblate settings.py

SINGLE_PROJECT

Weblate preferences:
- Hide completed translations on the dashboard
- Translation editor mode: Full editor
- Zen editor mode: Top to bottom
- Number of nearby strings: 15
- Number of nearby strings to show in each direction in the full editor
- Show secondary translations in the Zen mode
- Hide source if a secondary translation exists

Editor link:
Enter a custom URL to be used as link to the source code. You can use {{branch}}, {{filename}}, and {{line}} as filename and line placeholders.

Special characters:
You can specify additional special visual keyboard characters to be shown while translating. It can be useful for characters you use frequently, but are hard to type on your keyboard.

Default dashboard view:
- Watched translations
- Suggested translations

Save
<table>
<thead>
<tr>
<th>Translation</th>
<th>Translated</th>
<th>Unfinished</th>
<th>Unfinished words</th>
<th>Checks</th>
<th>Suggestions</th>
</tr>
</thead>
<tbody>
<tr>
<td>WeblateOrg/Android — Czech</td>
<td>76%</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/Django — Hungarian</td>
<td>69%</td>
<td>8</td>
<td>109</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/Django — Czech</td>
<td>96%</td>
<td>1</td>
<td>12</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/Django — Hebrew</td>
<td>92%</td>
<td>2</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/Djangojs — Hebrew</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/Djangojs — Hungarian</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/Django — Czech</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/Language names — Hungarian</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/Language names — Hebrew</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/Language names — Czech</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/WeblateOrg — Czech</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/WeblateOrg — Hungarian</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeblateOrg/WeblateOrg — Hebrew</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Special characters

Additional special characters to include in the `{{{}}}`.
Automatically watch projects on contribution
Whenever you translate a string in a project, you will start watching it.

**Watched projects**

**Available:** 
- Marketing

**Choose:** 
- Marketing

You can receive notifications for watched projects and they are shown in the dashboard by default.

Add all projects you want to translate to them as watched projects on the dashboard.

**Save**

**Notification settings**

- **Component with notification**:  
  - Repository failures: Do not notify
  - Repository operation: Do not notify
  - Component locking: Do not notify
  - Changes in comments: Do not notify
  - Changes in contributions: Do not notify
  - Changes in comments: Do not notify
  - Translation notifications: Do not notify

You will receive notifications for every such event in your watched projects.

**Language with notifications**:  
- New language: Do not notify
- New translation component: Do not notify
- New announcement: Do not notify
- News alert: Do not notify

You will only receive these notifications for your translated languages in your watched projects.

**Save**
### Account

**Username**

testuser

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

**Full name**

Webstate Test

**E-mail**

webstate@example.org

You can add another e-mail address below.

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

Save

### Current user identities

<table>
<thead>
<tr>
<th>Identity</th>
<th>User ID</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>testuser</td>
<td>Change password</td>
</tr>
<tr>
<td>E-mail</td>
<td><a href="mailto:webstate@example.org">webstate@example.org</a></td>
<td>Disconnect</td>
</tr>
<tr>
<td>Google</td>
<td><a href="mailto:webstate@example.org">webstate@example.org</a></td>
<td>Disconnect</td>
</tr>
<tr>
<td>GitHub</td>
<td>123456</td>
<td>Disconnect</td>
</tr>
<tr>
<td>Bitbucket</td>
<td>webstate</td>
<td>Disconnect</td>
</tr>
</tbody>
</table>

### 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
ENABLE AVATARS
https://gravatar.com/

API

API

Weblate

Weblate

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

Add new translation component

Powered by Weblate 4.11 | About Weblate | Legal | Contact | Documentation | Donate to Weblate
2.18

Alt+Home
Alt+End
Alt+PageUp or Ctrl+↑ or Alt+↑ or Cmd+↑
Alt+PageDown or Ctrl+↓ or Alt+↓ or Cmd+↓
Alt+Enter or Ctrl+Enter or Cmd+Enter
Ctrl+Shift+Enter or Cmd+Shift+Enter
Ctrl+E or Ctrl+I or Ctrl+M
Cmd+E
Cmd+U
Ctrl+M
Ctrl+I
Ctrl+J
Cmd+S
Ctrl+O
Ctrl+Y
Cmd+Y

User configured Special characters defined in the Weblate RTL SPECIAL_CHARS

3 USER defined.

User configured Special characters defined in the

Weblate SPECIAL_CHARS

SPECIAL_CHARS
Keeping translations same across components
Bulk edit add-on

Tools

Zen

access control
<table>
<thead>
<tr>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>gettext PO</td>
</tr>
<tr>
<td>gettext XLIFF</td>
</tr>
<tr>
<td>XLIFF 1.1</td>
</tr>
<tr>
<td>TermBase eXchange</td>
</tr>
<tr>
<td>Translation Memory eXchange</td>
</tr>
<tr>
<td>gettext MO (gettext PO)</td>
</tr>
<tr>
<td>CSV</td>
</tr>
<tr>
<td>Excel Open XML</td>
</tr>
<tr>
<td>JSON</td>
</tr>
<tr>
<td>Android</td>
</tr>
<tr>
<td>iOS</td>
</tr>
</tbody>
</table>

**GET:**
```
/api/translations/(string:project)/(string:component)/(string:language)/file/
```
The uploaded file will be merged with the current translation. To overwrite already translated strings, don't forget to turn it on.

File

File upload mode
- Add as translation
- Add as suggestion
- Add as translation needing edit
- Replace existing translation file

Processing of strings needing edit
- Do not import

Conflict handling
- Update translated strings

Whether to overwrite existing translations if the string is already translated.

Author name
- Weblate Test

Author e-mail
- weblate@example.org

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

**4.5** Weblate
4.5 .
↓
read-only

4.5 .
↓
forbidden

23
4.5 Terminology

variants

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

**Translation**

**English**
- **Singular**: (%(count)s word)
- **Plural**: (%(count)s words)

**Czech, One**
- několik slov

**Czech, Few**
- několik slov

**Czech, Other**
- (%(count)s slov)

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

**Things to check**
- **Python format**: Following format strings are missing: %counts
- **Missing plurals**: Some plural forms are untranslated

**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**: `/webate/templates/translation/html149`
- **String age**: 9 seconds ago
- **Source string age**: 9 seconds ago
- **Translation file**: `/webate/locale/cs/LC_MESSAGES/ES/django.po`, string 5

**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**
Translation

Singular
%{count} word

Plural
%{count} words

Czech, One
%{count} slovo

Czech, Few
%{count} slova

Czech, Other
%{count} slov

Plural formula: (n=1) ? 0 : (n>=2 & n<=6) ? 1:2

Needs editing

Save and continue  Save and stay  Suggest  SKIP

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
5 seconds ago

Source string age
5 seconds ago

Translation file
weblate/locale/cs/LC_MESSAGE
6/django-po, string 5

Browse all component changes

History
AngularJS

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

AngularJS [https://angular.jp/guide/interpolation]

C

```
There are %d apples
Your balance is %1$d %2$s
```

C format strings [C printf format]

C#

```
There are {0} apples
```

C# [https://docs.microsoft.com/ja-jp/dotnet/api/system.string.format?view=netframework-4.7.2]

ECMAScript

```
There are ${number} apples
```

ECMAScript [https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Template_literals]
There are `{{number}}` apples
There are `$t(number)` apples

---

**i18next**

[See documentation online](https://www.i18next.com/translation-function/interpolation)

**ICU MessageFormat**

[See documentation online](https://unicode.org/reports/tr35/#XML)

**Java**
Java MessageFormat

Java MessageFormat

webate.checks.format.JavaMessageFormatCheck
java-messageformat
auto-java-messageformat
ignore-java-messageformat
There are \(0\) apples

JavaScript

JavaScript

webate.checks.format.JavaScriptFormatCheck
javascript-format
ignore-javascript-format
There are \(d\) apples

Lua

Lua

webate.checks.format.LuaFormatCheck
lua-format
ignore-lua-format
There are \(d\) apples

Object Pascal

Object Pascal

webate.checks.format.ObjectPascalFormatCheck
object-pascal-format
ignore-object-pascal-format
There are \(d\) apples
Perl

Perl

weblate.checks.format.PerlFormatCheck
perl-format
ignore-perl-format
There are %d apples
Your balance is %1$d %2$s

PHP

PHP

weblate.checks.format_PHPFormatCheck
php-format
ignore-php-format
There are %d apples
Your balance is %1$d %2$s

Python

Python

weblate.checks.format.PythonBraceFormatCheck
python-brace-format
ignore-python-brace-format
{amount} {currency}

Python brace Python
Python

Python

weblate.checks.format.PythonFormatCheck
python-format
ignore-python-format
There are %d apples
%(amount)d %(currency)s

Ruby

Ruby

weblate.checks.ruby.RubyFormatCheck
ruby-format
ignore-ruby-format
There are %d apples
%1$s %2$s
Your balance is %+2<amount>f %<currency>s
Your balance is %{amount} %{currency}

Qt

Qt

weblate.checks.qt.QtFormatCheck
qt-format
ignore-qt-format
%1

Qt

Qt

weblate.checks.qt.QtPluralCheck
qt-plural-format
ignore-qt-plural-format
There are %Ln apple(s)

Ruby

Ruby

weblate.checks.ruby.RubyFormatCheck
ruby-format
ignore-ruby-format
There are %d apples
%1$s %2$s
Your balance is %+<amount>f %<currency>s
Your balance is %{amount} %{currency}
Scheme

```
weblate.checks.format.SchemeFormatCheck
scheme-format
ignore-scheme-format
```

Vue I18n

```
weblate.checks.format.VueFormattingCheck
vue-format
ignore-vue-format
```

```
Vue I18n
```

```
weblate.checks.consistency.TranslatedCheck
ignore-translated
```

```
weblate.checks.consistency.ConsistencyCheck
ignore-inconsistent
```

Webate
Keeping translations same across components

**Kashida**

webate.checks.chars.KashidaCheck
ignore-kashida

Kashida Wikipedia [1]

**Markdown**

webate.checks.markup.MarkdownLinkCheck
md-text
ignore-md-link

**Markdown**

webate.checks.markup.MarkdownRefLinkCheck
md-text
ignore-md-reflink

**Markdown**

webate.checks.markup.MarkdownSyntaxCheck
md-text
ignore-md-syntax

**Markdown span**
weblate.checks.chars.MaxLengthCheck
max-length
ignore-max-length
max-length:100
replacements:
xmл-text

weblate.checks.render.MaxSizeCheck
max-size
ignore-max-size
max-size:500:2, font-family:ubuntu, font-size:22
replacements:
xmл-text

weblate.checks.chars.EscapedNewlineCountingCheck
ignore-escaped-newline


\n
weblate.checks.chars.EscapedNewlineCountingCheck
ignore-escaped-newline
\n
weblate.checks.chars.EndColonCheck
ignore-end-colon

Wikipedia

weblate.checks.chars.EndEllipsisCheck
ignore-end-ellipsis

Wikipedia

weblate.checks.chars.EndExclamationCheck
ignore-end-exclamation

Wikipedia

weblate.checks.chars.EndStopCheck
ignore-end-stop

Wikipedia

37
weblate.checks.chars.EndQuestionCheck
ignore-end-question

Wikipedia

weblate.checks.chars.EndSemicolonCheck
ignore-end-semicolon

Wikipedia

weblate.checks.chars.NewLineCountCheck
ignore-newline-count


weblate.checks.consistency.PluralsCheck
ignore-plurals

weblate.checks.placeholders.PlaceholderCheck
placeholders
ignore-placeholders

placeholders:$URL$:$TARGET$:"some long text"
placeholders: r"^[^ ]" 

3.9

weblate.checks.chars.PunctuationSpacingCheck
ignore-punctuation-spacing

Wikipedia: 

weblate.checks.placeholders.RegexCheck
regex
ignore-regex

regex: ^foo|bar$

weblate.checks.consistency.SamePluralsCheck
ignore-same-plurals

weblate.checks.chars.BeginNewlineCheck
ignore-begin-newline

Wikipedia
weblate.checks.chars.BeginSpaceCheck
ignore-begin-space

weblate.checks.chars.EndNewlineCheck
ignore-end-newline

weblate.checks.chars.EndSpaceCheck
ignore-end-space

weblate.checks.same.SameCheck
ignore-same
strict-same

40
HTML

3.9 HTML

weblate.checks.markup.SafeHTMLCheck

safe-html
ignore-safe-html

3.5 URL

weblate.checks.markup.URLCheck

url
ignore-url

2.8 XML

weblate.checks.markup.XMLValidityCheck

ignore-xml-invalid

41
weblate.checks.chars.ZeroWidthSpaceCheck
ignore-zero-width-space

weblate.checks.source.EllipsisCheck
ignore-ellipsis

weblate.checks.icu.ICUSourceCheck
icu-message-format
ignore-icu-message-format

ICU MessageFormat syntax

ICU MessageFormat

weblate.checks.source.LongUntranslatedCheck
ignore-long-untranslated
4.1...

```python
from gettext import ngettext
print(ngettext("Selected %d file", "Selected %d files", files) % files)
```

4.1...

When no field is defined, the lookup happens source, target, and context strings.
Weblate approved, translated, needs-editing, empty, read-only

VCS — plural, context, suggestion, comment, check, dismissed-check, translation, variant, label

this is a quoted string
'another quoted string'
pending, translated, untranslated

AND state:translated AND (source:hello OR source:bar)

translated

2019

= hello world

4.4

change_action:marked-for-edit AND change_time:2018

AND change_action:"Marked for edit"
2 ~ 5\n\texttt{source:r"[2-5]"}
Untranslated strings: state:empty
Unfinished strings: state:translated
Translated strings: state:=translated
Strings marked for edit: state:needs-editing
Strings with suggestions: has:suggestion
Strings with variants: has:variant
Strings with labels: has:label
Strings with context: has:context
Unfinished strings without suggestions: state:translated AND NOT has:suggestion
Strings with comments: has:comment
Strings with any failing checks: has:check
Approved strings: state:approved
Strings waiting for review: state:translated

Explanation
No explanation currently provided.

Labels
No labels currently set.

Flags
python-format

Source string location
webate/template/translation.html149

String age
5 seconds ago

Source string age
6 seconds ago

Translation file
webate/locale/cs/LC_MESSAGE
5/django.po, string 5

You can use Markdown and mention users by @username.
access control

This state is only available when reviews are enabled.

ON
OFF

per-project access control

49
Weblate 2.18

per-project access control

50
Development process

- Developers
  - Intermediate file

Localization process

- Editors
  - Monolingual base language file
- Translators
  - Translation language file
1. Weblate Git
2. Weblate URL
3. Weblate SSH

SSH

API

**Weblate**

1. Weblate Git
2. Weblate URL
3. Weblate SSH

SSH

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
# Merge Weblate changes:
git merge weblate/main
# Resolve conflicts:
```

(52)
edit ...
git add ...
...
git commit

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

# Open Weblate for translation:
wlc unlock

Weblate:

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

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

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

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

gettext PO:

Weblate Git:

# 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 '*.*'`; do
    msgcat --use-first /path/to/weblate/snapshot/$PO
        /path/to/upstream/snapshot/$PO.o $PO.merge
    msgmerge --previous --lang=${PO%.po} $PO.merge domain.pot -o $PO
    rm $PO.merge
    git add $PO
done

git commit

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

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

Keeping translations same across components

```
# Keep translations same across components

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

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

Weblate:

```
# Keep translations same across components

vcs/<project>/<component>/
```

Weblate:

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

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

54
For commits where no authorship is known (for example anonymous suggestions or machine translation results), the authorship is credited to the anonymous user (see `ANONYMOUS_USER_NAME`). You can change the name and e-mail in the management interface.
Weblate

1

Weblate:

pre-commit

updating-target-files

WIP:

Gettext PO

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

POT PO (msgmerge):

updating-target-files

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

WIP:

updating-target-files
# Go to DATA_DIR directory

cd data/vcs

# Compress all Git repositories

```bash
for d in */* ; do
    pushd $d
    git gc
    popd
done
```

**DATA_DIR**

---

**Bad Request (400)**

**ALLOWED_HOSTS**

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

---

**Weblate**

*en*

**Weblate**

ja  ja  ja  fr  fr_FR  de  de

---

**Weblate**

*Git*

**Mercurial**

**VCS**

---

**VCS**

**Weblate**

*Git*

*Mercurial*

Subversion

VCS

Mercurial

VCS

**VCS**

**Weblate**

*Git*

*Mercurial*
Weblate

VCS

: list_translators

Weblate

PO

Weblate

sr@latin

zh@CN

RFC 5646

Weblate

 POSIX

Weblate

translate-toolkit

Translation Related File Formats

GNU gettext

XLIFF

Apple iOS

Android

Weblate
<table>
<thead>
<tr>
<th>Tool</th>
<th>GNU gettext</th>
<th>yes</th>
<th>yes</th>
<th>yes</th>
<th>yes</th>
<th>yes?</th>
<th>yes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>gettext</td>
<td></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes?</td>
<td>yes?</td>
</tr>
<tr>
<td>XLIFF</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes?</td>
<td>yes?</td>
</tr>
<tr>
<td>Java</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>mi18n</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>GWT</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Joomla</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Qt Linguist .ts</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes?</td>
<td>yes?</td>
</tr>
<tr>
<td>Android</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes?</td>
<td>yes?</td>
</tr>
<tr>
<td>Apple iOS</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>PHP</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>JSON</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>JSON i18next</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>go-i18n</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ARB</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>WebExtension</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>JSON</td>
<td></td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>XML</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes?</td>
<td>yes?</td>
</tr>
<tr>
<td>CSV</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes?</td>
</tr>
<tr>
<td>YAML</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Ruby YAML</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DTD</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>XML</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes?</td>
<td>yes?</td>
</tr>
<tr>
<td>Windows RC</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Excel OpenDocument</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes?</td>
</tr>
<tr>
<td>HTML</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>OpenDocument</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>IDML</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>INI</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Inno Setup INI</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>TermBase eXchange</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes?</td>
<td>yes?</td>
</tr>
<tr>
<td>Stringsdict</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Fluent</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>no?</td>
<td>no?</td>
<td>no?</td>
<td>no?</td>
<td>no?</td>
<td>no?</td>
<td>no?</td>
<td>no?</td>
</tr>
</tbody>
</table>
Weblate 3.10

GNU gettext

libre

gnu-gettext gettext PO:

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

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

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

Weblate

gettext PO:

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

XML

```
<http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html#maxwidth>`_
```

Laravel

```bash
devel/gettext|devel/sphinx|Gettext Wikipedia |.po Files|configure ALL_LINGUAS gettext
```

gettext PO:

```python
#: 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"

#: 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"

---

**XLIFF**

XML Localization Interchange File Format (XLIFF) is a standard for localizing software. Weblate supports XLIFF as a format for localization.

XML

```
<target>
  approved="yes"
</target>
```

Weblate XLIFF supports the following state:

- approved
- needs-translation
- needs-adaptation
- needs-l10n

61
<?xml version="1.0" encoding="UTF-8"?>
<trans-unit id="10" approved="yes">
    <source xml:space="preserve">hello</source>
    <target xml:space="preserve">Hello, world!</target>
</trans-unit>

<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>

XLIFF Wikipedia: font attribute in XLIFF 1.2 maxwidth attribute in XLIFF 1.2
Java

Weblate ISO-8859-1 UTF-8 UTF-16 Unicode

UTF-8

src/app/Bundle_*.properties
src/app/Bundle.properties
Empty

Java ISO-8859-1

Mozilla and Java properties files

mi18n

GWT

src/app/Bundle_*.properties
src/app/Bundle.properties
Empty

GWT

Mozilla and Java properties files
INI Files

Java

INI Files

Joomla

INI Files

Weblate

INI Files

Joomla

INI Files

Weblate

INI Files

Inno Setup INI

INI Files

Unicode

INI Files

Joomla
Qt Linguist .ts

Qt Linguist manual

Android

Android string resources

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

```xml
<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>
```

strings.xml: https://gist.github.com/paour/11291062
Apple iOS

iOS, iPhone/iPad

Weblate

Resources/*.lproj/Localizable.strings
Resources/en.lproj/Localizable.strings
Resources/Base.lproj/Localizable.strings
Empty
iOS (UTF-8)

Stringsdict
Apple "strings files" documentation
Mac OS X strings

PHP

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

Laravel PHP

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

JSON

2.0

2.16

Laravel 4.1

Web: Javascript I18n

vue-i18n
react-intl

i18next

go-i18n

66
"Hello, world!": "Ahoj světe! ",
"Orangutan has %d banana.": "",
"Try Weblate at https://demo.weblate.org/!": "",
"Thank you for using Weblate.": ""
}

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

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

"hello": "Hello",
"apple": "I have an apple",
"apple_plural": "I have {{count}} apples",
"apple_negative": "I have no apples"
Weblate
langs/*.json
langs/en.json
Empty
i18next JSON

JSON
i18next JSON Format

go-i18n JSON

ARB

WebExtension JSON

Mozilla Firefox Google Chromium

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

Weblate

_locales/*/messages.json
_locales/en/messages.json
Empty
WebExtension JSON

.NET Resource files (.resx)

.location
.source
.target
.ID
.fuzzy
.context
.translator_comments
developer_comments

translate-toolkit

.location
.source
.target
.ID
.fuzzy
.context
.translator_comments
developer_comments

.NET Resource files (.resx) updating-target-files ref: addon-weblate.cleanup.generic
Thank you for using Weblate.,Děkujeme za použití Weblate.

**CSV**

```
locale/*.csv
locale/en.csv
```

**YAML**

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

**Ruby YAML**

```ruby
weblate:
  hello: ""
  orangutan: ""
  try: ""
  thanks: ""
```
DTD 2.18

DTD:

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

XML 3.9

XML:

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

Flat XML

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT
STRINGTABLE
BEGIN
  IDS_MSG1 "Hello, world!"
  IDS_MSG2 "Orangutan has %d banana."
  IDS_MSG3 "Try Weblate at http://demo.weblate.org/!"
  IDS_MSG4 "Thank you for using Weblate."
END
```
**Windows RC files**

**Triple-T gradle-play-publisher**

**Fastlane**

**F-Droid**

**Weblate**

**Subtitles**
Excel Open XML

3.2 XLSX.
Excel Open XML.xlsx source target context XLSX

HTML

4.1 HTML.

4.6 HTML.

Simple Text Documents

OpenDocument

4.1 HTML.

OpenDocument Format

IDML

4.1 HTML.

Adobe InDesign
**TermBase eXchange**

**TBX Wikipedia**

**Stringsdict**

**Stringsdict**

**Apple iOS**

**Fluent**

**Fluent**
Translation Related File Formats

Weblate: 

Git, GitHub, Gerrit, Subversion, Mercurial

VCS Weblate URL: https://github.com/WeblateOrg/weblate.git

Hosted Weblate

GitHub, Bitbucket, Codeberg, GitLab, hosted@weblate.org

SSH:

git@github.com:WeblateOrg/weblate.git
Weblate SSH

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

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAAGhXyAHkdRZJcsS+N7WGPasLkm7y1iw6VxHWkWdQV2xkCF1sJFqot8Xs
5LcT57rDOO/22uH/VnMfJpCk7x3D4eIevm6kNo5Y1v0hZK3wua

Download private key

Known host keys

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Keytype</th>
<th>Fingerprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>github.com</td>
<td>ssh-ed25519</td>
<td>+DH3w0w6W+U1h6zV2s5j/zL80zP8sM+h4kn4kSGkU</td>
</tr>
<tr>
<td>github.com</td>
<td>ed25519-openssh</td>
<td>025770353c58613b784e0b7032b024f3f505f7ed</td>
</tr>
<tr>
<td>github.com</td>
<td>ssh-rsa</td>
<td>nThnq6KkKpJGzYZGtKj103k9OCspRmTxsEARuKh4EB5Y8</td>
</tr>
</tbody>
</table>

Add host key

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

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Submit
SSH :

**GitHub**

Another way of accessing a remote repository is via SSH. To use SSH with GitHub, you need to connect using an SSH key.

To access GitHub with SSH, you need to:

1. Download your SSH private key.
2. Add your SSH key to your GitHub account:
   
   - Click on the SSH button.
   - Upload your private key.
   - Add the key to your account.

Once your key is added to your account, you can push your changes to the remote repository using SSH:

```
$ ssh add
```

**Weblate**

Another way of accessing a remote repository is via SSH. To use SSH with Weblate, you need to connect using an SSH key.

To access Weblate with SSH, you need to:

1. Download your SSH private key.
2. Add your SSH key to your Weblate account:
   
   - Click on the SSH button.
   - Upload your private key.
   - Add the key to your account.

Once your key is added to your account, you can push your changes to the remote repository using SSH:

```
$ ssh add
```
Weblate URL

weblate :// project/component

VCS

HTTPS

GitHub URL: https://user:your_access_token@github.com/WeblateOrg/weblate.git

HTTP/HTTPS

cURL documentation

http_proxy
https_proxy
all_proxy

Git config documentation

The cURL manpage

Git

Git 2.12
Git

Weblate VCS
HOME=$DATA_DIR/home
DATA_DIR/.git

remote helpers
Bazaar & Mercurial & GitHub & GitHub:

Git-remote-hg & git-remote-bzr

Bazaar & Launchpad & gnuhhello
bzr::lp:gnuhhello

Mercurial selenic.com & hello
hg::http://selenic.com/repo/hello

GitHub

Pushing changes from Weblate

GitLab

Pushing changes from Weblate
Gitea pull requests

This just adds a thin layer atop Git using the Gitea API to allow pushing translation changes as pull requests instead of pushing directly to the repository. There is no need to use this to access Git repositories, ordinary Git works the same, the only difference is how pushing to a repository is handled. With Git changes are pushed directly to the repository, while Gitea pull requests creates pull requests.

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

Pushing changes from Weblate, GITEA_USER, GITEA_TOKEN, GITEA_CREDENTIALS

Pagure

Git

Gitea pull requests

This just adds a thin layer atop Git using the Gitea API to allow pushing translation changes as pull requests instead of pushing directly to the repository. There is no need to use this to access Git repositories, ordinary Git works the same, the only difference is how pushing to a repository is handled. With Git changes are pushed directly to the repository, while Gitea pull requests creates pull requests.

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

Pushing changes from Weblate, GITEA_USER, GITEA_TOKEN, GITEA_CREDENTIALS

Pagure

Git

Gerrit

Git

Mercurial

Git

Subversion

Git

Perl
Subversion

Weblate

Weblate

DATA_DIR

'svn'

'SSHOME' DATA_DIR

# Use DATA_DIR as configured in Weblate settings.py, it is /app/data in

HOME=${DATA_DIR}/home

svn co https://svn.example.com/example

Git

Git

3.8

Weblate

VCS

Weblate

VCS

Weblate

Weblate

REST API

2.6

REST API

Weblate 2.6

API

/api/ URL

Django REST framework

Weblate

API

100

API

ANY

API

format -- Accept -- Accept json

api REST API URL

page -- next previous

Accept -- Authorization: Token YOUR-TOKEN

Content-Type -- Accept

Allow -- HTTP detail

json

HTTP

config

HTTP

ᴴ²

HTTP
200 OK
201 Created
204 No Content
400 Bad Request
403 Forbidden
429 Too Many Requests

CURL:

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

HTTP/1.0 200 OK
Date: Fri, 25 Mar 2016 09:46:12 GMT
Server: WSGI Server/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/"
}

CURL:


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

JSON:

82
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"}

CURL

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

CURL JSON

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

API 400

API settings.py Throttling in Django REST framework documentation

Docker WEBLATE_API_RATELIMIT_ANON WEBLATE_API_RATELIMIT_USER

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
Allow: GET, HEAD, OPTIONS

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

**GET /api/users/**

```json
POST /api/users/
```

`username` (string) -- 

`full_name` (string) --

`email` (string) --

`is_superuser` (boolean) --

`is_active` (boolean) --

`is_bot` (boolean) -- Is user bot? (optional) (used for project scoped tokens)

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

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

`username` (string) -- 

`full_name` (string) --

`email` (string) --

`is_superuser` (boolean) --

`is_active` (boolean) --

`is_bot` (boolean) --

`date_joined` (string) --

`groups` (array) --

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

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

username (string) -- JSON

full_name (string) --
email (string) --
is_superuser (boolean) --
is_active (boolean) --
is_bot (boolean) --
dateJoined (string) --

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

username (string) -- JSON

full_name (string) --
email (string) --
is_superuser (boolean) --
is_active (boolean) --
is_bot (boolean) --
dateJoined (string) --

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

username (string) --

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

username (string) --

string group_id --

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

username (string) -- JSON

translated (int) --
suggested (int) --
uploaded (int) --
commented (int) --
languages (int) --

85
GET /api/users/(str: username)/notifications/

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

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

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

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

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

POST /api/groups/

name (string) --
project_selection (int) --
language_selection (int) --
defining_project (str) --

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

id (int) -- ID

name (string) --
project_selection (int) --
language_selection (int) --
roles (array) --
projects (array) --
components (array) --
componentlists (array) --
defining_project (str) --

JSON Value:

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

id(int) -- ID

name(string) -- JSON name

project_selection(int) -- ID

language_selection(int) -- ID

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

id(int) -- ID

name(string) -- JSON name

project_selection(int) -- ID

language_selection(int) -- ID

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

id(int) -- ID

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

id(int) -- ID

string role_id -- ID

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

id(int) -- ID

string component_id -- ID

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

int: component_id

id(int) -- ID

component_id(int) -- ID

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

id(int) -- ID

string project_id -- ID

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

int: project_id

id(int) -- ID

project_id(int) -- ID
POST /api/groups/ (int: id)/languages/  

id (int) -- [ID] 

string language_code -- [ID] 

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

string: language_code 

id (int) -- [ID] 

language_code (string) -- [ID] 

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

id (int) -- [ID] 

string component_list_id -- [ID] 

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

int: component_list_id 

id (int) -- [ID] 

component_list_id (int) -- [ID] 

GET /api/roles/  

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

POST /api/roles/  

name (string) -- [ID] 

permissions (array) -- [ID] 

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

id (int) -- [ID] 

JSON: 

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

id(int)--ID
name(string)--
permissions(array)--

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

id(int)--ID
name(string)--
permissions(array)--

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

id(int)--ID

GET /api/languages/

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

POST /api/languages/

code(string)--
name(string)--
direction(string)--
plural(object)--

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

language(string)--
code(string)--
direction(string)--
plural(object)--
aliases(array)--

JSON response:

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

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

language(string)-- HTTP
name(string)-- JSON
direction(string)--
plural(object)--
PATCH /api/languages/(string: language)/

language(string)--
name(string)--
direction(string)--
plural(object)--
DELETE /api/languages/(string: language)/

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

language(string)--
total(int)--
total_words(int)--
last_change(timestamp)--
recent_changes(int)--
translated(int)--
translated_percent(float)--
translated_words(int)--
translated_words_percent(int)--
translated_chars(int)--
translated_chars_percent(int)--
total_chars(int)--
fuzzy(int)--
fuzzy_percent(int)--
failing(int)--
failing--

91
GET /api/projects/
    "GET /api/projects/(string:project)/"

POST /api/projects/
    3.9
    "name (string) -- "
    "slug (string) -- "
    "web (string) -- Web URL"

GET /api/projects/(string:project)/
    "project (string) -- URL"
    "name (string) -- "
    "slug (string) -- "
    "web (string) -- Web URL"
    "components_list_url (string) -- URL: GET /api/projects/ (string:project)/components/
    "repository_url (string) -- URL: GET /api/projects/ (string:project)/repository/
    "changes_list_url (string) -- URL: GET /api/projects/(string:project)/changes/
    "translation_review (boolean) -- "
    "source_review (boolean) -- "
    "set_language_team (boolean) -- "Language-Team"
    "enable_hooks (boolean) -- "
    "instructions (string) -- "
    "language_aliases (string) -- "

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)/
    4.3

PATCH
    "project (string) -- URL URL"
    "component (string) -- URL URL"
PUT /api/projects/(string: project) /

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

project (string) -- URL

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

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

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

GET /api/changes/ (int:id) /

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

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

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

VCS

needs_commit (boolean) -- JSON

needs_merge (boolean) -- JSON

needs_push (boolean) -- JSON

operation (string) -- push, pull, commit, reset, cleanup, file-sync

result (boolean) --

CURL:

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

JSON:

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

{"operation":"pull"}
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/

project (string)-- URL
results (array)-- GET /api/components/(string:project)/

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

3.9
4.3 zipfile and docfile VCS: Weblate URL

4.6 Weblate URL disable_autoshare

1 VCS Weblate URL

project (string)-- URL
file zipfile -- Weblate ZIP file
docfile --

boolean disable_autoshare-- Weblate URL

object -- GET /api/components/(string:project)/

result (object) GET /api/components/(string:project)/

zipfile and docfile JSON multipart/form-data

CURL

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

CURL JSON

94
JSON request to create a new component from Git:

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

JSON request to create a new component from another one:

```bash
curl \n   --data-binary '{
     "file_format": "po",
     "filemask": "po/*.po",
     "name": "Weblate",
     "slug": "weblate",
     "repo": "weblate://weblate/hello",
     "template": "",
     "new_base": "po/hello.pot",
     "vcs": "git"
   }' \n   -H "Content-Type: application/json" \n   -H "Authorization: Token TOKEN" \n   http://example.com/api/projects/hello/components/
```
GET /api/projects/(string: project)/languages/

project (string)-- [URL](http://example.com/api/projects/hello/)

results (array)--

language (string)--

code (string)--

total (int)--

translated (int)--

translated_percent (float)--

translated_words (int)--

words_percent (float)--

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

project (string)-- [URL](http://example.com/api/projects/hello/)

total (int)--

translated (int)--

translated_percent (float)--

translated_words (int)--

translated_words (int)--
words_percent (float) -- [156x185]  

## GET  
/api/components/  

### GET /api/components/(string:project)/  

project (string) -- [120x156] URL  

component (string) -- [120x156] URL  

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

project (object) -- [120x156] URL  

name (string) --  

slug (string) --  

comp (string) --  

repo (string) --  

git_export (string) -- [192x198] URL  

branch (string) --  

push_branch (string) -- [144x192] push  

filemask (string) --  

template (string) --  

edit_template (string) --  

intermediate (string) --  

clean (string) --  

git_clean (string) --  

new_base (string) --  

file_format (string) --  

template (string) --  

language_code_style (string) --  

source_language (object) -- [242x242] URL  

check_flags (string) --  

priority (string) --  

enforced_checks (string) --  

restricted (string) --  

repoweb (string) --  

report_source_bugs (string) --  

merge_style (string) --  

commit_message (string) --  

add_message (string) --  

delete_message (string) --  

97
merge_message (string) -- `merge_message`
addon_message (string) -- `addon_message`
allow_translation_propagation (string) -- `allow_translation_propagation`
enable_suggestions (string) -- `enable_suggestions`
suggestion_voting (string) -- `suggestion_voting`
suggestion_autoaccept (string) -- `suggestion_autoaccept`
push_on_commit (string) -- `push_on_commit`
commit_pending_age (string) -- `commit_pending_age`
auto_lock_error (string) -- `auto_lock_error`
language_regex (string) -- `language_regex`
variant_regex (string) -- `variant_regex`
repository_url (string) -- URL: GET /api/components/(string:project)/(string:component)/repository/
translations_url (string) -- URL: GET /api/components/(string:project)/(string:component)/translations/
lock_url (string) -- URL: GET /api/components/(string:project)/(string:component)/lock/
changes_list_url (string) -- URL: GET /api/components/(string:project)/(string:component)/changes/
task_url (string) -- URL: GET /api/tasks/(str:uuid)/

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

{
    "name": "new name"
}

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/"
    },

CURL:

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

JSON:

PATCH (string) -- URL
component (string) -- URL
source_language (string) -- JSON
name (string) --
slug (string) --
repo (string) -- VCS URL

CURL:

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

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

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

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

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

GET /api/changes/(int:id)/

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

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

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

GET /api/changes/

GET /api/components/

PUT /api/components/

"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/"
```
project (string) -- URL
component (string) -- URL

GET /api/screenshots/(int:id)/
GET /api/components/(string: project)/
string: component/lock/

JSON:
{
  "locked": false
}

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

JSON:

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

JSON:

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

{"lock": true}

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/

VCS

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

101
```
project (string) -- URL

component (string) -- URL

needs_commit (boolean)
needs_merge (boolean)
needs_push (boolean)
remote_commit (string)
status (string) -- VCS

merge_failure -- null

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

CURL:

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

JSON:

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

{"operation":"pull"}

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/

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

102
**GET /api/components/(string: project)/**

**string: component/translations/**

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

**string: component/translations/**

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

**JSON results (array)** --

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

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

string: component/translations/

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

JSON result (object) --**

**CURL** --

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

**JSON REQUEST:**

```
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"}
```

**JSON RESPONSE:**

```
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": {```
GET /api/components/(string: project)/
string: component/statistics/

JSON

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

result: 4.5

project (string) -- URL
component (string) -- URL
JSON

results (array) -- GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/

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

result: 4.5

project (string) -- URL
component (string) -- URL
JSON

projects (array) -- GET /api/projects/(string:project)/

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

result: 4.5

project (string) -- URL
component (string) -- URL

string project_slug --

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

result: 4.5

project (string) -- URL
component (string) -- URL

project_slug (string) --
GET /api/translations/

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

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

project (string) -- URL
component (string) -- URL
language (string) -- JSON

component (object) -- GET /api/components/(string:project)/(string:component)/

failing_checks (int) -- 
failing_checks_percent (float) -- 
failing_checks_words (int) -- 
filename (string) -- 
fuzzy (int) -- 
fuzzy_percent (float) -- 
fuzzy_words (int) -- 
have_comment (int) -- 
have_suggestion (int) -- 
is_template (boolean) -- 
language (object) -- GET /api/languages/(string:language)/

language_code (string) -- 
last_author (string) -- 
last_change (timestamp) -- 
revision (string) -- 
share_url (string) -- URL

total (int) -- 
total_words (int) -- 
translate_url (string) -- URL

translated (int) -- 
translated_percent (float) -- 
translated_words (int) -- 

repository_url (string) -- URL: GET /api/translations/(string:project)/(string:component)/(string:language)/repository/

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

changes_list_url (string) -- URL: GET /api/translations/(string:project)/(string:component)/(string:language)/changes/

units_list_url (string) -- URL: GET /api/translations/(string:project)/(string:component)/(string:language)/units/

JSON
DELETE /api/translations/(string: project)/
string: component/string: language/

3.9.0

106
GET /api/translations/(string: project)/string: component/string: language/changes/

GET /api/changes/

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

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

VCS format: ref: download 107
format -- po, mo, xfl, xflf11, tbx, css, xlsx, json, aresource, strings

project (string) -- URL
component (string) -- URL
language (string) --

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

project (string) -- URL
component (string) -- URL
language (string) --

string conflict -- ignore, replace-translated, replace-approved
file file --
string email --
string author --
string method -- translate, approve, suggest, fuzzy, replace, source, add

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/

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

project (string) -- URL
component (string) -- URL
language (string) --

POST /api/translations/(string: project)/
string: component/string: language/repository/

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

project (string) --
component (string) --
language (string) --
operation (string) -- push, pull, commit, reset, cleanup

result (boolean) --

GET /api/translations/(string: project)/
string: component/string: language/statistics/
2.7 URL

project (string) -- URL
component (string) -- URL
language (string) -- URL
code (string) -- URL
failing (int) -- URL
failing_percent (float) -- URL
fuzzy (int) -- URL
fuzzy_percent (float) -- URL
total_words (int) -- URL
last_author (string) -- URL
last_change (timestamp) -- URL
name (string) -- URL
total (int) -- URL
translated (int) -- URL
translated_percent (float) -- URL
url (string) -- URL
url_translate (string) -- URL

---
unit

2.10 URL

GET /api/units/

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

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

id (int) -- ID

translation (string) -- URL

source (array) -- URL

previous_source (string) -- URL

target (array) -- URL

id_hash (string) -- URL

target (string) -- URL

context (string) -- URL

note (string) -- URL

flags (string) -- URL
state (int) -- 0 - 10 - 20 - 30 - 100
fuzzy (boolean) -- "fuzzy"
translated (boolean) --
approved (boolean) --
position (int) --
has_suggestion (boolean) --
has_comment (boolean) --
has_failing_check (boolean) --
um_words (int) --
priority (int) -- 100
id (int) --
explanation (string) -- Additional info on source strings
extra_flags (string) --
web_url (string) -- URL
source_unit (string) -- GET /api/units/(int:id)/
pending (boolean) -- whether the unit is pending for write
timestamp (timestamp) -- string age
PATCH /api/units/(int: id) /

id (int) -- ID

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

DELETE /api/units/(int: id) /
### 2.10 GET

**GET /api/changes/**

- **GET /api/changes/(int:id)/**
- **GET /api/screenshots/**
- **GET /api/screenshots/(int:id)/**

### 2.14 GET

**GET /api/screenshots/**

- **GET /api/screenshots/(int:id)/**
- **GET /api/units/(int:id)/file/**

---

111
POST /api/screenshots/(int: id)/file/

id(int)-- ID

file image--

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/

id(int)-- ID

string unit_id-- ID

name(string)--

translation(string)--

file_url(string)-- GET /api/screenshots/(int:id)/file/

units(array)-- GET /api/units/(int:id)/

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

id(int)-- ID

unit_id--

POST /api/screenshots/

file image--

string name--

string project_slug--

string component_slug--

string language_code--

name(string)--

component(string)--

file_url(string)-- GET /api/screenshots/(int:id)/file/

units(array)-- GET /api/units/(int:id)/

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

id(int)-- ID

name(string)--

component(string)--

file_url(string)-- GET /api/screenshots/(int:id)/file/

units(array)-- GET /api/units/(int:id)/
PUT /api/screenshots/(int:id)/

id(int) -- ID
name(string) -- URL
component(string) -- URL
file_url(string) -- URL
units(array) -- URL

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

id(int) -- ID

4.4.1 API
GET /api/addons/

GET /api/addons/(int:id)/

id(int) -- ID
name(string) -- URL
component(string) -- URL
configuration(object) --

POST /api/components/(string:project)/
string: component/addons/

project_slug(string) --
component_slug(string) --
name(string) --
configuration(object) --

PATCH /api/addons/(int:id)/

id(int) -- ID
configuration(object) --

PUT /api/addons/(int:id)/

id(int) -- ID
configuration(object) --
DELETE /api/addons/(int: id)/

id (int) -- ID

GET /api/component-lists/

slug (string) -- JSON
name (string) -- JSON
show_dashboard (boolean) --
components (array)

GET /api/component-lists/(str: slug)/

slug (string) -- JSON
name (string) -- JSON
show_dashboard (boolean) --
components (array)

PUT /api/component-lists/(str: slug)/

slug (string) -- JSON
name (string) -- JSON
show_dashboard (boolean) --
components (array)

PATCH /api/component-lists/(str: slug)/

slug (string) -- JSON
name (string) -- JSON
show_dashboard (boolean) --
components (array)

DELETE /api/component-lists/(str: slug)/

slug (string) --

POST /api/component-lists/(str: slug)/components/

slug (string) --
string component_id -- ID

DELETE /api/component-lists/(str: slug)/components/

str: component_slug
```

slug (string) --
component_slug (string) --

---

4.5 API
---

GET /api/tasks/
---

GET /api/tasks/ (str: uuid)/
---

uuid (string) -- UUID
JSON

completed (boolean) --
progress (int) --
result (object) --
log (string) --

---

GET /api/metrics/
---

units (int) --
units_translated (int) --
users (int) --
changes (int) --
projects (int) --
components "(int) --
translations" (int) --
languages "(int) --
checks" (int) --
configuration_errors" (int) --
suggestions" (int) --
celery_queues (object) -- Celery
name (string) --
```
POST /api/projects/(string:project)/repository/
GET /hooks/update/(string: project)/string: component/

2.6 POST /api/components/(string:project)/(string:component)/repository/ACL

2.6 POST /api/projects/(string:project)/repository/ACL

POST /hooks/github/
GitHub Automatically receiving changes from GitHub
GitHub https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks
GitHub Webhooks ENABLE_HOOKS

Webhook URL

POST /hooks/gitlab/
GitLab Automatically receiving changes from GitLab
GitLab https://docs.gitlab.com/ee/user/project/integrations/webhooks.html
GitLab Webhooks ENABLE_HOOKS

Webhook URL

POST /hooks/bitbucket/
Bitbucket Automatically receiving changes from Bitbucket
Bitbucket https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/
Bitbucket ENABLE_HOOKS

Webhook URL

POST /hooks/pagure/
Pagure Pagure
Pagure https://docs.pagure.org/pagure/usage/using_webhooks.html
Pagure WEB ENABLE_HOOKS

Webhook URL

POST /hooks/azure/
Azure Repos Azure Repos
Automatically receiving changes from Azure Repos
Azure Repos
Azure Repos Web Hooks
ENABLE_HOOKS
Weblate
POST /hooks/gitea/
Gitea Webhook
Automatically receiving changes from Gitea Repos
Gitea
https://docs.gitea.io/en-us/webhooks/
Generic information about Gitea Webhooks
ENABLE_HOOKS
Weblate
POST /hooks/gitee/
Gitee Webhook
Automatically receiving changes from Gitee Repos
Gitee
https://gitee.com/help/categories/40
Gitee
ENABLE_HOOKS
Weblate
GET /exports/stats/ (string: project) /
string: component/
format (string) -- json | csv
HTTP 2.6
GET /api/components/(string:project)/(string:component)/statistics/
GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/
ACL

Weblate
GET /exports/stats/weblate/main/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript

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", 117
Weblate 2.7

Weblate 2.7 comes with a new Weblate 2.7 wlc API.

**Weblate Python**

```bash
pip3 install wlc
```

**Docker**

You can pull the Weblate Docker image from the Docker Hub:

```bash
docker pull weblate/wlc
```

To run the Weblate Docker container with the API URL and API KEY from your local Weblate instance, you can use:

```bash
docker run --rm weblate/wlc [WLC_ARGS]
```

You might want to pass your current directory as a volume to the Docker container, the easiest approach is to add your current directory as `PWD`:

```bash
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

**wlc**

The `wlc` command can be used to interact with the Weblate API. For other locations, you can set the `url` and `keys` sections in the `~/.config/weblate` file:

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

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

```bash
wlc ls
wlc commit sandbox/hello-world
```
wlc [arguments] <command> [options]

Weblate Python Weblate REST API Weblate Weblate REST API

--format {csv, json, text, html}

--url URL
API URL for REST API URL: https://hosted.weblate.org/api/

--key KEY
API KEY: [your key]

--config PATH
API config path: [your config path]

--config-section SECTION
API config section: [your config section]

version
list-languages
list-projects
list-components
list-translations
show
ls
commit
pull
push

120
reset
Weblate cleanup
Weblate repo statistics
Weblate lock-status
lock
unlock
changes
download
--convert
--output
upload
--overwrite
--input
--method
--fuzzy
--author-name
--author-email

```
$ wlc ls --help
```

121
Windows

```
[weblate]
--config-section: weblate
key
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

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

```
VCS .weblate wlc
```

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

```
$ 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/
```

```
$ wlc upload project/component/language --input /tmp/hello.po
```

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

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

$ wlc commit

**Weblate** Python API

```
Python API Weblate API wlc API

pip install wlc

wlc

WeblateException

exception wlc.WeblateException

Weblate

class wlc.Weblate (key='', url=None, config=None)

key (str) -- key
url (str) -- API URL
config (wlc.config.WeblateConfig) -- API URL

get (path)

path (str) -- API GET

object

post (path, **kwargs)

path (str) -- API POST

object
```

123
class wlc.config.WeblateConfig(section='wlc')

section(str) --
XDG

load(path=None)

path(str) --

XDG /etc/xdg/wlc

~/.config/wlc

class wlc.main.Command(args, config, stdout=None)

settings(list) --
stdout(object) --

sys.stdout

args(list) --

sys.args

@wlc.main.register_command(command)

main()
1. Weblate-docker:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. `docker-compose.override.yml`:

```yaml
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
```

3. Weblate:

```
docker-compose up
```

Enjoy your Weblate deployment, it's accessible on port 80 of the weblate container.

#### 2.15-2
- The setup has changed recently, prior there was separate web server container, since 2.15-2 the web server is embedded in the Weblate container.

#### 3.7.1-6
- In July 2019 (starting with the 3.7.1-6 tag), the containers are not running as a root user. This has changed the exposed port from 80 to 8080.

---

**Invoking management commands**

### Choosing Docker hub tag

You can use following tags on Docker hub, see [https://hub.docker.com/r/weblate/weblate/tags/](https://hub.docker.com/r/weblate/weblate/tags/) for full list of available ones.

- **latest**: Weblate stable release, matches latest tagged release
  - Use case: Rolling updates in a production environment
- `<VERSION>-<PATCH>`: Weblate stable release with development changes in the Docker container (for example updated dependencies)
  - Use case: Well defined deploy in a production environment
- **edge**: Weblate stable release with development changes in the Docker container (for example updated dependencies)
  - Use case: Rolling updates to test upcoming Weblate features
- **bleeding**: Development version Weblate from Git
  - Use case: Well defined deploy to test upcoming Weblate features

Every image is tested by our CI before it gets published, so even the bleeding version should be quite safe to use.
Docker container with HTTPS support

Please see [link] for generic deployment instructions, this section only mentions differences compared to it.

Using own SSL certificates

**New in 3.8-3**

In case you have own SSL certificate you want to use, simply place the files into the Weblate data volume (see Docker container volumes):

- `ssl/fullchain.pem` containing the certificate including any needed CA certificates
- `ssl/privkey.pem` containing the private key

Both of these files must be owned by the same user as the one starting the docker container and have file mask set to 600 (readable and writable only by the owning user).

Additionally, Weblate container will now accept SSL connections on port 4443, you will want to include the port forwarding for HTTPS in docker compose override:

```yaml
version: '3'
services:
  weblate:
    ports:
    - 80:8080
    - 443:4443
```

If you already host other sites on the same server, it is likely ports 80 and 443 are used by a reverse proxy, such as NGINX. To pass the HTTPS connection from NGINX to the docker container, you can use the following configuration:

```nginx
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>;
  }
}
```

Replace `<SITE_URL>`, `<SITE>` and `<EXPOSED_DOCKER_PORT>` with actual values from your environment.

Automatic SSL certificates using Let's Encrypt

In case you want to use Let's Encrypt automatically generated SSL certificates on public installation, you need to add a reverse HTTPS proxy an additional Docker container, `https-portal` will be used for that. This is made use of in the `docker-compose-https.yml` file. Then create a `docker-compose-https.override.yml` file with your settings:

```yaml
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
```

(continued)
Whenever invoking `docker-compose` you need to pass both files to it, and then do:

```
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml build
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml up
```

### Upgrading the Docker container

Usually it is a good idea to only update the Weblate container and keep the PostgreSQL container at the version you have, as upgrading PostgreSQL is quite painful and in most cases does not bring many benefits.

```
Weblate 4.10-1  Docker 4.10-1 PostgreSQL 10 Django 4.0
```

You can do this by sticking with the existing `docker-compose` and just pull the latest images and then restart:

```
# Fetch latest versions of the images
docker-compose pull
# Stop and destroy the containers
docker-compose down
# Spawn new containers in the background
docker-compose up -d
# Follow the logs during upgrade
docker-compose logs -f
```

The Weblate database should be automatically migrated on first startup, and there should be no need for additional manual actions.

---

**Note:** Upgrades across 3.0 are not supported by Weblate. If you are on 2.x series and want to upgrade to 3.x, first upgrade to the latest 3.0.1-x (at time of writing this it is the 3.0.1-7) image, which will do the migration and then continue upgrading to newer versions.

```
docker-compose stop weblate cache
```

```
docker-compose exec database pg_dumpall --clean --username weblate > backup.sql
```

```
docker-compose stop database
```

```
docker-compose rm -v database
```

```
docker volume remove weblate_postgres-data
```

---

**PostgreSQL 12**:  


2. docker-compose stop database

3. docker-compose rm -v database

4. docker-compose volume remove weblate_postgres-data
5. **PostgreSQL**

6. **FILE**: `docker-compose.yml`

   ```bash
docker-compose up -d database
   
   cat backup.sql | docker-compose exec -T database psql --username weblate --
   >>> dbname postgres
   ```

7. **FILE**

8. **FILE**

   ```bash
docker-compose up -d
   ```

---

After container setup, you can sign in as `admin` user with password provided in `WEBLATE_ADMIN_PASSWORD`, or a random password generated on first start if that was not set.

To reset `admin` password, restart the container with `WEBLATE_ADMIN_PASSWORD` set to new password.

The number of worker processes for both uWSGI and Celery is determined automatically based on number of CPUs. This works well for most cloud virtual machines as these typically have few CPUs and good amount of memory.

**CPU**

```yaml
environment:
  WEBLATE_WORKERS: 2
```

You can also fine-tune individual worker categories:

```yaml
environment:
  WEB_WORKERS: 4
  CELERY_MAIN_OPTIONS: --concurrency 2
  CELERY_NOTIFY_OPTIONS: --concurrency 1
  CELERY_TRANSLATE_OPTIONS: --concurrency 1
```

**WEBLATE_WORKERS, CELERY_MAIN_OPTIONS, CELERY_NOTIFY_OPTIONS, CELERY_TRANSLATE_OPTIONS**

---

**Scaling horizontally**

You can run multiple Weblate containers to scale the service horizontally. The `/app/data` volume has to be shared by all containers, it is recommended to use cluster filesystem such as GlusterFS for this. The `/app/cache` volume should be separate for each container.

Each Weblate container has defined role using `WEBLATE_SERVICE` environment variable. Please follow carefully the documentation as some of the services should be running just once in the cluster and the ordering of the services matters as well.

You can find example setup in the `docker-compose` repo as `docker-compose-split.yml`.
Docker environment variables

Many of Weblate's settings can be set in the Docker container using environment variables:

**Generic settings**

**WEBLATE_DEBUG**
Configures Django debug mode using `DEBUG`.

```yaml
environment:
  WEBLATE_DEBUG: 1
```

**WEBLATE_LOGLEVEL**
Configures the logging verbosity.

**WEBLATE_SITE_TITLE**
Changes the site-title shown in the header of all pages.

**WEBLATE_SITE_DOMAIN**

**WEBLATE_ADMIN_NAME**
**WEBLATE_ADMIN_EMAIL**
Configures the site-admin's name and e-mail. It is used for both `ADMINS` setting and creating `admin` user (see `WEBLATE_ADMIN_PASSWORD` for more info on that).

```yaml
environment:
  WEBLATE_ADMIN_NAME: Weblate admin
  WEBLATE_ADMIN_EMAIL: noreply@example.com
```

**WEBLATE_ADMIN_PASSWORD**
Sets the password for the `admin` user.
If not set and `admin` user does not exist, it is created with a random password shown on first container startup.
If not set and `admin` user exists, no action is performed.
If set the `admin` user is adjusted on every container startup to match `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME` and `WEBLATE_ADMIN_EMAIL`.

```yaml
WEBLATE_ADMIN_PASSWORD
```

It might be a security risk to store password in the configuration file. Consider using this variable only for initial setup (or let Weblate generate random password on initial startup) or for password recovery.

**WEBLATE_ADMIN_PASSWORD_FILE**
Sets the path to a file containing the password for the `admin` user.

```yaml
WEBLATE_ADMIN_PASSWORD
```
WEBLATE_SERVER_EMAIL
The email address that error messages are sent from.

WEBLATE_DEFAULT_FROM_EMAIL
The default email address used by email notifications.

WEBLATE_CONTACT_FORM
The email address used for contact form.

WEBLATE_ALLOWED_HOSTS
The list of hosts that are allowed to access Weblate.

WEBLATE_REGISTRATION_OPEN
Configures whether registrations are open by toggling `REGISTRATION_OPEN`.

WEBLATE_REGISTRATION_ALLOW_BACKENDS
Configure which authentication methods can be used to create new account via `REGISTRATION_ALLOW_BACKENDS`.

WEBLATE_TIME_ZONE
Configures the used time zone in Weblate, see `TIME_ZONE`.

To change the time zone of the Docker container itself, use the `TZ` environment variable.

WEBLATE_ENABLE_HTTPS
Makes Weblate assume it is operated behind a reverse HTTPS proxy, it makes Weblate use HTTPS in e-mail and API links or set secure flags on cookies.

Please see `ENABLE_HTTPS` documentation for possible caveats.

This does not make the Weblate container accept HTTPS connections, you need to configure that as well, see `Docker container with HTTPS support` for examples.
WEBLATE_ENABLE_HTTPS: 1

WEBLATE_IP_PROXY_HEADER

This environment variable lets Weblate fetch the IP address from any given HTTP header. Use this when using a reverse proxy in front of the Weblate container.

The environment variable enables IP_BEHIND_REVERSE_PROXY and sets IP_PROXY_HEADER.

WEBLATE_SECURE_PROXY_SSL_HEADER

A tuple representing a HTTP header/value combination that signifies a request is secure. This is needed when Weblate is running behind a reverse proxy doing SSL termination which does not pass standard HTTPS headers.

WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR

WEBLATE_SECURE_PROXY_SSL_HEADER: HTTP_X_FORWARDED_PROTO, https

WEBLATE_REQUIRE_LOGIN

Enables REQUIRE_LOGIN to enforce authentication on whole Weblate.

WEBLATE_REQUIRE_LOGIN: 1

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS
WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS
WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

These environment variables add or remove URL exceptions for authentication required for the whole Weblate installation using LOGIN_REQUIRED_URLS_EXCEPTIONS.

WEBLATE_GOOGLE_ANALYTICS_ID

Configures ID for Google Analytics by changing GOOGLE_ANALYTICS_ID.

WEBLATE_GITHUB_USERNAME

Configures GitHub username for GitHub pull-requests by changing GITHUB_USERNAME.

WEBLATE_GITHUB_TOKEN

Configures GitHub personal access token for GitHub pull-requests via API by changing GITHUB_TOKEN.
WEBLATE_GITLAB_USERNAME
Configures GitLab username for GitLab merge-requests by changing GITLAB_USERNAME.

GitLab

WEBLATE_GITLAB_TOKEN
Configures GitLab personal access token for GitLab merge-requests via API by changing GITLAB_TOKEN.

GitLab

WEBLATE_PAGURE_USERNAME
Configures Pagure merge-requests for Pagure.

Pagure

WEBLATE_PAGURE_TOKEN
Configures Pagure API for Pagure.

Pagure

WEBLATE_DEFAULT_PULL_MESSAGE
Configures DEFAULT_PULL_MESSAGE.

WEBLATE_SIMPLIFY_LANGUAGES
Configures the language simplification policy, see SIMPLIFY_LANGUAGES.

WEBLATE_DEFAULT_ACCESS_CONTROL
Configures the default DEFAULT_ACCESS_CONTROL for new projects, see DEFAULT_ACCESS_CONTROL.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT
Configures the default value for DEFAULT_RESTRICTED_COMPONENT for new components, see DEFAULT_RESTRICTED_COMPONENT.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION
Configures the default value for DEFAULT_TRANSLATION_PROPAGATION for new components, see DEFAULT_TRANSLATION_PROPAGATION.

WEBLATE_DEFAULT_COMMITER_EMAIL
Configures DEFAULT_COMMITER_EMAIL.

WEBLATE_DEFAULT_COMMITER_NAME
Configures DEFAULT_COMMITER_NAME.

WEBLATE_DEFAULT_SHARED_TM
Configures DEFAULT_SHARED_TM.

WEBLATE_AKISMET_API_KEY
Configures the Akismet API key, see AKISMET_API_KEY.

WEBLATE_GPG_IDENTITY
Configures GPG signing of commits, see WEBLATE_GPG_IDENTITY.

WEBLATE_URL_PREFIX
Configures URL prefix where Weblate is running, see URL_PREFIX.

WEBLATE_SILENCED_SYSTEM_CHECKS
Configures checks which you do not want to be displayed, see SILENCED_SYSTEM_CHECKS.

WEBLATE_CSP_SCRIPT_SRC
WEBLATE_CSP_IMG_SRC
WEBLATE_CSP_CONNECT_SRC
WEBLATE_CSP_STYLE_SRC
WEBLATE_CSP_FONT_SRC
Allows to customize Content-Security-Policy HTTP header.

WEBLATE_LICENSE_FILTER
LICENSE_FILTER
WEBLATE_LICENSE_REQUIRED
LICENSE_REQUIRED
WEBLATE_WEBSITE_REQUIRED
WEBSITE_REQUIRED
WEBLATE_HIDE_VERSION
HIDE_VERSION
WEBLATE_BASIC_LANGUAGES
BASIC_LANGUAGES
WEBLATE_DEFAULT_AUTOWATCH
DEFAULT_AUTO_WATCH
WEBLATE_RATELIMIT_ATTEMPTS
WEBLATE_RATELIMIT_LOCKOUT
WEBLATE_RATELIMIT_WINDOW
WEBLATE_API_RATELIMIT_ANON
WEBLATE_API_RATELIMIT_USER
WEBLATE_ENABLE_AVATARS
WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
WEBLATE_SSH_EXTRA_ARGS
WEBLATE_BORG_EXTRA_ARGS

WEBLATE_API_RATELIMIT_ANON
WEBLATE_API_RATELIMIT_USER
WEBLATE_ENABLE_AVATARS
WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
WEBLATE_SSH_EXTRA_ARGS
WEBLATE_BORG_EXTRA_ARGS

---

You can set configuration for any rate limiter scopes. To do that add WEBLATE_ prefix to any of setting described in [here].

WEBLATE_API_RATELIMIT_ANON
WEBLATE_API_RATELIMIT_USER
WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
WEBLATE_SSH_EXTRA_ARGS
WEBLATE_BORG_EXTRA_ARGS
Machine translation settings

- **WEBLATE_MT_APERTIUM_API**
  Enables Apertium machine translation and sets `MT_APERTIUM_API`

- **WEBLATE_MT_AWS_REGION**
  Configures AWS machine translation.

- **WEBLATE_MT_AWS_ACCESS_KEY_ID**
- **WEBLATE_MT_AWS_SECRET_ACCESS_KEY**
  Enables AWS machine translation.

- **WEBLATE_MT_DEEPL_KEY**
  Enables DeepL machine translation and sets `MT_DEEPL_KEY`

- **WEBLATE_MT_LIBRETRANSLATE_KEY**
  Enables LibreTranslate machine translation and sets `MT_LIBRETRANSLATE_KEY`

- **WEBLATE_MT_GOOGLE_KEY**
  Enables Google Translate and sets `MT_GOOGLE_KEY`

- **WEBLATE_MT_GOOGLE_CREDENTIALS**
  Enables Google Translate API V3 (Advanced) and sets `MT_GOOGLE_CREDENTIALS`

- **WEBLATE_MT_GOOGLE_PROJECT**
  Enables Google Translate API V3 (Advanced) and sets `MT_GOOGLE_PROJECT`

- **WEBLATE_MT_GOOGLE_LOCATION**
  Enables Google Translate API V3 (Advanced) and sets `MT_GOOGLE_LOCATION`

- **WEBLATE_MT_MICROSOFT_COGNITIVE_KEY**
  Enables Microsoft Cognitive Services Translator and sets `MT_MICROSOFT_COGNITIVE_KEY`

- **WEBLATE_MT_MICROSOFT_ENDPOINT_URL**
  Sets `MT_MICROSOFT_ENDPOINT_URL`, please note this is supposed to contain domain name only.

- **WEBLATE_MT_MICROSOFT_REGION**
  Sets `MT_MICROSOFT_REGION`

- **WEBLATE_MT_MICROSOFT_BASE_URL**
  Sets `MT_MICROSOFT_BASE_URL`

- **WEBLATE_MT_MODERNMT_KEY**
  Enables ModernMT and sets `MT_MODERNMT_KEY`.

- **WEBLATE_MT_MYMEMORY_ENABLED**
  Enables MyMemory machine translation and sets `MT_MYMEMORY_EMAIL` to `WEBLATE_ADMIN_EMAIL`.

```
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_MYMEMORY_ENABLED: 1
```

**WEBLATE_MT_GLOSBE_ENABLED**

Glosbe
WEBLATE_MT_GLOSBE_ENABLED

Microsoft Terminology Service

WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED

Configures SAP Translation Hub machine translation.

WEBLATE_MT_SAP_BASE_URL
WEBLATE_MT_SAP_SANDBOX_APIKEY
WEBLATE_MT_SAP_USERNAME
WEBLATE_MT_SAP_PASSWORD
WEBLATE_MT_SAP_USE_MT

Authentication settings

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

Example for direct bind:

Example for search and bind:
environment:
WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name, email:mail
WEBLATE_AUTH_LDAP_USER_SEARCH: CN=Users,DC=example,DC=com

Example for union search and bind:

environment:
WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name, email:mail
WEBLATE_AUTH_LDAP_USER_SEARCH_UNION: ou=users,dc=example,ou=otherusers,dc=example,dc=com

Example with search and bind against Active Directory:

environment:
WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS: 0
WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name, email:mail
WEBLATE_AUTH_LDAP_USER_SEARCH: CN=Users,DC=example,DC=com
WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER: (sAMAccountName=%(user)s)

GitHub
WEBLATE_SOCIAL_AUTH_GITHUB_KEY
WEBLATE_SOCIAL_AUTH_GITHUB_SECRET
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID

Bitbucket
WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY
WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET
Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY
WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET
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
Google OAuth 2

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY
WEBLATE_SOCIAL_AUTH_GITLAB_SECRET
WEBLATE_SOCIAL_AUTH_GITLAB_API_URL
GitLab OAuth 2

Azure Active Directory

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET
Enables Azure Active Directory authentication, see Microsoft Azure Active Directory.

Azure Active Directory with Tenant support

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID
Enables Azure Active Directory authentication with Tenant support, see Microsoft Azure Active Directory.

Keycloak

WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY
WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET
WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM
WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL
Enables Keycloak authentication, see documentation.
Linux vendors

You can enable authentication using Linux vendors authentication services by setting following variables to any value.

`WEBLATE_SOCIAL_AUTH_FEDORA`
`WEBLATE_SOCIAL_AUTH_OPENSUSE`
`WEBLATE_SOCIAL_AUTH_UBUNTU`

Slack

`WEBLATE_SOCIAL_AUTH_SLACK_KEY`
`SOCIAL_AUTH_SLACK_SECRET`
Enables Slack authentication, see Slack.

SAML

Self-signed SAML keys are automatically generated on first container startup. In case you want to use own keys, place the certificate and private key in `/app/data/ssl/saml.crt` and `/app/data/ssl/saml.key`.

`WEBLATE_SAML_IDP_ENTITY_ID`
`WEBLATE_SAML_IDP_URL`
`WEBLATE_SAML_IDP_X509CERT`
SAML Identity Provider settings, see SAML.

Other authentication settings

`WEBLATE_NO_EMAIL_AUTH`
Disables e-mail authentication when set to any value. See WEBLATE.

PostgreSQL database setup

`docker-compose.yml`

Weblate PostgreSQL

`POSTGRES_PASSWORD`
PostgreSQL password.

`POSTGRES_PASSWORD_FILE`
Path to the file containing the PostgreSQL password. Use as an alternative to `POSTGRES_PASSWORD`.

`POSTGRES_USER`
PostgreSQL username.

`POSTGRES_DATABASE`
PostgreSQL database name.

`POSTGRES_HOST`
PostgreSQL server hostname or IP address. Defaults to `database`.

`POSTGRES_PORT`
PostgreSQL server port. Defaults to none (uses the default value).

`POSTGRES_SSL_MODE`
Configure how PostgreSQL handles SSL in connection to the server, for possible choices see SSL Mode Descriptions.

`POSTGRES.Alter_ROLE`
Configures name of role to alter during migrations, see Weblate PostgreSQL.
POSTGRES_CONN_MAX_AGE

4.8.1

The lifetime of a database connection, as an integer of seconds. Use 0 to close database connections at the end of each request (this is the default behavior).

Enabling connection persistence will typically cause more open connection to the database. Please adjust your database configuration prior enabling.

```
<table>
<thead>
<tr>
<th>environment:</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSTGRES_CONN_MAX_AGE: 3600</td>
</tr>
</tbody>
</table>
```

CONN_MAX_AGE, Persistent connections

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

4.9.1

Disable server side cursors in the database. This is necessary in some `pgbouncer` setups.

```
<table>
<thead>
<tr>
<th>environment:</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSTGRES_DISABLE_SERVER_SIDE_CURSORS: 1</td>
</tr>
</tbody>
</table>
```

DISABLE_SERVER_SIDE_CURSORS, Transaction pooling and server-side cursors

Database backup settings

WEBLATE_DATABASE_BACKUP

Configures the daily database dump using `DATABASE_BACKUP`. Defaults to `plain`.

Caching server setup

Using Redis is strongly recommended by Weblate and you have to provide a Redis instance when running Weblate in Docker.

```
<table>
<thead>
<tr>
<th>REDIS_HOST</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Redis server hostname or IP address. Defaults to <code>cache</code>.</td>
</tr>
<tr>
<td>REDIS_PORT</td>
</tr>
<tr>
<td>The Redis server port. Defaults to 6379.</td>
</tr>
<tr>
<td>REDIS_DB</td>
</tr>
<tr>
<td>The Redis database number, defaults to 1.</td>
</tr>
<tr>
<td>REDIS_PASSWORD</td>
</tr>
<tr>
<td>The Redis server password, not used by default.</td>
</tr>
<tr>
<td>REDIS_TLS</td>
</tr>
<tr>
<td>Enables using SSL for Redis connection.</td>
</tr>
<tr>
<td>REDIS_VERIFY_SSL</td>
</tr>
<tr>
<td>Can be used to disable SSL certificate verification for Redis connection.</td>
</tr>
</tbody>
</table>
```
Email server setup

To make outgoing e-mail work, you need to provide a mail server.

Example TLS configuration:

```bash
environment:
  WEBSITE_EMAIL_HOST: smtp.example.com
  WEBSITE_EMAIL_HOST_USER: user
  WEBSITE_EMAIL_HOST_PASSWORD: pass
```

Example SSL configuration:

```bash
environment:
  WEBSITE_EMAIL_HOST: smtp.example.com
  WEBSITE_EMAIL_PORT: 465
  WEBSITE_EMAIL_HOST_USER: user
  WEBSITE_EMAIL_HOST_PASSWORD: pass
  WEBSITE_EMAIL_USE_TLS: 0
  WEBSITE_EMAIL_USE_SSL: 1
```

---

**WEBSITE_EMAIL_HOST**

Mail server hostname or IP address.

**WEBSITE_EMAIL_PORT**

Mail server port, defaults to 25.

**WEBSITE_EMAIL_HOST_USER**

**WEBSITE_EMAIL_HOST_PASSWORD**

**WEBSITE_EMAIL_HOST_PASSWORD_FILE**

**WEBSITE_EMAIL_USE_SSL**

Whether to use an implicit TLS (secure) connection when talking to the SMTP server. In most e-mail documentation, this type of TLS connection is referred to as SSL. It is generally used on port 465. If you are experiencing problems, see the explicit TLS setting **WEBSITE_EMAIL_USE_TLS**.

**WEBSITE_EMAIL_USE_TLS**

Whether to use a TLS (secure) connection when talking to the SMTP server. This is used for explicit TLS connections, generally on port 587 or 25. If you are experiencing connections that hang, see the implicit TLS setting **WEBSITE_EMAIL_USE_SSL**.
WEBLATE_EMAIL_PORT
WEBLATE_EMAIL_USE_SSL
WEBLATE_EMAIL_BACKEND
Configures Django back-end to use for sending e-mails.

WEBLATE_AUTO_UPDATE
WEBLATE_AUTO_UPDATE
This is a Boolean setting (use "true" or "false").

WEBLATE_GET_HELP_URL
WEBLATE_STATUS_URL
WEBLATE_LEGAL_URL
WEBLATE_PRIVACY_URL

Error reporting

It is recommended to collect errors from the installation systematically, see WEBLATE_EMAIL_BACKEND.
To enable support for Rollbar, set the following:

ROLLBAR_KEY
Your Rollbar post server access token.

ROLLBAR_ENVIRONMENT
Your Rollbar environment, defaults to production.
To enable support for Sentry, set following:

SENTRY_DSN
Your Sentry DSN.

SENTRY_ENVIRONMENT
Your Sentry Environment (optional).

CDN

WEBLATE_LOCALIZE_CDN_URL
WEBLATE_LOCALIZE_CDN_PATH
4.2.1
Configuration for JavaScript CDN.
The WEBLATE_LOCALIZE_CDN_PATH is path within the container. It should be stored on the persistent volume
and not in the transient storage.
One of possibilities is storing that inside the Weblate data dir:

environment:
 WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
 WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
You are responsible for setting up serving of the files generated by Weblate, it only does stores the files in configured location.

```
weblate-cdn LOCALIZE_CDN_URL LOCALIZE_CDN_PATH
```

3.8-5

The built-in configuration of enabled checks, add-ons or autofixes can be adjusted by the following variables:

WEBLATE_ADD_APPS
WEBLATE_REMOVE_APPS
WEBLATE_ADD_CHECK
WEBLATE_REMOVE_CHECK
WEBLATE_ADD_AUTOFIX
WEBLATE_REMOVE_AUTOFIX
WEBLATE_ADD_ADDONS
WEBLATE_REMOVE_ADDONS

```environment:
  WEBLATE_ADD_AUTOFIX: weblate.trans.autofixes.whitespace.
 WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon,customize.addons.OtherAddon
```

```
CHECK_LIST
AUTOFIX_LIST
WEBLEATE_ADDONS
INSTALLED_APPS
```

WEBLATE_WORKERS

Base number of worker processes running in the container. When not set it is determined automatically on container startup based on number of CPU cores available.

```
WEBLATE_WORKERS
```

```
CELERY_MAIN_OPTIONS
CELERY_NOTIFY_OPTIONS
CELERY_MEMORY_OPTIONS
CELERY_TRANSLATE_OPTIONS
CELERY_BACKUP_OPTIONS
CELERY_BEAT_OPTIONS
```

These variables allow you to adjust Celery worker options. It can be useful to adjust concurrency (--concurrency 16) or use different pool implementation (--pool=gevent).

By default, the number of concurrent workers is based on WEBLATE_WORKERS.

```environment:
  CELERY_MAIN_OPTIONS: --concurrency 16
```
Celery worker options

**WEB_WORKERS**
Configure how many uWSGI workers should be executed.
It defaults to `WEBLATE_WORKERS`.

```
environment:
  WEB_WORKERS: 32
```

**WEBLATE_SERVICE**
Defines which services should be executed inside the container. Use this for Scaling horizontally.

```
WEBLATE_SERVICE:
  celery-beat
  celery-backup
  celery-celery
  celery-memory
  celery-notify
  celery-translate
  web
```

Docker container volumes

There are two volumes (data and cache) exported by the Weblate container. The other service containers (PostgreSQL or Redis) have their data volumes as well, but those are not covered by this document.

The data volume is used to store Weblate persistent data such as cloned repositories or to customize Weblate installation.

The placement of the Docker volume on host system depends on your Docker configuration, but usually it is stored in `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/` (the path consist of name of your docker-compose directory, container, and volume names). In the container it is mounted as `/app/data`.

The cache volume is mounted as `/app/cache` and is used to store static files. Its content is recreated on container startup and the volume can be mounted using ephemeral filesystem such as `tmpfs`.

When creating the volumes manually, the directories should be owned by UID 1000 as that is user used inside the container.

Further configuration customization

You can further customize Weblate installation in the data volume, see Docker container volumes.
Custom configuration files

You can additionally override the configuration in `/app/data/settings-override.py` (see Docker container volumes). This is executed at the end of built-in settings, after all environment settings are loaded, and you can adjust or override them.

Replacing logo and other static files

The static files coming with Weblate can be overridden by placing into `/app/data/python/customize/static` (see Docker container volumes). For example creating `/app/data/python/customize/static/favicon.ico` will replace the favicon.

This approach can be also used to override Weblate templates. For example Weblate documents can be placed into `/app/data/python/customize/templates/legal/documents`. Alternatively you can also include own module (see Customizing Weblate) and add it as separate volume to the Docker container, for example:

```yaml
weblate:
  volumes:
    - weblate-data:/app/data
    - ./weblate_customization/weblate_customization:/app/data/python/
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

Adding own Python modules

You can place own Python modules in `/app/data/python/` (see Docker container volumes) and they can be then loaded by Weblate, most likely by using Custom configuration files.

Customizing Weblate

Configuring PostgreSQL server

The PostgreSQL container uses default PostgreSQL configuration and it won’t effectively utilize your CPU cores or memory. It is recommended to customize the configuration to improve the performance.

The configuration can be adjusted as described in Database Configuration at https://hub.docker.com/_/postgres. The configuration matching your environment can be generated using https://pgtune.leopard.in.ua/.

Debian 🟢 Ubuntu 🟢 3.8-5 🟢 3.8-5 🟢 3.8-5 🟢
apt install \libxml2-dev libxslt-dev libfreetype6-dev libjpeg-dev libz-dev libyamld-dev \libffi-dev libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev \libacl1-dev libssl-dev libpq-dev libjpeg62-turbo-dev build-essential \python3-gdbm python3-dev python3-pip python3-virtualenv virtualenv virtualenv git

apt install tesseract-ocr libtesseract-dev libleptonica-dev
apt install libldap2-dev libldap-common libsasl2-dev
apt install libxmlsec1-dev

# Web server option 1: NGINX and uWSGI
apt install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with `mod_wsgi`
apt install apache2 libapache2-mod-wsgi-py3

# Caching backend: Redis
apt install redis-server

# Database server: PostgreSQL
apt install postgresql postgresql-contrib

# SMTP server
apt install exim4

```
1. Weblate virtualenv
   virtualenv --python=python3 ~/weblate-env

2. Weblate virtualenv
   . ~/weblate-env/bin/activate

3. # pkgconfig is needed to install borgbackup 1.2
   pip install pkgconfig
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi_ 
library seen at runtime is different from the 'ffi.h' file seen at_ 
compile-time)

```
pip install --force-reinstall --no-binary :all: cffi
```

Weblate

```
1. ~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py
2. settings.py
3. Weblate PostgreSQL
4. weblate migrate
5. weblate createadmin
6. weblate collectstatic
7. Celery
8. weblate runserver
```

Web ref: `server`

```
http://localhost:8000/
```

Ctrl+C

Web

2. VCS

Web

Weblate

ref: gettext
Android
Apple
iOS
Java
Stringsdict
Fluent

3. VCS

Web

SUSE
openSUSE

Web

2 GB RAM
2 CPU
1 GB HDD or SSD

Web

2 GB RAM
2 CPU
1 GB HDD or SSD

Web

2 GB RAM
2 CPU
1 GB HDD or SSD

Web

Python

zypper install
libxml2-devel freetype-devel libjpeg-devel zlib-devel
libyaml-devel libffi-devel cairo-devel pango-devel
gobject-introspection-devel libacl-devel python3-pip python3-virtualenv
python3-devel git

zypper install tesseract-ocr tesseract-devel leptonica-devel
zypper install libldap2-devel libtasl2-devel
zypper install libxmlsec1-devel

Web

Celery

# 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

Python

1. Weblate virtualenv:

   virtualenv --python=python3 ~/weblate-env

2. Weblate virtualenv:

   . ~/weblate-env/bin/activate

3. Weblate:

   # pkgconfig is needed to install borgbackup 1.2
   pip install pkgconfig
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"

Weblate

1. Weblate Linux libffi:

   ffi_prep_closure(): bad user_data (it seems that the version of the libffi_
   library seen at runtime is different from the 'ffi.h' file seen at...
   compile-time)

2. PyPI:

   pip install --force-reinstall --no-binary :all: cffi

Weblate

1. -/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py
   ~/weblate-env/lib/python3.7/site-packages/weblate/settings.py

2. settings.py

3. Weblate

   weblate migrate

4. weblate createadmin

5. Web

ref: server ref: static-files
weblate collectstatic

6. JavaScript  CSS  

weblate compress

7. Celery  

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

8. 

weblate runserver

```
http://localhost:8000/  
URL:  
weblate check --deploy
```


2. VCS  

```
ref: gettext  Android  Apple  iOS  Java  Stringsdict  Fluent  
```

3. VCS  

RedHat  Fedora  CentOS  

Weblate  

```
2 GB  RAM  
2 CPU  
1 GB  HDD or SSD
```

100  300 MB  

Weblate  

```

weblate
```

149
Python

dnf install
libxml2-devel freetype-devel libjpeg-devel zlib-devel
libyaml-devel libffi-devel cairo-devel python3-pip python3-virtualenv
python3-devel git

dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
libldap2-devel libsasl2-devel
libxmlsec1-devel

Weblate virtualenv

1. Weblate virtualenv:
   virtualenv --python=python3 ~/weblate-env

2. Weblate virtualenv:
   ./~/weblate-env/bin/activate

3. Weblate:
   # pkgconfig is needed to install borgbackup 1.2
   pip install pkgconfig
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"

libffi:
ffi_prep_closure(): bad user_data (it seems that the version of the libffi
library seen at runtime is different from the 'ffi.h' file seen at compile-time)
pip install --force-reinstall --no-binary :all: cffi

**Weblate**

1.~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py
2.~/weblate-env/lib/python3.7/site-packages/weblate/settings.py
3. Weblate PostgreSQL
4. Weblate migrate
5. Weblate createadmin
6. JavaScript CSS
7. Celery
8. weblate runserver

http://localhost:8000/

Ctrl+C /manage/performance/
weblate check --deploy

2. **VCS**: Weblate
   
   ref: gettext
   
   Android  Apple  iOS  Java  Strings dict  Fluent

3. **macOS**

   **Web**
   
   2 GB RAM
   2 CPU

   1 GB HDD or SSD

   **Python**
   
   brew install python pango cairo gobject-introspection libffi glib libyaml
   pip3 install virtualenv

   pip install brew install tesseract

   brew install tesseract

   # 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
```

## Python

```
# virtualenv
weblate virtualenv virtualenv 
doc: venv: user_guide

1. Weblate virtualenv:
   virtualenv --python=python3 ~/weblate-env

2. Weblate virtualenv:
   . ~/weblate-env/bin/activate

3. Weblate:
   # pkgconfig is needed to install borgbackup 1.2
   pip install pkgconfig
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
```

## Weblate

```
# ffi_prep_closure(): bad user_data (it seems that the version of the libffi-
library seen at runtime is different from the 'ffi.h' file seen at
compile-time)
```

```
# PyPI
pip install --force-reinstall --no-binary :all: cffi
```

## Weblate

```
# Django
settings.py
```

```
1. ~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py
   weblate-env/lib/python3.7/site-packages/weblate/settings.py

2. Django:
   weblate migrate

3. Weblate PostgreSQL
   weblate createadmin

4. Weblate
def ref: server def ref: static-files:

5. Web
def ref: static-files:

6. JavaScript CSS
def ref: static-files:
```

153
weblate compress

7. Celery

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

8. weblate runserver

```
http://localhost:8000/
```

Web

2. VCS

```
weblate check --deploy
```


2. VCS

```
http://localhost:8000/manage/performance/
```

3. VCS

```
/gettext/Android Apple iOS Java Stringsdict Fluent
```

1. Debian Ubuntu

```
https://weblate.org/
```

2. Git Weblate tarball

```
git clone https://github.com/WeblateOrg/weblate.git weblate-src
```

3. Weblate virtualenv

```
. ~/weblate-env/bin/activate
pip install -e weblate-src
```

4. weblate/settings_example.py

```
weblate/settings.py
```

5. Django

```
```

6. Weblate

```
```

7. Django

```
```

154
weblate migrate
weblate collectstatic
weblate compress

OpenShift

OpenShift template-external-postgresql.yml

Web template.yml

CLI

  -n <PROJECT>

  -n <PROJECT> weblate

  -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

$ oc delete all -l app=<APPLICATION_NAME>
$ oc delete configmap -l app=<APPLICATION_NAME>
$ oc delete secret -l app=<APPLICATION_NAME>
  # ATTENTION! The following command is only optional and will permanently...
  delete all of your data.
$ oc delete pvc -l app=<APPLICATION_NAME>

$ oc delete all -l app=weblate \
  && oc delete secret -l app=weblate \
  && oc delete configmap -l app=weblate \
  && oc delete pvc -l app=weblate

helm repo add weblate https://github.com/WeblateOrg/helm/  
  https://artifacthub.io/packages/helm/weblate/weblate

helm install my-release weblate/weblate

Docker environment variables:

Docker Virtualenv:
Debian Ubuntu SUSE openSUSE RedHat Fedora CentOS macOS
Weblate installs PostgreSQL, Redis, and SMTP.

Python installs Weblate and optional packages. Weblate requires Python 3.6.

Install Weblate with:

```
C:\> pip install "Weblate[PHP,Fluent]"
```

Install Weblate with all packages:

```
C:\> pip install "Weblate[all]"
```
pip install Weblate

Webate PostgreSQL MySQL MariaDB

https://git-scm.com/
https://cairographics.org/
https://pango.gnome.org/
Pango Cairo

https://pypi.org/project/git-review/
https://git-scm.com/docs/git-svn
https://github.com/tesseract-ocr/tesseract
https://github.com/licensee/licensee

Python pip Wheels

Pango Cairo

3.7

Weblate Michal Čihař PGP

63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D

https://keybase.io/nijel>

$ 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 87E673AP83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Can't check signature: public key not found

wkd

$ gpg --auto-key-locate wkd --locate-keys michal@cihar.com
pub rsa496 2009-06-17 [SC]
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>
Michal

```
$ gpg --import wmxth3chu9jfxdxywj1skpmsj311mzm
```

```
$ gpg --keyserver hkp://pgp.mit.edu --recv-keys
  87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
```

```
gpg: key 9C27B31342B7511D: "Michal Čihař <michal@cihar.com>" imported
```

```
gpg: Total number processed: 1
```

```
gpg: 
```

```
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>
```

```
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the
```

```
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

```
$ gpg --verify Weblate-3.5.tar.xz.asc
```

```
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Sun Mar 3 16:43:15 2019 CET
```

```
gpg: using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
```

```
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg: aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg: aka "[jpeg image of size 8848]" [ultimate]
gpg: aka "Michal Čihař (Braiins) <michal.cihar@braiins.cz>
```

```
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the
```

```
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

```
$ gpg --verify Weblate-3.5.tar.xz.asc
```

```
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Sun Mar 3 16:43:15 2019 CET
```

```
gpg: using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
```

```
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg: aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg: aka "[jpeg image of size 8848]" [ultimate]
gpg: aka "Michal Čihař (Braiins) <michal.cihar@braiins.cz>
```

```
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the
```

```
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

Weblate

```
Weblate [DATA_DIR] WSGI Celery
```

```
Weblate [DATA_DIR] WSGI Celery
```

```
Docker /app/data weblate /app/data weblate UID 1000
```
**PostgreSQL notes**

# If PostgreSQL was not installed before, set the main password
```bash
sudo -u postgres psql postgres -c "$password postgres"
```

# Create a database user called "weblate"
```bash
sudo -u postgres createuser --superuser --pwprompt weblate
```

# Create the database "weblate" owned by "weblate"
```bash
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

```py
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
    }
}
```
Weblate: ALTER ROLE weblate@hostname

```
psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist
```

Azure Database for PostgreSQL Weblate

MySQL  MariaDB

```
ALTER_ROLE
```

MySqlnotes  MariaDBnotes

Weblate: MySQL  MariaDB Django

```
UTF8MB4 Unicode

innodb_file_per_table

READ COMMITTED

SQL STRICT_TRANS_TABLES
```

MySQL  MariaDB

```
/etc/my.cnf.d/server.cnf

innodb_file_per_table

innodb_large_prefix

innodb_large_prefix = 1

innodb_file_format = Barracuda

innodb_file_per_table = 1

innodb_buffer_pool_size = 2G

sql_mode = STRICT_TRANS_TABLES
```

```
#1071 - Specified key was too long; max key length is 767 bytes
```

```
#2006 - MySQL server has gone away
```

161
settings.py

```python
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": {},
    }
}
```

MySQL  MariaDB  weblate

```bash
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

SMTP

```python
EMAIL_HOST = EMAIL_HOST_PASSWORD = EMAIL_USE_TLS = EMAIL_USE_SSL = EMAIL_HOST_USER = EMAIL_PORT = DJANGO
```

SMTP AUTH extension not supported by server.

Not receiving e-mails from Weblate. Configuring outgoing e-mail in Docker container.

IP

```bash
IP BEHIND REVERSE_PROXY  IP_PROXY_HEADER  IP_PROXY_OFFSET  SECURE_PROXY_SSL
```

Apache

```bash
ProxyPreserveHost On
```

Docker

```bash
IP_BEHIND_REVERSE_PROXY  IP_PROXY_HEADER  IP_PROXY_OFFSET  SECURE_PROXY_SSL
```
HTTP

```python
import os

os.environ['http_proxy'] = 'http://proxy.example.com:8080'
os.environ['HTTPS_PROXY'] = 'http://proxy.example.com:8080'
```

```python
ADMINS = ['

ALLOWED_HOSTS = ['demo.weblate.org']

ALLOWED_HOSTS = ['*']
```

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

```python
DEBUG
```

```python
DEFAULT_FROM_EMAIL
```
```python
from django.conf import settings

WEBSITE = settings.SITE_URL

DEBUG = False
SECRET_KEY = 'supersecretkey'
SERVER_EMAIL = 'info@mywebsite.com'

# Django settings

WEBSITE = settings.SITE_URL

DEBUG = False
SECRET_KEY = 'supersecretkey'
SERVER_EMAIL = 'info@mywebsite.com'

# weblate settings

weblate migrate --noinput createadmin

weblate check --deploy
```

Choose what languages you want in the preferences, to see overview of available translations for those languages in your watched projects.

Powered by Weblate 4.1

Django DEBUG = False

```
ADMINS = ("Your Name", "your_email@example.com"),

HTTPS

DISABLED

ENABLED

ENABLED

SECURE_HSTS_SECONDS

0

POSTGRESQL

Weblate

PGTune

Weblate

PostgreSQL
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",
        },
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 604800,
        "OPTIONS": {
            "MAX_ENTRIES": 1000,
        },
    },
}

Weblate EMAIL_BACKEND = django.core.mail.backends.dummy.EmailBackend
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "admin@example.org"

Django's cache framework

Weblate EMAIL_BACKEND = django.core.mail.backends.dummy.EmailBackend
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "admin@example.org"
EMAIL_BACKEND
DEFAULT_FROM_EMAIL
SERVER_EMAIL

HTTP '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS.

Docker WEBLATE_ALLOWED_HOSTS

ALLOWED_HOSTS
WEBLATE_ALLOWED_HOSTS

Django

SECRET_KEY

Weblate generates a SECRET_KEY cookie.

Weblate configuration

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

Linux

```
/etc/passwd
/etc/web/WEB
```

Weblate configuration

```python
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",
            ],
        },
    },
]  ```

`167`
```
    "weblate.trans.context_processors.weblate_context",
    },
    "loaders": [
        {
            "django.template.loaders.cached.Loader",
            ["django.template.loaders.filesystem.Loader",
             "django.template.loaders.app_directories.Loader",
             ],
        },
    ],
},
```

**Django Template Loaders**

- ```django.template.loaders.cached.Loader``
- ```django.template.loaders.filesystem.Loader``
- ```django.template.loaders.app_directories.Loader``

**Celery**

- Lazy commits: `commit_pending`
- AUTO_UPDATE
- ```dump_memory```
- ```cleanuptrans```

**Weblate 3.2**

- Celery Weblate

**Ubuntu**

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

**CentOS**

- ```export LANG='en_US.UTF-8'```
Weblate **HTTP**  **SSL**  Debian **CA**  **update-ca-certificates**  

```bash
 Weblate HTTP  SSL  Debian CA update-ca-certificates
```

Python  CA  CA  settings.py  Debian  

```python
import os
os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Weblate **JavaScript**  **CSS**  Weblate  

```javascript
import os
os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Weblate **JavaScript**  **CSS**  Weblate  

```javascript
import os
os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

**Docker**  **Common Deployment Scenarios**  

```bash
weblate compress
```

**Docker**  **Common Deployment Scenarios**  

```bash
weblate compress
```

**Docker**  **Common Deployment Scenarios**  

```bash
weblate compress
```

**Docker**  **Common Deployment Scenarios**  

```bash
weblate compress
```
Web

Weblate Django weblate.runserver

Django runserver

Django runserver DEBUG

NGINX uWSGI Apache Gunicorn

How to deploy

How to deploy static files

Nginx uWSGI

WEB Weblate wsgi virtual env

~/.weblate-env/lib/python3.7/site-packages/weblate/wsgi.py

Python virtualenv = /home/user/weblate-env

NGINX WEB Weblate uWSGI

NGINX weblate/examples/weblate.nginx.conf

# This example assumes Weblate is installed in virtualenv in /home/weblate/
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match...
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;
}

uWSGI [example] weblate/examples/weblate.uwsgi.ini

# This example assumes Weblate is installed in virtualenv in /home/weblate/
# weblate-env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match...
# your setup.
[uwsgi]
plugins = python3
master = true
protocol = uwsgi
socket = 127.0.0.1:8080
wsgi-file = /home/weblate/weblate-env/lib/python3.9/site-packages/
    weblate/wsgi.py

# Add path to Weblate checkout if you did not install
# Weblate by pip
# python-path = /path/to/weblate

# In case you're using virtualenv uncomment this:
# virtualenv = /home/weblate/weblate-env

# Needed for OAuth/OpenID
buffer-size = 8192

# Reload when consuming too much of memory
reload-on-rss = 250

# Increase number of workers for heavily loaded sites
workers = 8

# Enable threads for Sentry error submission
enable-threads = true

# Child processes do not need file descriptors
close-on-exec = true

# Avoid default 0000 umask
umask = 0022
# Run as weblate user
uid = weblate
gid = weblate

# Enable harakiri mode (kill requests after some time)
# harakiri = 3600
# harakiri-verbose = true

# Enable uWSGI stats server
# stats = :1717
# stats-http = true

# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true

---

Django [ uWSGI ]

Apache

Weblate [ WSGI ]

```text
VirtualHost *:80
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env
    user=weblate request-timeout=600
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}
    WSGIScriptAlias / /home/weblate/weblate-env/lib/python3.7/site-packages/weblate/wsgi.py process-group=weblate
    WSGIPassAuthorization On
    <Directory /home/weblate/weblate-env/lib/python3.7/site-packages/weblate/>
```
<Files wsgi.py>
  Require all granted
</Files>
</Directory>

<VirtualHost *: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>
# # VirtualHost for Weblate, running under /weblate path
# # This example assumes Weblate is installed in virtualenv in /home/weblate/
# # and DATA_DIR is set to /home/weblate/data, please adjust paths to match your setup.
#<VirtualHost *:80>
  ServerAdmin admin@weblate.example.org
  ServerName weblate.example.org
  # DATA_DIR/static/favicon.ico
  Alias /weblate/favicon.ico /home/weblate/data/static/favicon.ico
  # DATA_DIR/static/
  Alias /weblate/static/ /home/weblate/data/static/
  <Directory /home/weblate/data/static/>
    Require all granted
  </Directory>
  # DATA_DIR/media/
  Alias /weblate/media/ /home/weblate/data/media/
  <Directory /home/weblate/data/media/>
    Require all granted
  </Directory>
  # Path to your Weblate virtualenv
  WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env
  user=weblate request-timeout=600
  WSGIProcessGroup weblate
  WSGIApplicationGroup %{GLOBAL}
  WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.7/site-packages/weblate/wsgi.py process-group=weblate
  WSGIPassAuthorization On
  <Directory /home/weblate/weblate-env/lib/python3.7/site-packages/weblate/>
    <Files wsgi.py>
      Require all granted
    </Files>
  </Directory>
</VirtualHost>

# weblate/settings.py
URL_PREFIX = ""
Celery

Weblate

Redis broker configuration in Celery

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

```
[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

/etc/default/celery-weblate:

# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
CELERY_BIN="/home/weblate/weblate-env/bin/celery"

# App instance to use
# comment out this line if you don't use an app
CELERY_APP="weblate.utils"

# Extra command-line arguments to the worker,
# increase concurrency if you get weblate.E019
CELERYD_OPTS="--beat:celery --queues:celery=celery --prefetch-multiplicator:celery=4 \ 
--queues:notify=notify --prefetch-multiplicator:notify=10 \ 
--queues:memory=memory --prefetch-multiplicator:memory=10 \ 
--queues:translate=translate --prefetch-multiplicator:translate=4 \ 
--concurrency:backup=1 --queues:backup=backup --prefetch-multiplicator: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"

/etc/logrotate.d/celery:

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

Weblate

settings.py

Lazy commits

Celery beats

Celery

celery_queues

Configuration and defaults

Workers Guide

Daemonization

Monitoring and Management

Guide

celery_queues

Weblate Kubernetes

/healthz/

Docker URL

GET /api/metrics/

API

`Munin Website <https://github.com/WeblateOrg/munin>`_

Sentry

Weblate

Rollbar

Rollbar notifier for Python

settings.py

SENTRY_DSN

= "https://id@your.sentry.example.com/"

# 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",
}
# Export current data
weblate dumpdata > /tmp/weblate.dump

# Import dump
weblate loaddata /tmp/weblate.dump
Weblate Cloudron

YunoHost Weblate

Docker

Upgrading the Docker container

1. wsgi Celery
2. Weblate

pip

```
cd weblate-src
git pull
# Update Weblate inside your virtualenv
. ~/weblate-env/bin/pip install -e
# Install dependencies directly when not using virtualenv
pip install --upgrade -r requirements.txt
# Install optional dependencies directly when not using virtualenv
pip install --upgrade -r requirements-optional.txt
```
3. Weblate
   settings_example.py

5. 
   weblate migrate --noinput

6. 
   weblate collectstatic --noinput --clear

7. JavaScript, CSS
   weblate compress

8. Git
   weblate compilemessages

9. 
   weblate check --deploy

10. Celery
    Celery

2.x

2.x
   3.0.1  3.0.2  3.0.3  3.0.4

   Weblate 3.0

3.x

3.x
   4.0.4  4.1.1

   Weblate 4.0

4.0  4.1

settings_example.py

MT_DEEPL_API_VERSION  4.7
MT_DEEPL_API_URL:setting:MT_DEEPL_API_URL
4.1 4.2 4.3

settings_example.py

4.0.4 4.1.1

CELERY_TASK_ROUTES

3.x 4.x

Check list

4.2 4.3

hub lab

hub

GitHub

GitLab

hub

CREDENTIALS

hub lab

weblate configuration

4.4

GETTEXT

gettext

gettext

hub lab

gettext

gettext

hub lab

gettext

gettext

hub lab

gettext
4.4 API INSTALLED_APPS

weblate.metrics

4.5.1 pyahocorasick

4.5 API INSTALLED_APPS

weblate.metrics

API INSTALLED_APPS

weblate.metrics

4.6.1 pyahocorasick

4.5.2 API INSTALLED_APPS

weblate.metrics

API INSTALLED_APPS

weblate.metrics

4.6 API INSTALLED_APPS

webhate.metrics

API INSTALLED_APPS

webhate.metrics

4.7.1 pyahocorasick

4.6 API WEBLATE_FORMATS

API Webhate URL

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

API WEBLATE_FORMATS

API Webhate URL

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

4.7 API WEBLATE_FORMATS

API Webhate URL

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

API WEBLATE_FORMATS

API Webhate URL

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

4.8.1 pyahocorasick

4.7 API MIDDLEWARE

settings_example.py

MIDDLEWARE

settings_example.py

DeepL

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2

MT_DEEPL_API_URL

MT_DEEPL_API_VERSION

Django 3.2
4.8 4.9 4.10

Django 4.0  Weblate  Django 3.2  Weblate  
PostgreSQL 9.6  Django 4.0  PostgreSQL 10  
CSRF_TRUSTED_ORIGINS

Docker  Django 4.0  

4.9 4.10 4.11

Weblate  Python 3.7  API  
Weblate  chardet  charset-normalizer

Changed in 4.11.1: There is a change in REST_FRAMEWORK setting (removal of one of the backends in DEFAULT_AUTHENTICATION_CLASSES).

Python 2  Python 3

PostgreSQL 2022
**PostgreSQL**

1. If PostgreSQL was not installed before, set the main password
   ```
   sudo -u postgres psql postgres -c "\password postgres"
   ```

2. Create a database user called "weblate"
   ```
   sudo -u postgres createuser -D -P weblate
   ```

3. Create the database "weblate" owned by "weblate"
   ```
   sudo -u postgres createdb -E UTF8 -O weblate weblate
   ```

**Django JSON**

Django JSON

```python
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...
            '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...
            "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": "",
    }
}
```
weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --
  --database=postgresql

3. PostgreSQL:
weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql

4. PostgreSQL DATABASES

pgloader PostgreSQL

pgloader PostgreSQL settings.py
1. PostgreSQL settings.py
2. PostgreSQL

pgloader

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';

Pootle

Webate

BorgBackup

14466185

3.9 FM.

Webate

Borg

4.4.1 FM: PostgreSQL & MySQL/MariaDB

Borg

6.9

Borg 6.9

185
**Borg**

**BorgBackup**

**Weblate**

SSH: `borg init`

**DATABASE_BACKUP**

**BORG_EXTRA_ARGS**

**Weblate**

1. [https://weblate.org/support/#backup](https://weblate.org/support/#backup)
2. 
3. 
4. **Borg**: `borg init`
5. **Weblate**: `borg init`

**Docker**

- `/path/to/backup`

**Docker Compose**:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

**UID 1000**

Weblate
Weblate SSH: SSH:
1. Installation
2. BorgBackup
3. Weblate SSH
4. Weblate
5. BorgBackup
6. Weblate

BorgBackup

1. installation
2. borg list REPOSITORY
3. borg extract REPOSITORY::ARCHIVE
4. Weblate
5. Weblate backups/settings.py
6. Docker

Docker container volumes:

Borg:

$ borg list /tmp/xxx
Enter passphrase for key /tmp/xxx:
Thu, 2019-09-26 14:56:08...
[de0e0f136435d35d090e9896bedc92a023050749ad3f2350e788f1a65576a5]
$ borg extract /tmp/xxx::2019-09-26T14:56:08
Enter passphrase for key /tmp/xxx:

Weblate

SILENCED_SYSTEM_CHECKS = weblate.I028
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
**Django**:

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pg_dump</code></td>
<td>Dumps PostgreSQL database data to a file.</td>
</tr>
<tr>
<td><code>mysqldump</code></td>
<td>Dumps MySQL database data to a file.</td>
</tr>
<tr>
<td><code>Django</code></td>
<td>Invokes management commands.</td>
</tr>
<tr>
<td><code>dumpdata</code></td>
<td>Generates a dump of the database data.</td>
</tr>
<tr>
<td><code>migrate</code></td>
<td>Migrates the database to a given state.</td>
</tr>
</tbody>
</table>

**Invoking management commands**

```bash
weblate shell
g>>> from weblate.auth.models import User
g>>> User.objects.get(username='anonymous').delete()
```

**Django environment**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DATA_DIR</code></td>
<td>Directory for backups and logs.</td>
</tr>
</tbody>
</table>

**PostgreSQL database**

```bash
psql --file=database.sql weblate
```
**DATA_DIR** /vcs

**SSH**

**DATA_DIR** /ssh

**DATA_DIR** /home

**DATA_DIR** /media

**Visual context for strings**

**Celery**

**Celery**

**Celery**

**cron job**

```bash
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz backups vcs ssh home media fonts secret
```

https://linux.die.net/man/1/xz

```bash
$ 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
```

**updategit**

```bash
weblate updategit --all
```
Welcome to Python Social Auth’s documentation!  

**SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True**

**Pipeline**

**AUTHENTICATION_BACKENDS**

**PYTHON SOCIAL AUTH**

**Weblate**

**PostgreSQL**

**Docker**

**GitLab**

**Ubuntu**

**Fedora**

**Django**

**Pootle**

**Weblate FORCE_EMAIL_VALIDATION**

**SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION**

**ENABLE_HTTPS**

**WEBLATE_ENABLE_HTTPS**

**Python Social Auth backend**
# Authentication configuration

```python
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",  
)
```

---

**GitHub**

GitHub OAuth

```python
AUTHENTICATION_BACKENDS = (  
    "social_core.backends.github.GithubOAuth2",  
    "social_core.backends.email.EmailAuth",  
    "weblate.accounts.auth.WeblateUserBackend",  
)
```

## Social auth backends setup

```
SOCIAL_AUTH_GITHUB_KEY = "GitHub Client ID"
SOCIAL_AUTH_GITHUB_SECRET = "GitHub Client Secret"
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
```

GitHub URL: https://example.com/accounts/complete/github/

There are similar authentication backends for GitHub for Organizations and GitHub for Teams. Their settings are named `SOCIAL_AUTH_GITHUB_ORG_*` and `SOCIAL_AUTH_GITHUB_TEAM_*`, and they require additional setting of the scope - `SOCIAL_AUTH_GITHUB_ORG_NAME` or `SOCIAL_AUTH_GITHUB_TEAM_ID`. Their callback URLs are `https://example.com/accounts/complete/github-org/` and `https://example.com/accounts/complete/github-teams/`.

---

**Bitbucket**

```python
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
```
Weblate URL

Bitbucket

Google OAuth 2

Google OAuth 2

[https://console.developers.google.com/]

Google+API

https://WEBLATE SERVER/accounts/complete/google-oauth2/

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"

Facebook OAuth 2

Facebook OAuth 2

https://WEBLATE SERVER/accounts/complete/facebook/

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]

Facebook
GitLab OAuth 2

GitLab OAuth 2 ✗ [https://gitlab.com/profile/applications](https://gitlab.com/profile/applications)

**URL** ✗ [https://WEBLATE SERVER/accounts/complete/gitlab/](https://WEBLATE SERVER/accounts/complete/gitlab/)

**read_user** ✗ [https://WEBLATE SERVER/accounts/complete/gitlab/](https://WEBLATE SERVER/accounts/complete/gitlab/)

```python
# 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/'
```

---

Microsoft Azure Active Directory

**Weblate** ✗ [https://WEBLATE SERVER/accounts/complete/azuread-oauth2/](https://WEBLATE SERVER/accounts/complete/azuread-oauth2/)

**URL** ✗ [https://WEBLATE SERVER/accounts/complete/azuread-tenant-oauth2/](https://WEBLATE SERVER/accounts/complete/azuread-tenant-oauth2/)

# Azure AD common

```python
# 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

```python
# Authentication configuration
AUTHENTICATION_BACKENDS = (  
    "social_core.backends.azuread_tenant.AzureADTenantOAuth2",  
    "social_core.backends.email.EmailAuth",  
    "weblate.accounts.auth.WeblateUserBackend",  
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET = ""

# Tenant ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID = ""
```
Microsoft Azure Active Directory

**Slack**

Slack OAuth 2 URL [https://api.slack.com/apps](https://api.slack.com/apps)

```python
# 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 = ""
```

**Slack**

```python
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

**openSUSE Open ID**

```python
# Authentication configuration
AUTHENTICATION_BACKENDS = ("social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
```

openSUSE Open ID
SAML

Python Social Auth

Webate IDP

SAML XML URL

SAML AUTHENTICATION configuration

```
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.saml.SAMLAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

```
# Social auth backends setup
SOCIAL_AUTH_SAML_SP_ENTITY_ID = "https://{SITE_DOMAIN}/accounts/metadata/saml/"
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = [
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbnl+ev0pe1xzFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
]
```

```
SOCIAL_AUTH_SAML_ORG_INFO = {
    "en-US": {
        "name": "example",
        "displayname": "Example Inc.",
        "url": "http://example.com"
    }
}
```

```
SOCIAL_AUTH_SAML_TECHNICAL_CONTACT = {
    "givenName": "Tech Gal",
    "emailAddress": "technical@example.com"
}
```

```
SOCIAL_AUTH_SAML_SUPPORT_CONTACT = {
    "givenName": "Support Guy",
    "emailAddress": "support@example.com"
}
```
Configuring SAML in Docker

SAML

LDAP

# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap

Docker:

Python LDAP 3.1.0:

```
AttributeError: 'module' object has no attribute '_trace_level'
```

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\},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
```

197
AUTHENTICATION_BACKENDS = [ 'social_core.backends.email.EmailAuth',
'weblate.accounts.auth.WeblateUserBackend',
'createadmin'
]

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)"
)

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

Django Authentication Using LDAP Authentication
# CAS

## CAS

**django-cas-ng**

**CAS v 1**

**CAS v 2**

**Weblate**

**django-cas-ng**

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

**settings.py**

```python
# 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")
```

**urls.py**

```python
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()
```

---

## Django CAS NG

**Django CAS NG**

**Django**

**Weblate**

**Weblate user backend**

**LDAP**

**CAS**

```python
# Add authentication backend here
"weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (  
    # Install authentication app here
)```
Publicly visible, translatable for all signed-in users.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.

You do not have permission to change project access control. Check your billing status.
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 team) would be sufficient.

---

**3.11 Users management interface**

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 team) would be sufficient.
Weblate manage users

default groups

Weblate

default groups

`ref:` automatic group assignment <autogroup>

`ref:` review workflow <reviews>

Table 2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

203
Weblate

<table>
<thead>
<tr>
<th>Role</th>
<th>Project</th>
<th>Language</th>
<th>Components</th>
<th>Component list</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Group</td>
<td>Role</td>
<td>Project</td>
<td>Language</td>
</tr>
<tr>
<td>Group</td>
<td>Role</td>
<td>Project</td>
<td>Language</td>
<td>Components</td>
</tr>
<tr>
<td>User</td>
<td>Group</td>
<td>Role</td>
<td>Project</td>
<td>Language</td>
</tr>
</tbody>
</table>

Weblate
Weblate setupgroups

Weblate: ANONYMOUS_USER_NAME

automatic group assignment: none

automatic group assignment: Power user

Review strings

Administration

REGISTRATION_OPEN: False

Weblate URL: 

invitations

Weblate Translate Toolkit: 

VCS: 

W
Django 3.2

Django 3.4

https://hosted.weblate.org/
Django

URL

"Language-Team"
The language codes are mapped when matching the translation files and the matches are case sensitive, so make sure you use the source language codes in same form as used in the filenames.
Pushing changes from Weblate

URL

URL of repository browser used to display source files (location of used messages). When empty, no such links will be generated. You can use \url{weblate://project/component}. For example on GitHub, use something like:

\url{https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}}

In case your paths are relative to different folder (path contains ..), you might want to strip leading directory by \url{parentdir} filter (see \url{weblate://project/component}):

\url{https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}}

214
URL

URL where changes made by Weblate are exported. This is important when [weblate] is not used, or when there is a need to manually merge changes. You can use Git [weblate] to automate this for Git repositories.

push

Which branch to checkout from the VCS, and where to look for translations.

push

Branch for pushing changes, leave empty to use [weblate].

push

This is currently only supported for Git, GitLab and GitHub, it is ignored for other VCS integrations.

push

Pushing changes from Weblate

push

Mask of files to translate, including path. It should include one "*" replacing language code (see [weblate] for info on how this is processed). In case your repository contains more than one translation file (e.g. more gettext domains), you need to create a component for each of them.

push

In case your filename contains special characters such as [, ], these need to be escaped as [[]] or [ ].

push

When set, the source strings are based on this file, but all other languages are based on [weblate]. In case the string is not translated into the source language, translating to other languages is prohibited. This provides

push
Base file used to generate new translations, e.g. .pot file with gettext.

**adding-translation**

Translation file format, see also .po.

You can turn off propagation of translations to this component from other components within same project. This really depends on what you are translating, sometimes it's desirable to have make use of a translation more than once. It's usually a good idea to turn this off for monolingual translations, unless you are using the same IDs across the whole project.

Default value can be changed by `DEFAULT_TRANSLATION_PROPAGATION`.

**Keeping translations same across components**

Whether translation suggestions are accepted for this component.

**Turns on vote casting for suggestions, see [voting](#).**

Automatically accept voted suggestions, see [voting](#).
List of checks which can not be ignored, see [here](#).

- **Enforcing the check does not automatically enable it, you still should enabled it using [here](#) or Additional info on source strings.**

License of the translation (does not need to be the same as the source code license).

How to handle requests for creation of new languages. Available options:

User can select desired language and the project maintainers will receive a notification about this. It is up to them to add (or not) the language to the repository.

User is presented a link to page which describes process of starting new translations. Use this in case more formal process is desired (for example forming a team of people before starting actual translation).

User can select language and Weblate automatically creates the file for it and translation can begin. There will be no option for user to start new translation.

- The project admins can add new translations even if it is disabled here when it is possible (either [here](#) or the file format supports starting from an empty file).

**adding-translation**

**4.5**

Configures whether users in Weblate will be allowed to add new strings and remove existing ones. Adjust this to match your localization workflow - how the new strings are supposed to be introduced.

For bilingual formats, the strings are typically extracted from the source code (for example by using `xgettext`) and adding new strings in Weblate should be disabled (they would be discarded next time you update the translation files). In Weblate you can manage strings for every translation and it does not enforce the strings in all translations to be consistent.

For monolingual formats, the strings are managed only on source language and are automatically added or removed in the translations. The strings appear in the translation files once they are translated.

**adding-new-strings**

**POST** `/api/translations/(string:project)/(string:component)/(string:language)/units/`
Rebasing can cause you trouble in case of complicated merges, so carefully consider whether or not you want to enable them.

You might need to enable force pushing by choosing Git as VCS, especially when pushing to a different branch.

Upstream repository changes are merged into Weblate one. This setting utilizes fast-forward when possible. This is the safest way, but might produce a lot of merge commits.

Upstream repository changes are merged into Weblate one with doing a merge commit every time (even when fast-forward would be possible). Every Weblate change will appear as a merge commit in Weblate repository.

Default value can be changed by DEFAULT_MERGE_STYLE.

Message used when committing a translation, see DEFAULT_MERGE_MESSAGE.

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 Lazy commits). To actually enable pushing Repository push URL has to be configured as well.

Sets how old (in hours) changes have to be before they are committed by background task or the commit_pending management command. All changes in a component are committed once there is at least one change older than this period.

Default value can be changed by COMMIT_PENDING_HOURS.

There are other situations where pending changes might be committed, see Lazy commits.

Locks the component (and linked components, see Weblate URL) upon the first failed push or merge into its upstream repository, or pull from it. This avoids adding another conflicts, which would have to be resolved manually. The component will be automatically unlocked once there are no repository errors left.
Language used for source strings. Change this if you are translating from something else than English.

In case you are translating bilingual files from English, but want to be able to do fixes in the English translation as well, choose *English (Developer)* as a source language to avoid conflict between the name of the source language and the existing translation.

For monolingual translations, you can use intermediate translation in this case, see [here](#).

You need to list language codes as they appear in the filename.

Some examples of filtering:

<table>
<thead>
<tr>
<th>Filter description</th>
<th>Filter description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected languages only</td>
<td>`(cs</td>
</tr>
<tr>
<td>Exclude languages</td>
<td>`^(?!([it</td>
</tr>
<tr>
<td>Filter two letter codes only</td>
<td><code>^..$</code></td>
</tr>
<tr>
<td>Exclude non language files</td>
<td><code>^(?!([blank])+$</code></td>
</tr>
<tr>
<td>Include all files (default)</td>
<td><code>^[.]+$</code></td>
</tr>
</tbody>
</table>

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 glossary will be accessible in all projects defined by `PROJECT_NAME`. It is recommended to enable `USE_GLOSSARIES` on glossaries in order to allow adding new words to them.

- `USE_GLOSSARIES`

Display color for a glossary used when showing word matches.

- `PROJECT_NAME`: The Django Template Language
- `URL`: The URL
{% if stats.translated_percent > 80 %}Well translated!{% endif %}

{% replace component "-" " " %}

{% replace component|capfirst "-" " " %}

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 }}

....Django...

VCS Weblate:

Weblate ref: production

Celery:

CPU: 221
Weblate ISO 639-1

-- cs_CZ [xx_XX (generated)]

URL /languages/
In many cases it is not a good idea to use macrolanguage code for a translation. The typical problematic case might be Kurdish language, which might be written in Arabic or Latin script, depending on actual variant. To get correct behavior in Weblate, it is recommended to use individual language codes only and avoid macrolanguages.

Weblate ISO 639-1
Weblate ISO 639-2
Weblate ISO 639-3
Weblate BCP 47

Gettext

GNU gettext utilities: Plural forms

Language Plural Rules by the Unicode Consortium
This is the process:
1. Developers make changes and push them to the VCS repository.
2. Optionally the translation files are updated (this depends on the file format, see Weblate).
3. Weblate pulls changes from the VCS repository.
4. Once Weblate detects changes in translations, translators are notified based on their subscription settings.
5. Translators submit translations using the Weblate web interface, or upload offline changes.
6. Once the translators are finished, Weblate commits the changes to the local repository (see Lazy commits) and pushes them back if it has permissions to do so (see Pushing changes from Weblate).
You should set up some way of updating backend repositories from their source.

Automatically receiving changes from GitHub
Automatically receiving changes from GitLab
Automatically receiving changes from Bitbucket
Pagure

Automatically receiving changes from Azure Repos

The merge conflicts from Weblate arise when same file was changed both in Weblate and outside it. There are two approaches to deal with that - avoid edits outside Weblate or integrate Weblate into your updating process, so that it flushes changes prior to updating the files outside Weblate.

The first approach is easy with monolingual files - you can add new strings within Weblate and leave whole editing of the files there. For bilingual files, there is usually some kind of message extraction process to generate translatable files from the source code. In some cases this can be split into two parts - one for the extraction generates template (for example gettext POT is generated using \texttt{xgettext}) and then further process merges it into actual translations (the gettext PO files are updated using \texttt{msgmerge}). You can perform the second step within Weblate and it will make sure that all pending changes are included prior to this operation.
The second approach can be achieved by using Weblate REST API to force Weblate to push all pending changes and lock the translation while you are doing changes on your side.

The script for doing updates can look like this:

```bash
# 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
# Push changes to upstream repository
git push
# Tell Weblate to pull changes (not needed if Weblate follows your repo automatically)
wlc pull
# Unlock translations
wlc unlock
```

If you have multiple components sharing same repository, you need to lock them all separately:

```bash
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

---

*Note:* The example uses Weblate, which needs configuration (API keys) to be able to control Weblate remotely. You can also achieve this using any HTTP client instead of wlc, e.g. curl, see Weblate REST API.

---

*Note:*
Weblate

---

Automatically receiving changes from GitHub

Weblate comes with native support for GitHub.

If you are using Hosted Weblate, the recommended approach is to install the Weblate app, that way you will get the correct setup without having to set much up. It can also be used for pushing changes back.

To receive notifications on every push to a GitHub repository, add the Weblate Webhook in the repository settings (Webhooks) as shown on the image below:
For the payload URL, append /hooks/github/ to your Weblate URL, for example for the Hosted Weblate service, this is https://hosted.weblate.org/hooks/github/.

You can leave other values at default settings (Weblate can handle both content types and consumes just the push event).

**POST /hooks/github/**

**Automatically receiving changes from Bitbucket**

Weblate has support for Bitbucket webhooks, add a webhook which triggers upon repository push, with destination to /hooks/bitbucket/ URL on your Weblate installation (for example https://hosted.weblate.org/hooks/bitbucket/).
Automatically receiving changes from GitLab

Weblate has support for GitLab hooks, add a project webhook with destination to /hooks/gitlab/ URL on your Weblate installation (for example: https://hosted.weblate.org/hooks/gitlab/).

Pagure

Pagure 3.3rc4.

Webhooks: https://hosted.weblate.org/hooks/pagure/ URL WEB

Project options: Activate Web-hooks.
POST /hooks/pagure/

Automatically receiving changes from Azure Repos

Weblate has support for Azure Repos web hooks, add a webhook for Code pushed event with destination to /hooks/azure/ URL on your Weblate installation (for example https://hosted.weblate.org/hooks/azure/). This can be done in Service hooks under Project settings.

Web hooks in Azure DevOps manual
Automatically receiving changes from Gitea Repos

Weblate has support for Gitea webhooks, add a Gitea Webhook for Push events event with destination to /hooks/gitea/ URL on your Weblate installation (for example https://hosted.weblate.org/hooks/gitea/). This can be done in Webhooks under repository Settings.

Weblate in Gitea manual [POST /hooks/gitea/](https://hosted.weblate.org/hooks/gitea/)

Automatically receiving changes from Gitee Repos

Weblate has support for Gitee webhooks, add a WebHook for Push event with destination to /hooks/gitee/ URL on your Weblate installation (for example https://hosted.weblate.org/hooks/gitee/). This can be done in WebHooks under repository Management.

Weblate in Gitee manual [POST /hooks/gitee/](https://hosted.weblate.org/hooks/gitee/)

Automatically updating repositories nightly

Weblate automatically fetches remote repositories nightly to improve performance when merging changes later. You can optionally turn this into doing nightly merges as well, by enabling AUTO_UPDATE.

Pushing changes from Weblate

Each translation component can have a push URL set up (see [URL](https://hosted.weblate.org/hooks/gitea/)), and in that case Weblate will be able to push change to the remote repository. Weblate can also be configured to automatically push changes on every commit (this is default, see [AUTO_UPDATE](https://hosted.weblate.org/hooks/gitea/)). If you do not want changes to be pushed automatically, you can do that manually under Repository maintenance or using API via `wlc push`.

The push options differ based on the [URL](https://hosted.weblate.org/hooks/gitea/) used, more details are found in that chapter.

<table>
<thead>
<tr>
<th>Desired setup</th>
<th>URL</th>
<th>push</th>
</tr>
</thead>
<tbody>
<tr>
<td>No push</td>
<td></td>
<td>empty</td>
</tr>
<tr>
<td>Push directly</td>
<td>Git</td>
<td>SSH URL</td>
</tr>
<tr>
<td>GitHub pull request from fork</td>
<td>GitHub</td>
<td>SSH URL</td>
</tr>
<tr>
<td>GitHub pull request from branch</td>
<td>GitHub</td>
<td>SSH URL</td>
</tr>
<tr>
<td>GitLab merge request from fork</td>
<td>GitLab</td>
<td>SSH URL</td>
</tr>
<tr>
<td>GitLab merge request from branch</td>
<td>GitLab</td>
<td>SSH URL</td>
</tr>
<tr>
<td>Pagure pull request from branch</td>
<td>Pagure</td>
<td>SSH URL</td>
</tr>
<tr>
<td>Pagure merge request from branch</td>
<td>Pagure</td>
<td>SSH URL</td>
</tr>
</tbody>
</table>

You can also enable automatic pushing of changes after Weblate commits, this can be done in [Repository maintenance](https://hosted.weblate.org/hooks/gitea/).

See [URL](https://hosted.weblate.org/hooks/gitea/) for setting up SSH keys, and Lazy commits for info about when Weblate decides to commit changes.

Can be empty in case [URL](https://hosted.weblate.org/hooks/gitea/) supports pushing.
Protected branches

If you are using Weblate on protected branch, you can configure it to use pull requests and perform actual review on the translations (what might be problematic for languages you do not know). An alternative approach is to waive this limitation for the Weblate push user.

For example on GitHub this can be done in the repository configuration:

Interacting with others

Weblate makes it easy to interact with others using its API.

- Weblate REST API

Lazy commits

The behaviour of Weblate is to group commits from the same author into one commit if possible. This greatly reduces the number of commits, however you might need to explicitly tell it to do the commits in case you want to get the VCS repository in sync, e.g. for merge (this is by default allowed for the Managers group, see [Managers](https://weblate.org/docs/admin/#managers)).

The changes in this mode are committed once any of the following conditions are fulfilled:

- Somebody else changes an already changed string.
- An explicit commit is requested.
- Change is older than period defined as [on](https://weblate.org/docs/admin/#period) on [Managers](https://weblate.org/docs/admin/#managers).

**Note:** Commits are created for every component. So in case you have many components you will still see lot of commits. You might utilize [Git](https://weblate.org/docs/admin/#git) add-on in that case.

If you want to commit changes more frequently and without checking of age, you can schedule a regular task to perform a commit:
CELERY_BEAT_SCHEDULE = {
    "commit": {
        "task": "weblate.trans.tasks.commit_pending",
        # Unconditionally commit all changes every 2 minutes
        # Commited 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,
    }
}

Processing repository with scripts

The way to customize how Weblate interacts with the repository is [scripts](#). Consult [scripts](#) for info on how to execute external scripts through add-ons.

Keeping translations same across components

Once you have multiple translation components, you might want to ensure that the same strings have same translation. This can be achieved at several levels.

Translation propagation

With [scripts](#) enabled (what is the default, see [scripts](#)), all new translations are automatically done in all components with matching strings. Such translations are properly credited to currently translating user in all components.

**Note**: The translation propagation requires the key to be match for monolingual translation formats, so keep that in mind when creating translation keys.

Consistency check

The [scripts](#) check fires whenever the strings are different. You can utilize this to review such differences manually and choose the right translation.

Automatic translation based on different components can be way to synchronize the translations across components. You can either trigger it manually (see [scripts](#)) or make it run automatically on repository update using add-on (see [scripts](#)).
Your profile

Languages Preferences Notifications Account Profile Licenses Audit log API access

Licenses

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

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

You have agreed to the following as a contributor:

- WeblateOrg/Language names

Licenses for individual translations

GNU General Public License v3.0 or later

WeblateOrg/Djangojs WeblateOrg/Django WeblateOrg/WeblateOrg WeblateOrg/Language names

MIT License

WeblateOrg/Android
Additional info on source strings

Edit additional string info

Explanation

Additional explanation to clarify meaning or usage of the string.

Labels

- Current status
- Next status

Translation labels can be defined in the project settings.

Translation flags

Additional comma-separated flags to influence Weblate behavior.

Save and continue

Save

Save and continue
### Nearby strings

<table>
<thead>
<tr>
<th>Context</th>
<th>English</th>
<th>Czech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files</td>
<td>Souřady</td>
<td>Souřady</td>
</tr>
<tr>
<td>Automatic translation</td>
<td>Automatic překlad</td>
<td>Automatic překlad</td>
</tr>
<tr>
<td>Add new translation string</td>
<td>Add new translation string</td>
<td>Add new translation string</td>
</tr>
<tr>
<td>Translation status</td>
<td>Souřady překladů</td>
<td>Souřady překladů</td>
</tr>
<tr>
<td>%Shorten word</td>
<td>%Souřady slovo</td>
<td>%Souřady slovo</td>
</tr>
<tr>
<td>Other components</td>
<td>Další součástí</td>
<td>Další součástí</td>
</tr>
<tr>
<td>Translation file</td>
<td>Souřady překladů</td>
<td>Souřady překladů</td>
</tr>
<tr>
<td>Download</td>
<td>Souřady překladů</td>
<td>Souřady překladů</td>
</tr>
</tbody>
</table>

**Automatic translation via machine translation engine**: Automatic translation is a service that uses machine translation engines to get the best possible translations and applies them in this project. You can add new translation strings here, which will be automatically applied throughout the application. 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.

**Automatic překlad prostřednictvím strojového překladu**: Automatic překlad je služba, která používá strojové překladové motory k získání nejlepších překladů a aplikacím jejich do projektu. Můžete zde přidat nové řetězce k překladu, které budou automaticky upřesněny ve všech součástech aplikace. Soubor s překladem budou spojen s použitím překladů.

**Translation file**: The uploaded file will be merged with the current translation. The file should contain a list of lines, each containing a key-value pair, where the key is the source text and the value is the translated text. The values should be in the format: `en:source_text;cs:translated_text`.

**Add term to glossary**: You can add a term to the glossary for future use. The term will be automatically translated and added to the project.

**Translation settings**: You can control the translation settings by editing the settings in the project configuration file. The file should contain a list of terms and their translations, each on a separate line.

**Translation file**: You can add new translation strings here, which will be automatically applied throughout the application. 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.

**Automatic translation via machine translation engine**: Automatic translation is a service that uses machine translation engines to get the best possible translations and applies them in this project.
Strings prioritization

2.0

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

This can be used to order the flow of translation in a logical manner.

2.4

3.3: Previously called `Quality checks flags`, it no longer configures only checks.

The string flags are also inherited from the `at` and flags from the translation file (see `at`).

4.1: In previous versions this has been called `Extra context`.

Use the explanation to clarify scope or usage of the translation. You can use Markdown to include links and other markup.

Visual context for strings

2.9

You can upload a screenshot showing a given source string in use within your program. This helps translators understand where it is used, and how it should be translated.
In addition to *Additional info on source strings*, screenshots have a separate management interface under the *Tools* menu. Upload screenshots, assign them to source strings manually, or use optical character recognition to do so.

Once a screenshot is uploaded, this interface handles management and source string association:
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
ICU MessageFormat

ICU MessageFormat
XML

replacements:%s:"John Doe"

variants

AngularJS
C
C#
ECMAScript
i18next
Java
Java MessageFormat
JavaScript
Lua
Object Pascal
Perl
PHP
Python
Python
Qt
Qt
Ruby
Scheme
Vue I18n

Markdown
HTML
URL

Ignore all quality checks.

BBCode
ICU MessageFormat syntax

<table>
<thead>
<tr>
<th>Special Syntax</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ignore-</td>
<td>Ignore the following</td>
</tr>
</tbody>
</table>

Example:

```plaintext
I18N: ignore-*
```

Weblate

- GNU gettext

- TrueType/OpenType

- Weblate

Version 3.11

Version 3.7

Weblate
### Font group

<table>
<thead>
<tr>
<th>Name</th>
<th>default-font</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default font</td>
<td>Source Sans 3 Bold</td>
</tr>
<tr>
<td>Japanese</td>
<td>Droid Sans fallback Regular</td>
</tr>
<tr>
<td>Korean</td>
<td>Droid Sans fallback Regular</td>
</tr>
</tbody>
</table>

#### Add language override

<table>
<thead>
<tr>
<th>Language</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Font</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Edit font group

<table>
<thead>
<tr>
<th>Font group name</th>
<th>default-font</th>
</tr>
</thead>
</table>

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

<table>
<thead>
<tr>
<th>Default font</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Sans 3 Bold</td>
<td></td>
</tr>
</tbody>
</table>

Default font is used unless per language override matches.

[Save]
Weblate 

Font groups

<table>
<thead>
<tr>
<th>Group name</th>
<th>Default font</th>
<th>Language overrides</th>
</tr>
</thead>
<tbody>
<tr>
<td>default-font</td>
<td>Source Sans 3 Bold</td>
<td>Japanese: Droid Sans Fallback Regular, Korean: Droid Sans Fallback Regular</td>
</tr>
</tbody>
</table>

Add font group

Font group name

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

-------

Default font is used unless per language override matches.

Save

---

Font

Font family: Droid Sans Fallback

Font style: Regular

File size: 3939052

Created: now

Uploaded by: [User]

Used in groups

Delete

---

Weblate
1. `weblate.checks.Check`
2. `check`
3. `check_single`

```python
# Copyright © 2012–2022 Michal Čihař <michal@cihar.com>
# This file is part of Weblate <https://weblate.org/>
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
```
```python
# """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


# """Quality check example for Czech plurals."""

from django.utils.translation import gettext_lazy as_
from weblate.checks.base import TargetCheck

class PluralCzechCheck(TargetCheck):
    # Used as identifier for check, should be unique
    # Has to be shorter than 50 characters
    check_id = "foo"

    # Short name used to display failing check
    name = _("Foo check")

    # Description for failing check
    description = _("Your translation is foo")
```

---

247
# Real check code

```python
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
```

## MT_SERVICES

**amaGama**

```python
several
weblate.machinery.tmserver.AmagamaTranslation
```

### Installing amaGama

**Apertium**

```python
several
weblate.machinery.apertium.ApertiumAPYTranslation
```

**AWS**

```python
several
```

### Installing AWS

**MT_AWS_REGION**
**MT_AWS_ACCESS_KEY_ID**
**MT_AWS_SECRET_ACCESS_KEY**

---

248
Baidu API machine translation

**3.2 Baidu**

This service uses an API and you need to obtain an ID and API key from Baidu to use it.

Turn on this service by adding `weblate.machinery.baidu.BaiduTranslation` to `MT_SERVICES` and set `MT_BAIDU_ID` and `MT_BAIDU_SECRET`.

---

**DeepL**

**2.20 DeepL**

DeepL is a paid service providing good machine translation for a few languages. You need to purchase `DeepL API` subscription or use the legacy `DeepL Pro (classic)` plan.

Turn on this service by adding `weblate.machinery.deepl.DeepLTranslation` to `MT_SERVICES` and set `MT_DEEPL_KEY`.

---

In case you have subscription for CAT tools, you are supposed to use "v1 API" instead of default "v2" used by Weblate (it is not really an API version in this case). In case you are on a free instead of a paid plan, you have to use `https://api-free.deepl.com/` instead of `https://api.deepl.com/` You can adjust both parameters by `MT_DEEPL_API_URL`.

---

**LibreTranslate**

**4.7.1 LibreTranslate**

LibreTranslate is a free and open-source service for machine translations. The public instance requires an API key, but LibreTranslate can be self-hosted and there are several mirrors available to use the API for free.

Turn on this service by adding `weblate.machinery.libretranslate.LibreTranslateTranslation` to `MT_SERVICES` and set `MT_LIBRETRANSLATE_API_URL` and `MT_LIBRETRANSLATE_KEY`.

---

**Glosbe**

Free dictionary and translation memory for almost every living language.

The API is gratis to use, but subject to the used data source license. There is a limit of calls that may be done from one IP in a set period of time, to prevent abuse.

Turn on this service by adding `weblate.machinery.glosbe.GlosbeTranslation` to `MT_SERVICES`.

---

Glosbe website
Google Translate

This service uses the Google Translation API, and you need to obtain an API key and turn on billing in the Google API console.

To turn on this service, add `weblate.machinery.google.GoogleTranslation` to `MT_SERVICES` and set `MT_GOOGLE_KEY`.

**mt_google_key**

Google translate documentation

Google Translate API V3 (Advanced)

This service differs from the former one in how it authenticates. To enable service, add `weblate.machinery.googlev3.GoogleV3Translation` to `MT_SERVICES` and set `MT_GOOGLE_CREDENTIALS`. 

If `location` fails, you may also need to specify `MT_GOOGLE_LOCATION`.

**mt_google_credential**

**mt_google_project**

**mt_google_location**

Microsoft Cognitive Services Translator


To enable this service, add `weblate.machinery.microsoft.MicrosoftCognitiveTranslation` to `MT_SERVICES` and set `MT_MICROSOFT_COGNITIVE_KEY`.

Translator Text API V2

The key you use with Translator API V2 can be used with API 3.

Translator Text API V3

The key you use with Translator API V2 can be used with API 3.

Microsoft Terminology Service

The Microsoft Terminology Service API Web. Weblate implements a service that uses the Microsoft Terminology Service API Web.

250
ModernMT

**Version 4.2**

Turn this service on by adding `weblate.machinery.modernmt.ModernMTTranslation` to `MT_SERVICES` and configure `MT_MODERNMT_KEY`.

**Requirements:**
- ModernMT API, `MT_MODERNMT_KEY`, `MT_MODERNMT_URL`

MyMemory

Huge translation memory with machine translation.

Free, anonymous usage is currently limited to 100 requests/day, or to 1000 requests/day when you provide a contact e-mail address in `MT_MYMEMORY_EMAIL`. You can also ask them for more.

Turn on this service by adding `weblate.machinery.mymemory.MyMemoryTranslation` to `MT_SERVICES` and set `MT_MYMEMORY_EMAIL`.

**Requirements:**
- `MT_MYMEMORY_EMAIL`
- `MT_MYMEMORY_USER`
- `MT_MYMEMORY_KEY`
- MyMemory website

NetEase Sight API machine translation

**Version 3.3**

NetEase [NetEase Sight](https://www.netease.com)

This service uses an API, and you need to obtain key and secret from NetEase.

Turn on this service by adding `weblate.machinery.youdao.NeteaseSightTranslation` to `MT_SERVICES` and set `MT_NETEASE_KEY` and `MT_NETEASE_SECRET`.

**Requirements:**
- `MT_NETEASE_KEY`
- `MT_NETEASE_SECRET`
- NetEase Sight Translation Platform

tmserver

You can run your own translation memory server by using the one bundled with Translate-toolkit and let Weblate talk to it. You can also use it with an amaGama server, which is an enhanced version of tmserver.

1. First you will want to import some data to the translation memory:

   ```
   build_tmdb -d /var/lib/tm/db -s en -t cs locale/cs/LC_MESSAGES/django.po
   build_tmdb -d /var/lib/tm/db -s en -t de locale/de/LC_MESSAGES/django.po
   build_tmdb -d /var/lib/tm/db -s en -t fr locale/fr/LC_MESSAGES/django.po
   ```

2. Turn on this service by adding `weblate.machinery.tmserver.TMServerTranslation` to `MT_SERVICES`.

3. Start tmserver to listen to your requests:

   ```
   tmserver -d /var/lib/tm/db
   ```

4. Configure Weblate to talk to it:

   ```
   MT_TMSERVER = "http://localhost:8888/tmserver/"
   ```

**Requirements:**
- `MT_TMSERVER`
- tmserver

Installing amaGama

Amagama [Amagama Translation Memory](https://www.amagama.com)
Yandex Translate

This service uses a Translation API, and you need to obtain an API key from Yandex.

Turn on this service by adding `weblate.machinery.yandex.YandexTranslation` to `MT_SERVICES`, and set `MT_YANDEX_KEY`.

MT_YANDEX_KEY

Yandex Translate API

Youdao Zhiyun API machine translation

This service uses an API, and you need to obtain an ID and an API key from Youdao.

Turn on this service by adding `weblate.machinery.youdao.YoudaoTranslation` to `MT_SERVICES` and set `MT_YOUDAO_ID` and `MT_YOUDAO_SECRET`.

MT_YOUDAO_ID MT_YOUDAO_SECRET

Youdao Zhiyun Natural Language Translation Service

Weblate

Weblate can be the source of machine translations as well. It is based on the Woosh fulltext engine, and provides both exact and inexact matches.

Turn on these services by adding `weblate.machinery.weblatetm.WeblateTranslation` to `MT_SERVICES`.

Weblate Translation Memory

The can be used as a source for machine translation suggestions as well.

Turn on these services by adding `weblate.memory.machine.WeblateMemory` to the `MT_SERVICES`. This service is turned on by default.

SAP Translation Hub

You need to have a SAP account (and the SAP Translation Hub enabled in the SAP Cloud Platform) to use this service.

`weblate.machinery.saptranslationhub.SAPTranslationHub` to `MT_SERVICES`.

MT_SERVICES

MT_SAP_BASE_URL MT_SAP_SANDBOX_APIKEY MT_SAP_USERNAME MT_SAP_PASSWORD MT_SAP_USE_MT

SAP Translation Hub API

252
Custom machine translation

You can also implement your own machine translation services using a few lines of Python code. This example implements machine translation in a fixed list of languages using the `dictionary` Python module:

```python
import dictionary

from weblate.machinery.base import MachineTranslation

class SampleTranslation(MachineTranslation):
    """Sample machine translation interface."""

    name = "Sample"

def download_languages(self):
    """Return list of languages your machine translation supports."""
    return {"cs"}

def download_translations(self, source, language, text: str, unit, user, search: bool, threshold: int = 75,):
    """Return tuple with translations."""
    for t in dictionary.translate(text):
        yield {"text": t, "quality": 100, "service": self.name, "source": text}
```

You can list your own class in `MT.Services` and Weblate will start using that.
### Installed add-ons

There are no add-ons currently installed.

### Available add-ons

- **Automatic translation**
  - Automatically translates strings using machine translation or other components.
  - ![Install](Install)

- **Add missing languages**
  - Ensures a consistent set of languages is used for all components within a project.
  - ![project-wide](project-wide)
  - ![Install](Install)

- **Component discovery**
  - Automatically adds or removes project components based on file changes in the version control system.
  - ![repository-wide](repository-wide)
  - ![Install](Install)

- **Bulk edit**
  - Bulk edit flags, labels, or states of strings.
  - ![Install](Install)

- **Statistics generator**
  - Generates a file containing detailed info about the translation status.
  - ![Install](Install)

- **Prefill translation with source**
  - Fills in translation strings with source string.
  - ![Install](Install)

- **Pseudolocale generation**
  - Generates a translation by adding prefix and suffix to source strings automatically.
  - ![Install](Install)

- **Contributors in comment**
  - Updates the comment part of the PO file header to include contributor names and years of contributions.
  - ![Install](Install)

- **Customize gettext output**
  - Allows customization of gettext output behavior, for example line wrapping.
  - ![Install](Install)

- **Generate MO files**
  - Automatically generates a MO file for every changed PO file.
  - ![Install](Install)

- **Update PO files to match POT (msgmerge)**
  - Updates all PO files (as configured by "File mask") to match the POT file (as configured by "Template for new translations") using msgmerge.
  - ![Install](Install)

- **Squash Git commits**
  - Squashes Git commits prior to pushing changes.
  - ![repository-wide](repository-wide)
  - ![Install](Install)

- **Stale comment removal**
  - Set a timeframe for removal of comments.
  - ![project-wide](project-wide)
  - ![Install](Install)

- **Stale suggestion removal**
  - Set a timeframe for removal of suggestions.
  - ![project-wide](project-wide)
  - ![Install](Install)

Some add-ons will ask for additional configuration during installation.
# 3.9 CLI

weblate.autotranslate.autotranslate

mode: 
- suggest: 
- translate: 
- fuzzy:

filter_type:
- all:
- nottranslated:
- todo:
- fuzzy:
- check:inconsistent:

auto:
- others:
- mt:

component:
- en:
- de:

threshold:

css_selector:

cookie_name:

files:

JavaScript CDN

# 4.2 CLI

weblate.cdn.cdnjs

daily post-commit post-update

JavaScript HTML CDN

HTML/JavaScript CDN

CDN

cdn-addon-config cdn-addon-extract cdn-addon-html
4.4 Weblate.
weblate.cleanup.blank
post-commit post-update

Weblate

weblate.cleanup.generic
pre-commit post-update

Weblate

weblate.consistency.languages
daily post-add

Weblate

weblate.discovery.discovery

match
file_format
name_template
base_file_template
ew_base_template
name_template
language_regex
copy_addons
remove
confirm

post-update

VCS import_project

VCS
Please review and confirm the matched components.

Component | Matched files
---|---
Django | web\site\locale\he\LC_MESSAGES\django.po (he)
web\site\locale\cs\LC_MESSAGES\django.po (cs)
web\site\locale\hu\LC_MESSAGES\django.po (hu)

Django | web\site\locale\he\LC_MESSAGES\django.po (he)
web\site\locale\cs\LC_MESSAGES\django.po (cs)

I confirm the above matches look correct.

Regular expression to match translation files against

website/locale/{language}/[\^T]*LC_MESSAGES/{component}[/\^T]*.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

website/locale/[component][poe]

Filename of file used for creating new translations. For gettext choose .po file.

Language filter

\^\*\*\*\*\*\*\*\*

Regular expression to filter translation files against when scanning for file mask

[[ component ]]

Remove components for inexistent files

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

<table>
<thead>
<tr>
<th>Regular expression</th>
<th>Example matched files</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{language}/[^/]+/[^/]+</td>
<td>cs/application.po cs/website.po de/application.po de/website.po</td>
<td>One folder per language containing translation files for components.</td>
</tr>
<tr>
<td>local\site\LC_MESSAGES\component[^/]+.po</td>
<td>local\site\LC_MESSAGES\application.po local\site\LC_MESSAGES\website.po</td>
<td>Short structure for storing gettext PO files.</td>
</tr>
<tr>
<td>src\locale[^/]+[/^T]*([^/]+)[/^T]+</td>
<td>src\locale\application.cs.po src\locale\website.cs.po</td>
<td>Using both component and language name within filename.</td>
</tr>
<tr>
<td>src\locale[^/]+[/^T]*([^/]+)[/^T]+</td>
<td>src\locale\application.de.po src\locale\website.de.po</td>
<td>Using language in both path and filename.</td>
</tr>
<tr>
<td>res/values-{language}/[^/]+[/^T]*strings-[^/]+[/^T]+.xml</td>
<td>res/values(cs)/strings-about.xml res/values(cs)/strings-help.xml res/values(de)/strings-about.xml res/values(de)/strings-help.xml</td>
<td>Android resource strings, split into several files.</td>
</tr>
</tbody>
</table>

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 uppercase case first letter
weblate.flags.bulk

```
q
state
-1
10
20
30
add_flags
remove_flags
add_labels
remove_labels
```

update

```
NOT has:label
```

```
Table6

NOT has:label
```

```
Table7
language:en AND key:changelogs/read-only
```

```
labels
```

```
```

weblate.flags.same_edit

post-create

VCS
weblate.flags.source_edit
post-create
VCS Weblate

weblate.flags.target_edit
post-create
VCS Weblate

weblate.generate.generate

filename
template

pre-commit

Django:
locale/{{ language_code }}.json

{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}"
}

prefill
updatedaily

4.11
weblate 4.5

weblate.generate.pseudolocale

source

target

prefix

var_prefix

suffix

var_suffix

var_multiplier

All strings in this translation will be overwritten

Fixed string prefix

Variable string prefix

Fixed string suffix

Variable string suffix

Variable part multiplier - How many times to repeat the variable part depending on the length of the source string.

This makes it possible to look for strings which might not fit into the user interface after the localization - it extends the text based on the source string length. The variable parts are repeated by length of the text multiplied by the multiplier. For example Hello world with variable suffix _ and variable multiplier of 1 becomes Hello world___________ - the suffix is repeated once for each character in the source string.

The strings will be generated using following pattern:

Fixed string prefix Variable string prefix Source string Variable string suffix Fixed string suffix

weblate.gettext.authors

pre-commit

PO:

# Pavel Borecki <pavel@example.com>, 2018, 2019.
# Filip Hron <filip@example.com>, 2018, 2019.
# anonymous <noreply@weblate.org>, 2019.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure ALL_LINGUAS</td>
<td>Configure localization support for all languages.</td>
</tr>
<tr>
<td>weblate.gettext.configure</td>
<td>Configure gettext settings.</td>
</tr>
<tr>
<td>post-add daily</td>
<td>Add files to the daily build.</td>
</tr>
<tr>
<td>configure configure.in configure.ac configure.ac</td>
<td>Configuregettext customization.</td>
</tr>
<tr>
<td>gettext</td>
<td>Execute gettext tasks.</td>
</tr>
<tr>
<td>post-load gettext</td>
<td>Load translations from the database.</td>
</tr>
<tr>
<td>subclass</td>
<td>Subclassifications are handled.</td>
</tr>
<tr>
<td>gettext</td>
<td>Execute gettext tasks.</td>
</tr>
<tr>
<td>post-commit PO MO</td>
<td>Commit messages from PO to MO files.</td>
</tr>
<tr>
<td>pre-commit</td>
<td>Pre-commit messages from PO to MO files.</td>
</tr>
<tr>
<td>PO MO</td>
<td>Translate messages from PO files to MO files.</td>
</tr>
</tbody>
</table>

**LINGUAS**

Configure localization support for specific languages.

**MO**

Configure MO localization files.

**MO path**

Set the path for MO files.

**pre-commit**

Commit messages from PO to MO files.
POT PO (msgmerge)

weblate.gettext.msgmerge

previous no_location fuzzy

post-update

msgmerge POT PO

post-commit

Git

weblate.git.squash

squash all language file author

ap- pend_trail RFC 822 "Co-authored-by: ..."

commit_message

JSON

weblate.json.customize

sort_keys indent style

post-load

264
Java 3.7

weblate.properties.sort
pre-commit
Java 3.7

weblate.removal.comments

daily

weblate.removal.suggestions

daily

RESX 3.9

weblate.resx.update
post-update

Weblate
YAML

# Copyright © 2012–2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""Example pre commit script."""

from django.utils.translation import gettext_lazy as _
from weblate.addons.events import EVENT_PRE_COMMIT
from weblate.addons.scripts import BaseScriptAddon
```python
class ExamplePreAddon(BaseScriptAddon):
    # Event used to trigger the script
    events = (EVENT_PRE_COMMIT,)
    # Name of the addon, has to be unique
    name = "weblate.example.pre"
    # Verbose name and long description
    verbose = _("Execute script before commit")
    description = _("This add-on executes a script.")

    # Script to execute
    script = "/bin/true"
    # File to add in commit (for pre commit event)
    # does not have to be set
    add_file = "po/{{ language_code }}.po"
```

---

**VCS**: WL_VCS

**Repository**: URL

**Path**: WL_PATH

**Branch**: WL_BRANCH

**Filemask**: WL_FILEMASK

**Template**: WL_TEMPLATE

**New base**: WL_NEW_BASE

**File format**: WL_FILE_FORMAT

**Language**: WL_LANGUAGE

**Previous head**: WL_PREVIOUS_HEAD

**Component slug**: WL_COMPONENT_SLUG

**Project slug**: WL_PROJECT_SLUG

**Component name**: WL_COMPONENT_NAME
VCS   Weblate  VCS

Gulp:

```bash
#!/bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

2.20

Weblate comes with a built-in translation memory consisting of the following:

Manually imported translation memory (see User interface).

Automatically stored translations performed in Weblate (depending on Translation memory scopes).

Automatically imported past translations.

Content in the translation memory can be applied one of two ways:

Manually, view while translating.

For installation tips, see Weblate Translation Memory, which is turned on by default.

Translation memory scopes

3.2: In earlier versions translation memory could be only loaded from a file corresponding to the current imported translation memory scope.

The translation memory scopes are there to allow both privacy and sharing of translations, to suit the desired behavior.
Imported translation memory

Importing arbitrary translation memory data using the `import_memory` command makes memory content available to all users and projects.

Per user translation memory

Stores all user translations automatically in the personal translation memory of each respective user.

Per project translation memory

All translations within a project are automatically stored in a project translation memory only available for this project.

All translation within projects with shared translation memory turned on are stored in a shared translation memory available to all projects.

Please consider carefully whether to turn this feature on for shared Weblate installations, as it can have severe implications:

The translations can be used by anybody else.
This might lead to disclosing secret information.

Managing translation memory

User interface

In the basic user interface you can manage per user and per project translation memories. It can be used to download, wipe or import translation memory.

Translation memory in JSON can be imported into Weblate, TMX is provided for interoperability with other tools.
There are several management commands to manipulate the translation memory content. These operate on the translation memory as whole, unfiltered by scopes (unless requested by parameters):

Exports the memory into JSON
Imports TMX or JSON files into the translation memory

```python
settings.py
```

Django, Weblate, WSGI, Apache, Celery, mod_wsgi, Django's documentation

**AKISMET_API_KEY**

Webblet, Akismet, Weblate documentation, API, akismet.com

**ANONYMOUS_USER_NAME**

Webblet, documentation

**AUDITLOG_EXPIRY**

Weblet documentation, 180 days

**AUTH_LOCK_ATTEMPTS**

Django, 10 attempts
AUTO_UPDATE

3.2
3.11 on/off

Weblate

3.1 on/off

Weblate

Celery

AVATAR_URL_PREFIX

URL: ${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}

AVATAR_URL_PREFIX = 'https://www.gravatar.com/
AVATAR_URL_PREFIX = 'https://www.libravatar.org/

ENABLE_AVATARS

AUTH_TOKEN_VALID

2.14 1728002

AUTH_PASSWORD_DAYS

2.15

AUTH_PASSWORD_DAYS 180

AUTOFIX_LIST

autofixer Python

...
AUTOFIX_LIST =
    [
      "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
      "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
    ]

BACKGROUND_TASKS

4.5.2

BASE_DIR

Weblate

DATA_DIR

BASIC_LANGUAGES

4.4

BASIC_LANGUAGES = 
    {"cs", "it", "ja", "en"}
**BORG_EXTRA_ARGS**

```
# 4.9
borg create
```

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/*"]
```

```
Weblate borg create
```

**CSP_SCRIPT_SRC**

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

**CHECK_LIST**

```
# check Python
```

```
CHECK_LIST = ()
```

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

```
updatechecks
```
COMMENT_CLEANUP_DAYS

3.6

COMMIT_PENDING_HOURS

2.10

CONTACT_FORM

4.6

Configures how e-mail from the contact form is being sent. Choose a configuration that matches your mail server configuration.

The sender is used in as Reply-To, this is the default behaviour.

The sender is used in as From. Your mail server needs to allow sending such e-mails.

DATA_DIR

Weblate

VCS

SSH

STATIC_ROOT

The Docker container uses a separate volume for this, see Docker container volumes.

MEDIA_ROOT

Django

Visual context for strings

Celery

User-uploaded fonts, see...

www-data

sudo chown www-data:www-data -R $DATA_DIR

BASE_DIR

Django:

www-data

sudo chown www-data:www-data -R $DATA_DIR
DATABASE_BACKUP

3.1

"plain"
"compressed"
"none"

Weblate

DEFAULT_ACCESS_CONTROL

3.3

ACL

Weblate

DEFAULT_AUTO_WATCH

4.5

Configures whether Automatically watch projects on contribution should be turned on for new users. Defaults to True.

DEFAULT_RESTRICTED_COMPONENT

4.1


DEFAULT_ADD_MESSAGE

DEFAULT_ADDON_MESSAGE

DEFAULT_COMMIT_MESSAGE

DEFAULT_DELETE_MESSAGE

DEFAULT_MERGE_MESSAGE
DEFAULT_ADDONS

```
DEFAULT_ADDONS = {
    # Add-on with no parameters
    "weblate.flags.target_edit": {},
    # Add-on with parameters
    "weblate.autotranslate.autotranslate": {
        "mode": "suggest",
        "filter_type": "todo",
        "auto_source": "mt",
        "component": "",
        "engines": ["weblate-translation-memory"],
        "threshold": "80",
    },
}
```

install_addon WEBLATE_ADDONS

DEFAULT_COMMITER_EMAIL

```
2.4.0
noreply@weblate.org
```

DEFAULT_COMMITER_NAME

```
2.4.0
Weblate
```

DEFAULT_COMMITER_EMAIL

DEFAULT_LANGUAGE

```
4.3.2
```

```
DEFAULT_MERGE_STYLE

rebase - merge

DEFAULT_SHARED_TM

Configures default value of and .

DEFAULT_TRANSLATION_PROPAGATION

True:

DEFAULT_PULL_MESSAGE

ENABLE_AVATARS

Gravatar - AVATAR_URL_PREFIX

ENABLE_HOOKS

ENABLE_HTTPS

Weblate HTTPS HTTP URL

HTTPS Django - Cookie HSTS

HTTPS URL

HTTPS Django SSL

HTTPS Django - For-

warded - Django SSL SE-

CURE_PROXY_SSL_HEADER

SESSION_COOKIE_SECURE CSRF_COOKIE_SECURE SECURE_SSL_REDIRECT SECURE_PROXY_SSL_HEADER
**ENABLE_SHARING**

Enable/Disable sharing on/off

**GET_HELP_URL**

4.5.2.

Weblate [GET_HELP_URL](#).

**GITEA_CREDENTIALS**

4.12.

List for credentials for Gitea servers.

*Note:* Use this in case you want Weblate to interact with more of them, for single Gitea endpoint stick with `GITEA_USERNAME` and `GITEA_TOKEN`.

```python
GITEA_CREDENTIALS = {
    "try.gitea.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitea.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

**GITEA_USERNAME**

4.12.

Gitea username used to send pull requests for translation updates.

*Note:* `GITEA_CREDENTIALS`, Gitea pull requests

**GITEA_TOKEN**

4.12.

Gitea personal access token used to make API calls to send pull requests for translation updates.

*Note:* `GITEA_CREDENTIALS`, Gitea pull requests, Creating a Gitea personal access token

**GITLAB_CREDENTIALS**

4.3.

GitLab credentials.

*Note:* Weblate [GET_HELP_URL](#) GitLab credentials, GitLab Username and Token.
GITLAB_CREDENTIALS = {
    "gitlab.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitlab.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}

GITLAB_USERNAME

GitLab

GITLAB_TOKEN

4.3

API

GitLab

GITLAB_CREDENTIALS

GitLab

GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "github.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}

GITHUB_USERNAME

GitHub

GITHUB_TOKEN

4.3

GitHub

GITHUB_CREDENTIALS

GitHub
**GITHUB_TOKEN**

GitHub API

**GITHUB_CREDENTIALS**

GitHub

**GOOGLE_ANALYTICS_ID**

Google ID

**HIDE_REPO_CREDENTIALS**

VCS

URL

```
https://user:password@git.example.com/repo.git
https://git.example.com/repo.git
```

**IP_BEHIND_REVERSE_PROXY**

IP PROXY HEADER

True

**HIDE_VERSION**

4.3.1

**IP_PROXY_HEADER**

IP PROXY OFFSET

280
**IP_PROXY_HEADER**

```
Weblate  IP
HTTP_X_FORWARDED_FOR: a, b, client-ip
```

**IP_PROXY_OFFSET**

```
Weblate  IP
SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY
```

**LEGAL_URL**

```
Weblate  URL
LEGAL_URL = "https://weblate.org/terms/"
```

**LICENSE_EXTRA**

```
LICENSE_EXTRA = [
    "AGPL-3.0",
    "GNU Affero General Public License v3.0",
    "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
];
```
4.3: LICENSE_FILTER

```python
LICENSE_FILTER = ("AGPL-3.0", "GPL-3.0-or-later")
```

```python
LICENSE_FILTER = set()
```

alerts

4.4: LICENSE_REQUIRED

5. LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

```python
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = 10
```

```python
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = False
```

```
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True
```

6. LOCALIZE_CDN_URL

```javascript
LOCALIZE_CDN_URL = "https://weblate-cdn.com/
```

```
LOCALIZE_CDN_PATH = Weblate CDN URL
```

7. LOGIN_REQUIRED_URLS

```python
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
```

```python
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```
REQUIRE_LOGIN

LOGIN_REQUIRED_URLS_EXCEPTIONS

LOGIN_REQUIRED_URLS

```
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
}
```

MATOMO_SITE_ID

Matomo Piwik ID:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

MT_SERVICES

API MACHINE_TRANSLATION_SERVICES

When using Docker container, this configuration is automatically generated based on provided API keys, see Machine translation settings.
MT_SERVICES = (  
    "weblate.machinery.apertium.ApertiumAPYTranslation",  
    "weblate.machinery.deepl.DeepLTranslation",  
    "weblate.machinery.glosbe.GlosbeTranslation",  
    "weblate.machinery.google.GoogleTranslation",  
    "weblate.machinery.libretranslate.LibreTranslateTranslation",  
    "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",  
    "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",  
    "weblate.machinery.tmserver.AmagamaTranslation",  
    "weblate.machinery.tmserver.TMServerTranslation",  
    "weblate.machinery.yandex.YandexTranslation",  
    "weblate.machinery.saptranslationhub.SAPTranslationHub",  
    "weblate.memory.machine.WeblateMemory",  
)  

MT_APERTIUM_APY

Apertium-APy [API URL] https://wiki.apertium.org/wiki/Apertium-apy

MT_AWS_ACCESS_KEY_ID

Amazon [API ID]

MT_AWS_SECRET_ACCESS_KEY

AWS [API KEY]

MT_AWS_REGION

Amazon [URL]

MT_BAIDU_ID

Baidu Zhiyun API [API ID] https://api.fanyi.baidu.com/api/trans/product/index

Baidu API machine translation [API ID]
**MT_BAIDU_SECRET**

Baidu Zhiyun API [Documentation](https://api.fanyi.baidu.com/api/trans/product/index)

**Baidu API machine translation**

**MT_DEEPL_API_URL**

4.7.1 API URL

DeepL API URL v1 API v2 API v3 API v4 API v5 API v6 API v7 API v8

CAT API

DeepL API URL


**MT_DEEPL_KEY**

DeepL API key for the LibreTranslate instance specified in MT_LIBRETRANSLATE_API_URL.

**MT_LIBRETRANSLATE_API_URL**

4.7.1 API URL

LibreTranslate API URL

Web API

MIRRORS are documented on the LibreTranslate GitHub repository, some of which can be used without authentication:

https://github.com/LibreTranslate/LibreTranslate#user-content-mirrors

**MT_LIBRETRANSLATE_KEY**

4.7.1 API URL

API key for the LibreTranslate instance specified in MT_LIBRETRANSLATE_API_URL.
MT_GOOGLE_KEY

Google Translate API v2 register at https://cloud.google.com/translate/docs

MT_GOOGLE_CREDENTIALS

Google API v3 JSON OS https://cloud.google.com/docs/authentication/getting-started

MT_GOOGLE_PROJECT

Google API v3 ID https://cloud.google.com/appengine/docs/standard/nodejs/building-app/creating-project

MT_GOOGLE_LOCATION

API v3 Google global https://cloud.google.com/appengine/docs/locations

MT_MICROSOFT_BASE_URL

"Base URLs" Azure api.cognitive.microsofttranslator.com
Azure China api.translator.azure.cn

MT_MICROSOFT_COGNITIVE_KEY

Microsoft Cognitive Services Translator API

MT_MICROSOFT_REGION

"The region" Azure

MT_MICROSOFT_ENDPOINT_URL

"URL" Azure api.cognitive.microsoft.com
Azure China Azure
**MT_MODERNMT_KEY**

ModernMT API key

**MT_MODERNMT_URL**

ModernMT URL: https://api.modernmt.com/

**MT_MYMEMORY_EMAIL**

MyMemory: API technical specifications

**MT_MYMEMORY_KEY**

MyMemory: API key generator

**MT_MYMEMORY_USER**

MyMemory: API key generator

**MT_NETEASE_KEY**

NetEase Sight API key: https://sight.youdao.com/

**MT_NETEASE_SECRET**

NetEase Sight API secret: https://sight.youdao.com/
**MT_TMSERVER**

tmservice URL:

**MT_YANDEX_KEY**

Yandex Translate API API key: [https://yandex.com/dev/translate/](https://yandex.com/dev/translate/)

**MT_YOUDAO_ID**


**MT_YOUDAO_SECRET**


**MT_SAP_BASE_URL**

SAP Translation Hub API URL:

**MT_SAP_SANDBOX_APIKEY**

SAP Translation Hub API key

**MT_SAP_USERNAME**

SAP Username
MT_SAP_PASSWORD

SAP: True

MT_SAP_USE_MT

True or False

NEARBY_MESSAGES

DEFAULT_PAGE_LIMIT

4.7

PAGURE_CREDENTIALS

Pagure

PAGURE CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}

PAGURE_USERNAME

Pagure

PAGURE CREDENTIALS

Pagure
PAGURE_TOKEN

4.3.2 Pagure API

PAGURE_CREDENTIALS

PRIVACY_URL

4.8.1 Weblate

Weblate PRIVACY_URL

PRIVACY_URL = "https://weblate.org/terms/"

LEGAL_URL

RATELIMIT_ATTEMPTS

3.2 Pagure

RATELIMIT_WINDOW

300 600

RATELIMIT_LOCKOUT

600 10

RATELIMIT_ATTEMPTS

RATELIMIT_WINDOW

RATELIMIT_LOCKOUT

RATELIMIT_ATTEMPTS

RATELIMIT_WINDOW
REGISTRATION_ALLOW_BACKENDS

```python
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

REGISTRATION_OPEN

```python
REGISTRATION_OPEN = True
```

REGISTRATION_CAPTCHA

```python
REGISTRATION_CAPTCHA = True
```

REGISTRATION_EMAIL_MATCH

```python
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

REGISTRATION_OPEN

```python
REGISTRATION_OPEN = True
```
REPOSITORY_ALERT_THRESHOLD

REPOSITORY_ALERT_THRESHOLD 4.0.2
alerts

REQUIRE_LOGIN

REQUIRE_LOGIN 4.1

SENTRY_DSN

SENTRY_DSN 3.9

SESSION_COOKIE_AGE_AUTHENTICATED

SESSION_COOKIE_AGE_AUTHENTICATED 4.3

SIMPLIFY_LANGUAGES

SIMPLIFY_LANGUAGES

SITE_DOMAIN

SITE_DOMAIN

# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
WEBLATE_ALLOWED_HOSTS

HTTPS WEBLATE_SITE_DOMAIN ENABLE_HTTPS

SITE_TITLE

Web

SPECIAL_CHARS

SPECIAL_CHARS = ("\t", "\n", "\u00a0", "...")

SINGLE_PROJECT

3.8 True

3.11:

SINGLE_PROJECT = "test"

SSH_EXTRA_ARGS

4.9

Allows to add custom parameters when Weblate is invoking SSH. This is useful when connecting to servers using legacy encryption or other non-standard features.

For example when SSH connection in Weblate fails with Unable to negotiate with legacyhost: no matching key exchange method found. Their offer: diffie-hellman-group1-sha1, you can enable that using:

SSH_EXTRA_ARGS = "-oKexAlgorithms=+diffie-hellman-group1-sha1"

STATUS_URL

Webblate URL
**SUGGESTION_CLEANUP_DAYS**

- 3.2.1 None

**UPDATE_LANGUAGES**

- 4.3.2 None

**URL_PREFIX**

- Weblate

```python
WSGI
WSGIScriptAlias /
URL_PREFIX = "/translations"
```

**VCS_BACKENDS**

- Weblate

```python
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

**VCS_CLONE_DEPTH**

- 3.10.2 None

Weblate

```python
fatal: protocol error: expected old/new/ref, got 'shallow <commit hash>'
```
VCS_CLONE_DEPTH = 0

WEBLATE_ADDONS

WEBLATE_ADDONS = {
    # Built-in add-ons
    "weblate.addons.gettext.GenerateMoAddon",
    "weblate.addons.gettext.UpdateLinguasAddon",
    "weblate.addons.gettext.UpdateConfigureAddon",
    "weblate.addons.gettext.MsgmergeAddon",
    "weblate.addons.gettext.GettextCustomizeAddon",
    "weblate.addons.gettext.GettextAuthorComments",
    "weblate.addons.cleanup.CleanupAddon",
    "weblate.addons.consistency.LanguageConsistencyAddon",
    "weblate.addons.discovery.DiscoveryAddon",
    "weblate.addons.flags.SourceEditAddon",
    "weblate.addons.flags.TargetEditAddon",
    "weblate.addons.flags.SameEditAddon",
    "weblate.addons.flags.BulkEditAddon",
    "weblate.addons.generate.GenerateFileAddon",
    "weblate.addons.json.JSONCustomizeAddon",
    "weblate.addons.properties.PropertiesSortAddon",
    "weblate.addons.removal.RemoveComments",
    "weblate.addons.removal.RemoveSuggestions",
    "weblate.addons.yaml.YAMLCustomizeAddon",
    "weblate.addons.git.GitSquashAddon",
    "weblate.addons.removal.RemoveSuggestions",
    "weblate.addons.removal.RemoveComments",
    "weblate.addons.resx.ResxUpdateAddon",
    "weblate.addons.autotranslate.AutoTranslateAddon",
    "weblate.addons.yaml.YAMLCustomizeAddon",
    "weblate.addons.cdn.CDNJSAddon",
    # Add-on you want to include
    "weblate.addons.example.ExampleAddon",
}

WEBLATE_EXPORTERS

WEBLATE_FORMATS

295
WEBLATE_GPG_IDENTITY

Weblate Git GPG IDENTITY:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

Weblate GPG DATA_DIR is home/.gnupg

GnuPG DATA_DIR is .gnupg

WEBSITE_REQUIRED

Defines whether Web has to be specified when creating a project. Turned on by default as that suits public server setups.

```
import os
import platform
from logging.handlers import SysLogHandler

# Title of site to use
SITE_TITLE = "Weblate"

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

# Django settings for Weblate project.

DEBUG = True

ADMINS = {
    # ("Your Name", "your_email@example.com"),
}
```
MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL, use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password.
        "PASSWORD": "",
        # Set to empty string for localhost.
        "HOST": "127.0.0.1",
        # Set to empty string for default.
        "PORT": "",
        # Customizations for databases.
        "OPTIONS": {
            # In case of using an older MySQL server, which has MyISAM as a default storage
            # "init_command": "SET storage_engine=INNODB",
            # 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
            # "connect_timeout": 28800,
            # Persistent connections
            "CONN_MAX_AGE": 0,
            # Disable server-side cursors, might be needed with pgbouncer
            "DISABLE_SERVER_SIDE_CURSORS": False,
        }
    }
}

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Data directory
DATA_DIR = os.path.join(BASE_DIR, "data")

# 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",
    "bg", "Български",
    "br", "Brezhoneg",
    "ca", "Català",
    "cs", "Čeština",
    "da", "Dansk",
    "de", "Deutsch",
    "en", "English"}
SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
USE_L10N = True

# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True

# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded_1
# files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(DATA_DIR, "static")
# URL prefix for static files.
STATIC_URL = f"{URL_PREFIX}/static/"

# Additional locations of static files
STATICFILES_DIRS = (
    # Put strings here, like "/home/html/static" or "C:/www/django/static".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    "django.contrib.staticfiles.finders.FileSystemFinder",
    "django.contrib.staticfiles.finders.AppDirectoriesFinder",
    "compressor.finders.CompressorFinder",
)

# Make this unique, and don't share it with anybody.
# You can generate it using weblate/examples/generate-secret-key
SECRET_KEY = ""

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
        },
        "APP_DIR": True,
    }
]

# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_USERNAME = None
GITLAB_TOKEN = None

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.github.GithubOAuth2",
    "social_core.backends.bitbucket.BitbucketOAuth",
    "social_core.backends.suse.OpenSUEReOpenId",
    "social_core.backends.ubuntu.UbuntuOpenId",
    "social_core.backends.fedora.FedoraOpenId",
    "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"
# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
SOCIAL_AUTH_GITHUB_ORG_KEY = ""
SOCIAL_AUTH_GITHUB_ORG_SECRET = ""
SOCIAL_AUTH_GITHUB_ORG_NAME = ""
SOCIAL_AUTH_GITHUB_TEAM_KEY = ""
SOCIAL_AUTH_GITHUB_TEAM_SECRET = ""
SOCIAL_AUTH_GITHUB_TEAM_ID = ""
SOCIAL_AUTH_BITBUCKET_KEY = ""
SOCIAL_AUTH_BITBUCKET_SECRET = ""
SOCIAL_AUTH_BITBUCKET_VERIFIED_EMAILS_ONLY = True
SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (
    "social_core.pipeline.social_auth.social_details",
    "social_core.pipeline.social_auth.social_uid",
    "social_core.pipeline.social_auth.auth_allowed",
    "social_core.pipeline.social_auth.social_user",
    "weblate.accounts.pipeline.store_params",
    "weblate.accounts.pipeline.verify_open",
    "social_core.pipeline.user.get_username",
    "weblate.accounts.pipeline.require_email",
    "social_core.pipeline.mail.mail_validation",
    "weblate.accounts.pipeline.revoke_mail_code",
    "weblate.accounts.pipeline.ensure_valid",
    "weblate.accounts.pipeline.remove_account",
    "social_core.pipeline.social_auth.associate_by_email",
    "weblate.accounts.pipeline.reauthenticate",
    "weblate.accounts.pipeline.verify_username",
    "social_core.pipeline.user.create_user",
    "social_core.pipeline.social_auth.associate_user",
    "social_core.pipeline.social_auth.load_extra_data",
    "weblate.accounts.pipeline.cleanup_next",
    "weblate.accounts.pipeline.user_full_name",
    "weblate.accounts.pipeline.store_email",
    "weblate.accounts.pipeline.notify_connect",
    "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",
    "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_email_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/"
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.MinLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
    {
        "NAME": "django.contrib.auth.password_validation.CommonPasswordValidator",
        "NAME": "django.contrib.auth.password_validation.PacksPasswordValidator"
    },
    {
        "NAME": "weblate.accounts.password_validation.PastPasswordsValidator"
    },
]

# Optional password strength validation by django-zxcvbn-password
# { "NAME": "zxcvbn_password.ZXCVBNValidator",
#   "OPTIONS": {
#       "min_score": 3,
#       "user_attributes": ("username", "email", "full_name")
#   }
# }

# Password hashing (prefer Argon)
PASSWORD_HASHERS = [
    "django.contrib.auth.hashers.Argon2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher",
    "django.contrib.auth.hashers.BCryptSHA256PasswordHasher",
]

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
]
"social_django.middleware.SocialAuthExceptionMiddleware",
"weblate.accounts.middleware.RequireLoginMiddleware",
"weblate.api.middleware.ThrottlingMiddleware",
"weblate.middleware.SecurityMiddleware",
"weblate.wladmin.middleware.ManageMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.forms",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",
    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
    "weblate.gitexport",
    # Standard Django modules
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    # 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)
        LOGGING['loggers']['weblate']['handlers'] = ['syslog']
        LOGGING['loggers']['weblate']['level'] = 'DEBUG'
    except:
        pass
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\n%(message)s"},
        "simple": {"format": "%(asctime)s\n%(levelname)s\n%(message)s"},
        "logfile": {"format": "%(asctime)s\n%(levelname)s\n%(message)s"},
        "django.server": {"()": "django.utils.log.ServerFormatter", "format": "\n%(server_time)s\n%(levelname)s\n%(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,
},
# Python Social Auth
"social": {
    "handlers": [DEFAULT_LOG],
    "level": DEFAULT_LOGLEVEL,
},
# Django Authentication Using LDAP
"django_auth_ldap": {
    "handlers": [DEFAULT_LOG],
    "level": DEFAULT_LOGLEVEL,
},
# SAML IdP
"djangosaml2idp": {
    "handlers": [DEFAULT_LOG],
    "level": DEFAULT_LOGLEVEL,
},

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = {
    # "weblate.machinery.apertium.ApertiumAPYTranslation",
    # "weblate.machinery.baidu.BaiduTranslation",
    # "weblate.machinery.deepl.DeepLTranslation",
    # "weblate.machinery.glosbe.GlosbeTranslation",
    # "weblate.machinery.google.GoogleTranslation",
    # "weblate.machinery.googlev3.GoogleV3Translation",
    # "weblate.machinery.libretranslate.LibreTranslateTranslation",
    # "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    # "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    # "weblate.machinery.modernmt.ModernMTTranslation",
    # "weblate.machinery.mymemory.MyMemoryTranslation",
    # "weblate.machinery.netease.NeteaseSightTranslation",
    # "weblate.machinery.tmserver.AmagamaTranslation",
    # "weblate.machinery.tmserver.TMServerTranslation",
    # "weblate.machinery.yandex.YandexTranslation",
    # "weblate.machinery.saptranslationhub.SAPTranslationHub",
    # "weblate.machinery.youdao.YoudaoTranslation",
    # "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.memory.machine.WeblateMemory",
}

# Machine translation API keys
# URL of the Apertium APy server
MT_APERTIUM_APY = None
# DeepL API key
MT_DEEPL_KEY = None
# LibreTranslate
MT_LIBRETRANSLATE_API_URL = None
MT_LIBRETRANSLATE_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

# ModernMT
MT_MODERNMT_KEY = None

# MyMemory identification email, see
# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
MT_GOOGLE_CREDENTIALS = None
MT_GOOGLE_PROJECT = None

# Baidu app key and secret
MT_BAIDU_ID = None
MT_BAIDU_SECRET = None

# Youdao Zhiyun app key and secret
MT_YOUDAO_ID = None
MT_YOUDAO_SECRET = None

# NetEase Sight (Jianwai) app key and secret
MT_NETEASE_KEY = None
MT_NETEASE_SECRET = None

# API key for Yandex Translate API
MT_YANDEX_KEY = None

# tmserver URL
MT_TMSERVER = None

# SAP Translation Hub
MT_SAP_BASE_URL = None
MT_SAP_SANDBOX_APIKEY = None
MT_SAP_USERNAME = None
MT_SAP_PASSWORD = None
MT_SAP_USE_MT = True

# Use HTTPS when creating redirect URLs for social authentication, see
# documentation for more details:
# https://python-social-auth-docs.readthedocs.io/en/latest/configuration/
# settings.html#processing-redirects-and-urlopen
SOCIAL_AUTH_REDIRECT_IS_HTTPS = ENABLE_HTTPS

# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view
CSRF_FAILURE_VIEW = "weblate.trans.views.error.csrf_failure"
SESSION_COOKIE_SECURE = ENABLE_HTTPS
SESSION_COOKIE_HTTPONLY = True
# SSL redirect
SECURE_SSL_REDIRECT = ENABLE_HTTPS
# Sent referrer only for same origin links
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",)  # Allowing HTTP access to health_→check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
SESSION_COOKIE_SAMESITE = "lax"
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 5000000

# Apply session cookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer
# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
SIMPLIFY_LANGUAGES = True
# Render forms using bootstrap
CRISPY_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
CHECK_LIST = (  
    "weblate.checks.same.SameCheck",  
    "weblate.checks.chars.BeginNewlineCheck",  
    "weblate.checks.chars.EndNewlineCheck",  
    "weblate.checks.chars.BeginSpaceCheck",  
    "weblate.checks.chars.EndSpaceCheck",  
    "weblate.checks.chars.DoubleSpaceCheck",  
    "weblate.checks.chars.EndStopCheck",  
    "weblate.checks.chars.EndColonCheck",  
    "weblate.checks.chars.EndQuestionCheck",  
    "weblate.checks.chars.EndExclamationCheck",  
    "weblate.checks.chars.EndSemicolonCheck",  
    "weblate.checks.chars.MaxLengthCheck",  
    "weblate.checks.chars.KashidaCheck",  
    "weblate.checks.chars.PunctuationSpacingCheck",  
    "weblate.checks.format.PythonFormatCheck",  
    "weblate.checks.format.PythonBraceFormatCheck",  
    "weblate.checks.format.PHPFormatCheck",  
    "weblate.checks.format.CFormatCheck",  
    "weblate.checks.format.PerlFormatCheck",  
    "weblate.checks.format.JavaScriptFormatCheck",  
    "weblate.checks.format.LuaFormatCheck",  
    "weblate.checks.format.ObjectPascalFormatCheck",  
    "weblate.checks.format.SchemeFormatCheck",  
    "weblate.checks.format.CSharpFormatCheck",  
    "weblate.checks.format.JavaFormatCheck",  
    "weblate.checks.format.JavaMessageFormatCheck",  
    "weblate.checks.format.PercentPlaceholdersCheck",  
    "weblate.checks.format.ESTemplateLiteralsCheck",  
    "weblate.checks.angularjs.AngularJSInterpolationCheck",  
    "weblate.checks.icu.ICUMessageFormatCheck",  
    "weblate.checks.icu.ICUSourceCheck",  
    "weblate.checks.qt.QtFormatCheck",  
    "weblate.checks.qt.QtPluralCheck",  
    "weblate.checks.ruby.RubyFormatCheck",  
    "weblate.checks.consistency.PluralsCheck",  
    "weblate.checks.consistency.SamePluralsCheck",  
    "weblate.checks.consistency.ConsistencyCheck",  
    "weblate.checks.consistency.TranslatedCheck",  
    "weblate.checks.chars.EscapedNewlineCountingCheck",  
    "weblate.checks.chars.NewLineCountCheck",  
    "weblate.checks.chars.ZeroWidthSpaceCheck",  
    "weblate.checks.render.MaxLengthCheck",  
    "weblate.checks.markup.XMLValidityCheck",  
    "weblate.checks.markup.XMLTagsCheck",  
    "weblate.checks.markup.MarkdownRefLinkCheck",  
    "weblate.checks.markup.MarkdownSyntaxCheck",  
    "weblate.checks.markup.URLCheck",  
    "weblate.checks.markup.SafeHTMLCheck",  
    "weblate.checks.placeholders.PlaceholderCheck",  
    "weblate.checks.placeholders.RegexCheck",  
    "weblate.checks.duplicate.DuplicateCheck",  
    "weblate.checks.source.OptionalPluralCheck",  
    "weblate.checks.source.EllipsisCheck",  
    "weblate.checks.source.MultipleFailingCheck",  
    "weblate.checks.source.LongUntranslatedCheck",  
    "weblate.checks.format.MultipleUnnamedFormatsCheck",  
)
List of automatic fixups

```python
AUTOFIX_LIST = (
    "weblate.checks.glossary.GlossaryCheck",
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
    "weblate.trans.autofixes.chars.RemoveZeroSpace",
    "weblate.trans.autofixes.chars.RemoveControlChars",
)
```

List of enabled addons

```python
WEBLATE_ADDONS = (
    "weblate.addons.gettext.GenerateMoAddon",
    "weblate.addons.gettext.UpdateLinguasAddon",
    "weblate.addons.gettext.UpdateConfigureAddon",
    "weblate.addons.gettext.MsgmergeAddon",
    "weblate.addons.gettext.GettextCustomizeAddon",
    "weblate.addons.cleanup.CleanupAddon",
    "weblate.addons.cleanup.RemoveBlankAddon",
    "weblate.addons.consistency.LanguageConsistencyAddon",
    "weblate.addons.discovery.DiscoveryAddon",
    "weblate.addons.autotranslate.AutoTranslateAddon",
    "weblate.addons.flags.SourceEditAddon",
    "weblate.addons.flags.TargetEditAddon",
    "weblate.addons.flags.SameEditAddon",
    "weblate.addons.flags.BulkEditAddon",
    "weblate.addons.generate.GenerateFileAddon",
    "weblate.addons.generate.PseudolocaleAddon",
    "weblate.addons.generate.PrefillAddon",
    "weblate.addons.json.JSONCustomizeAddon",
    "weblate.addons.properties.PropertiesSortAddon",
    "weblate.addons.git.GitSquashAddon",
    "weblate.addons.remove.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.

```python
SERVER_EMAIL = "noreply@example.com"
```

Default email address to use for various automated correspondence from the site managers. Used for registration emails.

```python
DEFAULT_FROM_EMAIL = "noreply@example.com"
```

List of URLs your site is supposed to serve

```python
ALLOWED_HOSTS = ["*"]
```

Configuration for caching

```python
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/1",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
        "KEY_PREFIX": "weblate",
    },
```
# 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",
}
# 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.
  "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
SENTRY_ENVIRONMENT = SITE_DOMAIN
AKISMET_API_KEY = None
You will find basic management commands (available as ./manage.py in the Django sources, or as an extended set in a script called `weblate` installable atop Weblate).

**Invoking management commands**

As mentioned before, invocation depends on how you installed Weblate.

If using virtualenv for Weblate, you can either specify the full path to `weblate`, or activate the virtualenv prior to invoking it:

```
# Direct invocation
~/weblate-env/bin/weblate
# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

If you are using source code directly (either from a tarball or Git checkout), the management script is `./manage.py` available in the Weblate sources. To run it:

```
python ./manage.py list_versions
```

If you've installed Weblate using the pip or pip3 installer, or by using the `./setup.py` script, the `weblate` is installed to your path (or virtualenv path), from where you can use it to control Weblate:

```
weblate list_versions
```

For the Docker image, the script is installed like above, and you can run it using `docker exec`:

```
docker exec --user weblate <container> weblate list_versions
```

For `docker-compose` the process is similar, you just have to use `docker-compose exec`:

```
docker-compose exec --user weblate weblate weblate list_versions
```

In case you need to pass it a file, you can temporary add a volume:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /tmp/users.json
```

**add_suggestions**

```
weblate add_suggestions <project> <component> <language> <file>
```

Imports a translation from the file to use as a suggestion for the given translation. It skips duplicated translations; only different ones are added.

---**author** USER@EXAMPLE.COM

E-mail of author for the suggestions. This user has to exist prior to importing (you can create one in the admin interface if needed).

```
weblate --author michal@cihar.com add_suggestions weblate application cs /tmp/suggestions-cs.po
```

---
**auto_translate**

`weblate auto_translate <project> <component> <language>`

**2.5+**

Performs automatic translation based on other component translations.

```
--source PROJECT/COMPONENT
```

Specifies the component to use as source available for translation. If not specified all components in the project are used.

```
--user USERNAME
```

Specify username listed as author of the translations. "Anonymous user" is used if not specified.

```
--overwrite
```

Whether to overwrite existing translations.

```
--inconsistent
```

Whether to overwrite existing translations that are inconsistent (see `--threshold`).

```
--add
```

Automatically add language if a given translation does not exist.

```
--mt MT
```

Use machine translation instead of other components as machine translations.

```
--threshold THRESHOLD
```

Similarity threshold for machine translation, defaults to 80.

```
--mode MODE
```

Specify translation mode, default is `translate` but `fuzzy` or `suggest` can be used.

```bash
weblate auto_translate --user nijel --inconsistent --source weblate/application weblate website cs
```

**celery_queues**

`weblate celery_queues`

**3.7**

Displays length of Celery task queues.

**checkgit**

`weblate checkgit <project|project/component>`

Prints current state of the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.
**commitgit**

`weblate commitgit <project|project/component>`

Commits any possible pending changes to the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

**commit_pending**

`weblate commit_pending <project|project/component>`

Commits pending changes older than a given age.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

`--age HOURS`

Age in hours for committing. If not specified the value configured in `WEBLATE` is used.

---

This is automatically performed in the background by Weblate, so there no real need to invoke this manually, besides forcing an earlier commit than specified by `COMMIT_PENDING_HOURS`.

---

**cleanuptrans**

`weblate cleanuptrans`

Cleans up orphaned checks and translation suggestions. There is normally no need to run this manually, as the cleanups happen automatically in the background.

---

**cleanup_ssh_keys**

`weblate cleanup_ssh_keys`

Performs cleanup of stored SSH host keys:

- Removes deprecated RSA keys for GitHub which might cause issues connecting to GitHub.
- Removes duplicate entries in host keys.

---

**createadmin**

`weblate createadmin`

Creates an `admin` account with a random password, unless it is specified.

`--password PASSWORD`

Provides a password on the command-line, to not generate a random one.

`--no-password`

Do not set password, this can be useful with `--update`.

`--username USERNAME`

Use the given name instead of `admin`. 313
Specify the admin e-mail address.

Specify the admin name (visible).

Update the existing user (you can use this to change passwords).

Added parameters --username, --email, --name and --update.

**dump_memory**

`weblate dump_memory`

Export a JSON file containing Weblate Translation Memory content.

`weblate dumpuserdata`

Dumps userdata to a file for later use by `importuserdata`.

This comes in handy when migrating or merging Weblate instances.

**import_demo**

`weblate import_demo`

Creates a demo project with components based on `<https://github.com/WeblateOrg/demo>`.

This can be useful when developing Weblate.

**import_json**

`weblate import_json <json-file>`

Batch import of components based on JSON data.

The imported JSON file structure pretty much corresponds to the component object (see GET /api/components/(string:project)/(string:component)/). You have to include the name and filemask fields.

Specifies where the components will be imported from.

Use the given VCS repository from this component for all of them.

Skip (already) imported components.

Update (already) imported components.

The parameters --ignore and --update are there to deal with already imported components.

Example of JSON file:
**import_memory**

**weblate import_memory <file>**

**2.20**

Imports a TMX or JSON file into the Weblate translation memory.

--language-map LANGMAP

Allows mapping languages in the TMX to the Weblate translation memory. The language codes are mapped after normalization usually done by Weblate.

--language-map en_US:en will for example import all en_US strings as en ones.

**TMX** Weblate **JSON** Weblate

**import_project**

**weblate import_project <project> <gitrepo> <branch> <filemask>**

**3.0**

The import_project command is now based on the Weblate add-on, leading to some changes in behavior and what parameters are accepted.

Batch imports components into project based on the file mask.

<project> names an existing project, into which the components are to be imported.

The <gitrepo> defines the Git repository URL to use, and <branch> signifies the Git branch. To import additional translation components from an existing Weblate component, use a weblate://<project>/<component> URL for the <gitrepo>.

The <filemask> defines file discovery for the repository. It can be either be made simple using wildcards, or it can use the full power of regular expressions.

The simple matching uses ** for component name and * for language, for example: **/*.po

The regular expression has to contain groups named component and language. For example: (?P<language>[^/]+)\([^-]*\)\.*.po

The import matches existing components based on files and adds the ones that do not exist. It does not change already existing ones.

--name-template TEMPLATE

Customize the name of a component using Django template syntax.

Documentation: {{ component }}
--base-file-template TEMPLATE

\n
{{ component }}}/res/values/string.xml

--new-base-template TEMPLATE

\n
{{ component }}}/ts/en.ts

--file-format FORMAT

You can also specify the file format to use (see \texttt{--file-format}), the default is auto-detection.

--language-regex REGEX

You can specify language filtering (see \texttt{--language-regex}) with this parameter. It has to be a valid regular expression.

--main-component

You can specify which component will be chosen as the main one—the one actually containing the VCS repository.

--license NAME

Specify the overall, project or component translation license.

--license-url URL

Specify the URL where the translation license is to be found.

--vcs NAME

In case you need to specify which version control system to use, you can do it here. The default version control is Git.

To give you some examples, let’s try importing two projects.

First The Debian Handbook translations, where each language has separate a folder with the translations of each chapter:

\begin{verbatim}
weblate import_project \\
 debian-handbook \\
 git://anonscm.debian.org/debian-handbook/debian-handbook.git \\
 squeeze/master \\
 '*/**.po'
\end{verbatim}

Then the Tanaguru tool, where the file format needs be specified, along with the base file template, and how all components and translations are located in single folder:

\begin{verbatim}
weblate import_project \\
--file-format=properties \\
--base-file-template=web-app/tgol-web-app/src/main/resources/i18n/%s-
\textbackslash{}I18N.properties \\
 tanaguru \\
 https://github.com/Tanaguru/Tanaguru \\
 master \\
 web-app/tgol-web-app/src/main/resources/i18n/**-I18N_*.properties
\end{verbatim}

More complex example of parsing of filenames to get the correct component and language out of a filename like src/security/Numerous_security_holes_in_0.10.1.de.po:

\begin{verbatim}
weblate import_project \\
tails \\
 git://git.tails.boum.org/tails master \\
 'wiki/src/security/(?P<component>.*)\.(?P<language>[^.]*)\.*.po$
\end{verbatim}

Filtering only translations in a chosen language:

\begin{verbatim}
./manage import_project \\
 --language-regex '^(cs|sk)$' \\
 weblate \\
 https://github.com/WeblateOrg/weblate.git \\
 'weblate/locale/*/LC_MESSAGES/**.po'
\end{verbatim}

Importing Sphinx documentation split to multiple files:

\begin{verbatim}
$ weblate import_project --name-template 'Documentation: %s' \\
 --file-format po \\
 project https://github.com/project/docs.git master \\
 'docs/locale/*/LC_MESSAGES/**.po'
\end{verbatim}
Importing Sphinx documentation split to multiple files and directories:

```
$ weblate import_project --name-template 'Directory 1: %s'\
  --file-format po \project https://github.com/project/docs.git master \'
docs/locale/*/LC_MESSAGES/dir1/**.po'
$ weblate import_project --name-template 'Directory 2: %s'\
  --file-format po \project https://github.com/project/docs.git master \'
docs/locale/*/LC_MESSAGES/dir2/**.po'
```

More detailed examples can be found in the starting chapter, alternatively you might want to use `import_json`.

**importuserdata**

`weblate importuserdata <file.json>`

Imports user data from a file created by `dumpuserdata`

**importusers**

`weblate importusers --check <file.json>`

Imports users from JSON dump of the Django auth_users database.

`--check`

With this option it will just check whether a given file can be imported and report possible conflicts arising from usernames or e-mails.

You can dump users from the existing Django installation using:

```
weblate dumpdata auth.User > users.json
```

**install_addon**

```
$ weblate install_addon --addon ADDON <project|project/component>\
  --configuration CONFIG\
  --update
```

Name of the add-on to install. For example `weblate.gettext.customize`.

`--configuration` `CONFIG`

You can either define which project or component to install the add-on in (for example `weblate/application`), or use `--all` to include all existing components.

To install `gettext` for all components:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --update --all
```
list_languages

`weblate list_languages <locale>`

Lists supported languages in MediaWiki markup - language codes, English names and localized names. This is used to generate [https://wiki.l10n.cz/Slovn%C3%C4k_s_n%C3%A1zvy_jazyk%C5%AF](https://wiki.l10n.cz/Slovn%C3%C4k_s_n%C3%A1zvy_jazyk%C5%AF).

list_translators

`weblate list_translators <project|project/component>`

Lists translators by contributed language for the given project:

```
[French]
Jean Dupont <jean.dupont@example.com>
[English]
John Doe <jd@example.com>
```

--language-code
List names by language code instead of language name.
You can either define which project or component to use (for example `weblate/application`), or use `--all` to list translators from all existing components.

list_versions

`weblate list_versions`

Lists all Weblate dependencies and their versions.

loadpo

`weblate loadpo <project|project/component>`

Reloads translations from disk (for example in case you have done some updates in the VCS repository).

--force
Force update, even if the files should be up-to-date.
--lang LANGUAGE
Limit processing to a single language.
You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

---
**Tip:** You seldom need to invoke this, Weblate will automatically load changed files for every VCS update. This is needed in case you manually changed an underlying Weblate VCS repository or in some special cases following an upgrade.

lock_translation

`weblate lock_translation <project|project/component>`

Prevents further translation of a component.

---
**Tip:** Useful in case you want to do some maintenance on the underlying repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

---

unlock_translation
**move_language**

*weblate move_language source target*

3.0

Allows you to merge language content. This is useful when updating to a new version which contains aliases for previously unknown languages that have been created with the *(generated)* suffix. It moves all content from the *source* language to the *target* one.

```bash
weblate move_language cze cs
```

After moving the content, you should check whether there is anything left (this is subject to race conditions when somebody updates the repository meanwhile) and remove the *(generated)* language.

**pushgit**

*weblate pushgit <project|project/component>*

VCS

--force-commit

Force commits any pending changes, prior to pushing.

You can either define which project or component to update (for example *weblate/application*), or use --all to update all existing components.

---

**unlock_translation**

*weblate unlock_translation <project|project/component>*

Unlocks a given component, making it available for translation.

---

**setupgroups**

*weblate setupgroups*

Configures default groups and optionally assigns all users to that default group.

--no-privs-update

Turns off automatic updating of existing groups (only adds new ones).

--no-projects-update

Prevents automatic updates of groups for existing projects. This allows adding newly added groups to existing projects, see  

---
setuplang

weblate setuplang
Updates list of defined languages in Weblate.

--no-update
Turns off automatic updates of existing languages (only adds new ones).

updatechecks

weblate updatechecks <project|project/component>
Updates all checks for all strings.

**Note:** Useful for upgrades which do major changes to checks.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

updategit

weblate updategit <project|project/component>
Fetches remote VCS repositories and updates the internal cache.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

**Note:** Usually it is better to configure hooks in the repository to trigger `updategit`, instead of regular polling by `updategit`.

---

4.0

Manage

Post announcement
Translations will be used only if they reach 60%.

Post announcement

Message

You can use Markdown and mention users by @username.

Category
Info (light blue)

Category defines color used for the message.

Expiry date
mm/dd/yyyy

The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

Notify users

The message is shown for all translations within the project, until its given expiry, or permanently until it is deleted.

Add
Add Announcement

Required fields are marked in bold.

Message: 
Translations will be used only if they reach 40%.

You can use Markdown and mention users by @username.

Project: WebateMsg

Component: 

Language: 

Category: Info (light blue)
Category defines color used for the message.

Notify users

<table>
<thead>
<tr>
<th>Notify users</th>
<th>Save and add another</th>
<th>Save and continue editing</th>
<th>Save</th>
</tr>
</thead>
</table>

The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.
1. Define **Automatic component list assignment** with `^.*$` as regular expression in both the project and the component fields, as shown on this image:
### Weblate administration

**Add Component list**

- **Component list name:** All components
- **URL slug:** all-components

**Show on dashboard:** When enabled this component list will be shown as a tab on the dashboard.

#### Components:

- Available components
- WeblateOrg/Synop
- WeblateOrg/language names
- WeblateOrg/WeblateOrg

---

### settings.py

```python
INSTALLED_APPS += ("weblate.gitexport",)
```

### weblate migrate

324
Weblate URL: https://example.org/git/weblate/main/

To clone the repository, run:

```
$ git clone https://example.org/git/weblate/main/
```

**Access control**

Access to the repository is controlled through `REQUIRE_LOGIN`.

To clone the repository with access control enabled, run:

```
$ git clone https://user:KEY@example.org/git/weblate/main/
```

### 2.4 Hosted Weblate

#### 1. Add `weblate.billing` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.billing",)
```

#### 2. Run `weblate migrate`
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. `weblate migrate`

3. `weblate/legal/templates/legal/`

   Weblate UI

   ENABLE_AVATARS

   Weblate UI

   Gravatar

   Libravatar

   AVATAR_URL_PREFIX

   ENABLE_AVATARS

Akismet

1. `akismet` Python

2. Akismet API

3. `WEBLATE_AKISMET_API_KEY` Docker

   Akismet API

   IP

   `WEBLATE_AKISMET_API_KEY`
GnuPG

Weblate

1. Turn on `WEBLATE_GPG_IDENTITY`. (Weblate will generate a GnuPG key when needed and will use it to sign all translation commits.)

2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking `gpg`.

WEBLATE_GPG_IDENTITY

3.2

4.6

Weblate

RATELIMIT_WINDOW

RATELIMIT_ATTEMPTS

RATELIMIT_CONTACT_ATTEMPTS

RATELIMIT_TRANSLATE_ATTEMPTS
Customizing Weblate

Extend and customize using Django and Python. Contribute your changes upstream so that everybody can benefit. This reduces your maintenance costs; code in Weblate is taken care of when changing internal interfaces or refactoring the code.

Neither internal interfaces nor templates are considered a stable API. Please review your own customizations for every upgrade, the interfaces or their semantics might change without notice.

Creating a Python module

If you are not familiar with Python, you might want to look into Python For Beginners, explaining the basics and pointing to further tutorials.

To write some custom Python code (called a module), a place to store it is needed, either in the system path (usually something like /usr/lib/python3.7/site-packages/) or in the Weblate directory, which is also added to the interpreter search path.

Better yet, turn your customization into a proper Python package:
1. Create a folder for your package (we will use weblate_customization).
2. Within it, create a setup.py file to describe the package:

   ```python
   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",
   )
   ```

Fedora Messaging

Fedora Messaging

AMQP

Weblate

Python

webate-fedora-messaging

<https://github.com/WeblateOrg/fedora_messaging/>

### Docker

```bash
Docker WEBLATE_ AUTH_LOCK_ATTEMPTS
WEBLATE_RATELIMIT_ATTEMPTS
```
3. Create a folder for the Python module (also called `weblate_customization`) for the customization code.
4. Within it, create a `__init__.py` file to make sure Python can import the module.
5. This package can now be installed using `pip install -e`. More info to be found in “Editable” Installs.
6. Once installed, the module can be used in the Weblate configuration (for example `weblate_customization.checks.FooCheck`).

Your module structure should look like this:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

You can find an example of customizing Weblate at <https://github.com/WeblateOrg/customize-example>, it covers all the topics described below.

### Changing the logo

1. **Django Creating a Python module**

   Branding appears in the following files:

   /weblate.svg Logo shown in the navigation bar.
   logo-*.png Web icons depending on screen resolution and web-browser.
   favicon.ico Web icon used by legacy browsers.
   weblate-*.png Avatars for bots or anonymous users. Some web-browsers use these as shortcut icons.
   email-logo.png Used in notifications e-mails.

2. Add it to `INSTALLED_APPS`:

   ```python
   INSTALLED_APPS = (
       # Add your customization as first
       "weblate_customization",
       # Weblate apps are here...
   )
   ```

3. Run `weblate collectstatic --noinput`, to collect static files served to clients.

---

### How to manage static files (e.g. images, JavaScript, CSS)

---

### Weblate Python Creating a Python module

1. **Python**

2. **Python WEBSITE_ADDONS CHECK_LIST AUTOFIX_LIST**

   ```python
   # Checks
   CHECK_LIST += ("weblate_customization.checks.FooCheck",)
   
   # Autofixes
   AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)
   
   # Add-ons
   WEBSITE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
   ```

---

---
Web: Weblate

Weblate version: 4.11 - 94f33b4b19b9da758e6f5b33e0b98c8c6a50e5

Support status: Community support

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

Web: Weblate

Weblate Celery

SSH: alerts

Django

Web: Weblate

Weblate

SSH: alerts

Django
# Weblate administration

## Site administration

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Action 1</th>
<th>Action 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reports</td>
<td></td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Accounts</td>
<td>Audit log entry</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>User profiles</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Verified e-mails</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Auth Token</td>
<td>Tokens</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Authentication</td>
<td>Groups</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Roles</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Users</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Billing</td>
<td>Billing plans</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Customer billings</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Invokes</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Fonts</td>
<td>Font groups</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Fonts</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Legal</td>
<td>TOS agreements</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Python Social Auth</td>
<td>Associations</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Nonces</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>User social audits</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Screenshots</td>
<td>Screenshots</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Translation Memory</td>
<td>Translation memory entries</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Website Configuration</td>
<td>Settings</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Website Languages</td>
<td>Languages</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td>Website Translators</td>
<td>Announcements</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Component lists</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Components</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Contributor agreements</td>
<td>Add</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Projects</td>
<td>Add</td>
<td>Change</td>
</tr>
</tbody>
</table>

## Recent actions

<table>
<thead>
<tr>
<th>Type</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>My actions</td>
<td>None available</td>
</tr>
</tbody>
</table>
### Add Project

**Project name:** Weblate.org

**URL slug:** weborg

**Project website:** https://web.org/ (Main source of translated project)

**Translation instructions:**

```
You can use Markdown and mention users by @username.
```

- [ ] Set "language-team" header
  - Let Weblate update the "language-team" file header of your project.

- [ ] Use shared translation memory
  - Uses the pool of shared translations between projects.

- [ ] Contribute to shared translation memory
  - Contributes to the pool of shared translations between projects.

**Access control:**

- [ ] Protected
  - How to request access to this project is detailed in the documentation.

- [ ] Enable reviewers
  - Requires dedicated reviewers to approve translations.

- [ ] Enable source reviewers
  - Requires dedicated reviewers to approve source strings.

- [ ] Enable hooks
  - Whether to allow updating this repository by remote hooks.

**Language aliases:**

- Comma-separated list of language code mappings, for example: en,de,es,fr,ru,en

---

**SSH:**

**Weblate translations:**

**Weblate languages:**
Weblate

Weblate Libre

<https://weblate.org/support/>

3.8

Weblate URL

SSH URL

Powered by Webate 4.11 About Webate Legal Contact Documentation Donate to Webate
Weblate 4.5.2

[AWeblate](https://weblate.org/discover/)

1. [https://weblate.org/user/]
2. Weblate [https://weblate.org/subscription/discovery/]

338
3. Confirm the service activation in your Weblate and turn on the discovery listing in your Weblate management page using *Enable discovery* button:

You can customize the listing by providing a text and image (570 x 260 pixels) at `<https://weblate.org/user/>`.

---

**Note**: Herein you will find various legal information you might need to operate Weblate in certain legal jurisdictions. It is provided as a means of guidance, without any warranty of accuracy or correctness. It is ultimately your responsibility to ensure that your use of Weblate complies with all applicable laws and regulations.

**ITAR**

Weblate can be run within your own datacenter or virtual private cloud. As such, it can be used to store ITAR or other export-controlled information, however, end users are responsible for ensuring such compliance. The Hosted Weblate service has not been audited for compliance with ITAR or other export controls, and does not currently offer the ability to restrict translations access by country.
Weblate does not contain any cryptographic code, but might be subject export controls as it uses third party components utilizing cryptography for authentication, data-integrity and confidentiality.

Most likely Weblate would be classified as ECCN 5D002 or 5D992 and, as publicly available libre software, it should not be subject to EAR (see Encryption items NOT Subject to the EAR).

Software components used by Weblate (listing only components related to cryptographic function):

- Weblate
- Git
- Python
- cURL

The strength of encryption keys depends on the configuration of Weblate and the third party components it interacts with, but in any decent setup it will include all export restricted cryptographic functions:

- In excess of 56 bits for a symmetric algorithm
- Factorisation of integers in excess of 512 bits for an asymmetric algorithm
- Computation of discrete logarithms in a multiplicative group of a finite field of size greater than 512 bits for an asymmetric algorithm
- Discrete logarithms in a group different than above in excess of 112 bits for an asymmetric algorithm

Weblate doesn’t have any cryptographic activation feature, but it can be configured in a way where no cryptography code would be involved. The cryptographic features include:

- HTTPS
- Generating signatures for code commits (PGP)

Export Controls (EAR) on Open Source Software

Weblate

Contributing to Weblate modules

Weblate

Weblate

Weblate

Weblate

Weblate

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.
You are welcome to improve the documentation page of your choice. Do it easily by clicking the Edit on GitHub button in the top-right corner of the page.

Please respect these guidelines while writing:
1. Don’t remove part of the documentation if it’s valid.
2. Use clear and easily-understandable language. You are writing tech docs, not a poem. Not all docs readers are native speakers, be thoughtful.
3. Don’t be afraid to ask if you are not certain. If you have to ask about some feature while editing, don’t change its docs before you have the answer. This means: You change or ask. Don’t do both at the same time.
4. Verify your changes by performing described actions while following the docs.
5. Send PR with changes in small chunks to make it easier and quicker to review and merge.
6. If you want to rewrite and change the structure of a big article, do it in two steps:
   1. Rewrite
   2. Once the rewrite is reviewed, polished, and merged, change the structure of the paragraphs in another PR.

You can translate the docs.

The language definitions are in the weblate-language-data repository.
You are welcome to add missing language definitions to languages.csv, other files are generated from that file.

Weblate

issue
donate page

Weblate

Weblate

Weblate

Weblate

Weblate

Yashiro Ccs
Cheng-Chia Tseng
Timon Reinhard
Cassidy James
Loic Dachary
Marozed
https://freedombox.org/
GNU Solidario (GNU Health)
BallotReady
Richard Nespithal
MyExpenses.Mobi
Weblate

1. Weblate:
   ```
   git clone https://github.com/WeblateOrg/weblate.git
   cd weblate
   ```

2. virtualenv:
   ```
   virtualenv .venv
   .venv/bin/activate
   ```

3. Weblate:
   ```
   pip install -e .
   pip install -r requirements-dev.txt
   ```

4. Weblate:
   ```
   weblate runserver
   ```

5. Celery Worker:
   ```
   ./weblate/examples/celery start
   ```

6. Weblate:
   ```
   ./scripts/test-database
   ./manage.py test
   ```

Docker Weblate

Docker:
   ```
   docker-compose
   ```

   ```
   ./rundev.sh
   ```

Dockerfile:
   ```
   docker-compose.yml
   ```

   ```
   ./rundev.sh test --failfast weblate.machine
   ```

Docker:
   ```
   docker ps
   ```

342
.
./rundev.sh logs

Docker

PyCharm  Weblate

PyCharm Python IDE

GitHub PyCharm IDE:

PyCharm virtualenv
IDE virtualenv

2 PyCharm Django

Django weblate

Django console

Languages & Frameworks

Django

Settings

Enable Django Support

Django project root:

Settings:

Do not use a Django test runner:

Manage tasks

Manage script:

Environment variables:

Folder pattern to track files:
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.
import_demo

Weblate

Any code should come with documentation explaining the behavior. Don’t forget documenting methods, complex code blocks, or user visible features. Any new code should utilize PEP 484 type hints. We’re not checking this in our CI yet as existing code does not yet include them.

Coding standard and linting the code

The easiest approach to enforce all this is to install pre-commit. The repository contains configuration for it to verify the committed files are sane. After installing it (it is already included in the requirements-lint.txt) turn it on by running pre-commit install in Weblate checkout. This way all your changes will be automatically checked.

```bash
pre-commit run --all
```

Debugging Weblate

Turning on debug mode will make the exceptions show in the web browser. This is useful to debug issues in the web interface, but not suitable for a production environment because it has performance consequences and might leak private data.

In a production environment, use ADMINS to receive e-mails containing error reports, or configure error collection using a third-party service.
Weblate logs

Weblate can produce detailed logs of what is going on in the background. In the default configuration it uses syslog and that makes the log appear either in /var/log/messages or /var/log/syslog (depending on your syslog daemon configuration).

The Celery process (see Celery) usually produces its own logs as well. The example system-wide setups logs to several files under /var/log/celery/.

Docker containers log to their output (as per usual in the Docker world), so you can look at the logs using docker-compose logs.

This contains LOGGING configuration.

Not processing background tasks

A lot of things are done in the background by Celery workers. If things like sending out e-mails or component removal does not work, there might a related issue.

Celery

Check the Celery queue status, either in celery, or using celery_queues

Look in the Celery logs for errors (see Weblate logs)

Not receiving e-mails from Weblate

You can verify whether outgoing e-mail is working correctly by using the sendtestemail management command (see Invoking management commands for instructions on how to invoke it in different environments) or by using config under the Tools tab.

These send e-mails directly, so this verifies that your SMTP configuration is correct (see ). Most of the e-mails from Weblate are however sent in the background and there might be some issues with Celery involved as well, please see Not processing background tasks for debugging that.

Analyzing application crashes

In case the application crashes, it is useful to collect as much info about the crash as possible. This can be achieved by using third-party services which can collect such info automatically. You can find info on how to set this up in.

Silent failures

Lots of tasks are offloaded to Celery for background processing. Failures are not shown in the user interface, but appear in the Celery logs. Configuring helps you to notice such failures easier.

In case Weblate performs badly in some scenario, please collect the relevant logs showing the issue, and anything that might help figuring out where the code might be improved.

In case some requests take too long without any indication, you might want to install dogslow along with and get pinpointed and detailed tracebacks in the error collection tool.
class weblate.addons.base.BaseAddon(storage=None)

classmethod can_install(component, user)

configure(settings)

daily(component)

classmethod get_add_form(user, component, **kwargs)

get_settings_form(user, **kwargs)

post_add(translation)

post_commit(component)

post_push(component)

post_update(component, previous_head: str, skip_push: bool)

previous_head(str) -- HEAD

skip_push (bool) -- commit and push if commit_pending

pre_commit(translation, author)

pre_push(component)

pre_update(component)

save_state()

store_post_load(translation, store)

It receives an instance of a file format class as an argument. This is useful to modify file format class parameters, for example adjust how the file will be saved.

unit_pre_create(unit)
from django.utils.translation import gettext_lazy as _

from weblate.addons.base import BaseAddon
from weblate.addons.events import EVENT_PRE_COMMIT

class ExampleAddon(BaseAddon):
    # Filter for compatible components, every key is
    # matched against property of component
    compat = {
        "file_format": {"po", "po-mono"}
    }
    # List of events add-on should receive
    events = (EVENT_PRE_COMMIT,)
    # Add-on unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example add-on")
    # Detailed add-on description
    description = _("This add-on does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return

Weblate

Weblate supports the latest, stable releases of all major browsers and platforms.

Alternative browsers which use the latest version of WebKit, Blink, or Gecko, whether directly or via the platform’s web view API, are not explicitly supported. However, Weblate should (in most cases) display and function correctly in these browsers as well.

Older browsers might work, but some features might be limited.
Adding new third-party library typically consists of:

```bash
# 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 .
```

```javascript
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'
```

Translation topic in the Django documentation

If you are not sure about your bug report or feature request, you can try Weblate.
In order to give the community time to respond and upgrade, you are strongly urged to report all security issues privately. HackerOne is used to handle security issues, and can be reported directly at HackerOne. Once you submit it there, community has limited but enough time to solve the incident.

Alternatively, report to security@weblate.org, which ends up on HackerOne as well.

If you don't want to use HackerOne, for whatever reason, you can send the report by e-mail to michal@cihar.com. You can choose to encrypt it using this PGP key 3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D. You can also get the PGP key from Keybase.

**Note:** Weblate depends on third-party components for many things. In case you find a vulnerability affecting one of those components in general, please report it directly to the respective project.

Some of these are:
- Django
- Django REST framework
- Python Social Auth

## Weblate testsuite and continuous integration

Testsuites exist for most of the current code, increase coverage by adding testcases for any new functionality, and verify that it works.

### Continuous integration

Current test results can be found on GitHub Actions and coverage is reported onCodecov.

There are several jobs to verify different aspects:

- **Documentation build and external links**
- **Code linting**
- **Setup verification** (ensures that generated dist files do not miss anything and can be tested)

The configuration for the CI is in .github/workflows directory. It heavily uses helper scripts stored in ci directory. The scripts can be also executed manually, but they require several environment variables, mostly defining Django settings file to use and database connection. The example definition of that is in scripts/test-database:

```bash
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=${1:-postgresql}

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

The simple execution can look like:

```bash
./scripts/test-database
./ci/run-migrate
./ci/run-test
./ci/run-docs
```
To run a testsuite locally, use:

```bash
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

**Note:** You will need a database (PostgreSQL) server to be used for tests. By default Django creates separate database to run tests with `test_` prefix, so in case your settings is configured to use `weblate`, the tests will use `test_weblate` database. See Weblate documentation for setup instructions.

The `weblate/settings_test.py` is used in CI environment as well (see Continuous integration) and can be tuned using environment variables:

```bash
# Simple way to configure test database from environment
# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=${1:-postgresql}

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

Prior to running tests you should collect static files as some tests rely on them being present:

```bash
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

You can also specify individual tests to run:

```bash
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.%
```

**Note:** The tests can also be executed inside developer docker container, see Docker for more info on running and writing tests for Django.

See Django documentation for more info on running and writing tests for Django.

**Weblate JSON Schema**

https://weblate.org/schemas/weblate-memory.schema.json

```json
{
    "schema":
    {
        "properties":
        {
            "category":
            {
                "type": "integer",
                "default": 1
            },
            "origin":
            {
                "type": "integer",
                "default": 0
            }
        }
    }
}
```
Table 8

<table>
<thead>
<tr>
<th>source</th>
<th>target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello</td>
<td>Ahoj</td>
</tr>
</tbody>
</table>

source_language:
- ^[^ ]+$

target_language:
- ^[^ ]+$

False:
- dump_memory
- import_memory

Weblate

https://weblate.org/schemas/weblate-userdata.schema.json

basic:
- username: Weblate
- full_name: Weblate
- email: noreply@example.com
- dateJoined: 2019-11-18T18:53:54.862Z

profile:
- language: cs
- ^+$
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>suggested</td>
<td>True</td>
</tr>
<tr>
<td>translated</td>
<td>True</td>
</tr>
<tr>
<td>uploaded</td>
<td>True</td>
</tr>
<tr>
<td>hide_completed</td>
<td>False</td>
</tr>
<tr>
<td>secondary_in_zen</td>
<td>True</td>
</tr>
<tr>
<td>hide_source_secondary</td>
<td>False</td>
</tr>
<tr>
<td>editor_link</td>
<td>True</td>
</tr>
<tr>
<td>translate_mode</td>
<td>True</td>
</tr>
<tr>
<td>zen_mode</td>
<td>True</td>
</tr>
<tr>
<td>special_chars</td>
<td>True</td>
</tr>
<tr>
<td>dashboard_view</td>
<td>True</td>
</tr>
<tr>
<td>dashboard_component_list</td>
<td>null</td>
</tr>
<tr>
<td>languages</td>
<td>cs</td>
</tr>
<tr>
<td>secondary_languages</td>
<td>sk</td>
</tr>
<tr>
<td>Table 9 –</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td><strong>watched</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>auditlog</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>address</strong></td>
<td><strong>IP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>user_agent</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>timestamp</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>activity</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Weblate**

---

356
GitHub: [GitHub](https://github.com/WeblateOrg/weblate/milestones)

---

1../scripts/list-translated-languages
2./scripts/prepare-release
3.make -j 12 -C docs update-screenshots
4.wlc push; git remote update; git merge origin/weblate
5./scripts/create-release --tag
6.Docker
7.GitHub
8.Docker
9.Helm
10./scripts/set-version
12.

---

Web: [GitHub](https://github.com/WeblateOrg/weblate)

---

**Tracking dependencies for vulnerabilities**

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.

---

There might be vulnerabilities in third-party libraries which do not affect Weblate, so those are not addressed by releasing bugfix versions of Weblate.
Docker

The Docker containers are regularly scanned using Anchore and Trivy security scanners. 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).

Continuous integration

Contributing to Weblate modules

Besides the main repository, Weblate consists of several Python modules. All these follow same structure and this documentation covers them all. For example, this covers:

- **wlc**, Python client library, see Weblate
- **translation-finder**, used to discover translatable files in the repository
- **language-data**, language definitions for Weblate, see

Coding guidelines

Any code should come with documentation explaining the behavior. Don't forget documenting methods, complex code blocks, or user visible features.

Any new code should utilize **PEP 484** type hints. We're not checking this in our CI yet as existing code does not yet include them.

Running tests

The tests are executed using **py.test**. First you need to install test requirements:

```
pip install -r requirements-test.txt
```

You can then execute the testsuite in the repository checkout:

```
py.test
```

The CI integration is very similar to Weblate testsuite and continuous integration.

Coding standard and linting the code

Any code should come with documentation explaining the behavior. Don't forget documenting methods, complex code blocks, or user visible features.

Any new code should utilize **PEP 484** type hints. We're not checking this in our CI yet as existing code does not yet include them.

Running tests

The tests are executed using **py.test**. First you need to install test requirements:

```
pip install -r requirements-test.txt
```

You can then execute the testsuite in the repository checkout:

```
py.test
```

The CI integration is very similar to Weblate testsuite and continuous integration.

Coding standard and linting the code

Any code should come with documentation explaining the behavior. Don't forget documenting methods, complex code blocks, or user visible features.

Any new code should utilize **PEP 484** type hints. We're not checking this in our CI yet as existing code does not yet include them.

Running tests

The tests are executed using **py.test**. First you need to install test requirements:

```
pip install -r requirements-test.txt
```

You can then execute the testsuite in the repository checkout:

```
py.test
```

The CI integration is very similar to Weblate testsuite and continuous integration.

Coding standard and linting the code

The easiest approach to enforce all this is to install **pre-commit**. The repository contains configuration for it to verify the committed files are sane. After installing it (it is already included in the requirements-lint.txt) turn it on by running **pre-commit install** in Weblate checkout. This way all your changes will be automatically checked.

```
pre-commit run --all
```

Weblate
Weblate

file formats

"Weblate" "web" "translate"

Web

https://weblate.org
https://hosted.weblate.org
https://docs.weblate.org

https://github.com/WeblateOrg/graphics

Michal Čihař michal@cihar.com

Weblate Michal Čihař 2012

Copyright © 2012–2022 Michal Čihař michal@cihar.com

Weblate 4.12

Not yet released.
Added support for Amharic in.
Added support for Burmese in.
Extended options of the add-on.
Added `ignore-all-checks` flag to ignore all quality checks on a string.
Avoid add-on to trigger failing checks.
Added support for Gitea pull requests.
Added Linux style language code to.
All changes in detail.
Weblate 4.11.2

2022 3 4

Fixed corrupted MO files in the binary release.
All changes in detail.

Weblate 4.11.1

2022 3 4

Fixed missing sanitizing of arguments to Git and Mercurial - CVE-2022-23915, see GHSA-3872-f48p-pxqj for more details.

CSV

API

ICU MessageFormat syntax

Indicate lock and contributor agreement on other occurrences listing.
Fixed updating PO files with obsolete strings or missing plurals.
Improved squash add-on compatibility with Gerrit.
Automatically initialize user languages based on the `Accept-Language` header.

Weblate Python 3.7

Sticky header on translations listing to improve navigation.

Java

Fixed Git operation with non-ascii branch names.

guilabel:fast-forward

Docker container automatically enables TLS/SSL for outgoing e-mail when needed.

borgbackup 1.2

Automatically translated

360
Weblate 4.10.1

2021 12 22

Documented changes introduced by upgrading to Django 4.0.
Fixed displaying of Automatically translated label.
Fixed API display of branch in components with a shared repository.
Fixed manually editing page when browsing changes.
Kashida

The Weblate Docker container now uses Python 3.10.

Weblate 4.10

2021 12 16

DeepL

Languages

XLSX

GitHub

API

GitHub SSH

Weblate 4.9.1

2021 11 19

API

GitHub SSH

Weblate 4.9

2021 11 10

The safe-html can now understand Markdown when used with md-text.
The max-length tag now ignores XML markup when used with xml-text.
Lowered app store title length to 30 to assist with upcoming Google policy changes.

**Weblate 4.8.1**

2021 9 10

Django

JavaScript

API

Added PRIVACY_URL setting to add privacy policy link to the footer.

gettext PO

Improved safe-html flag behavior with XML checks.

**Weblate 4.8**

2021 8 21

Apple stringsdict

PostgreSQL

Improved squash add-on compatibility with Gerrit.

Fluent

362
**Weblate 4.7.2**

2021 7 15

1 API
---
Git exporter URL
Windows RC
XLIFF
---

**Weblate 4.7.1**

2021 6 30

---
LibreTranslate
---

**Weblate 4.7**

2021 6 17

---
gettext PO
---
Object Pascal
---
mi18n
---
SAML

Fixed Gerrit integration to better handle corner cases.

Weblate now requires Django 3.2.
---

**Weblate 4.6.2**

2021 5 8

---
RTL
---
Git
---

363
Weblate 4.6.1

2021 5 2

Update list of user interface languages in Docker.

Pagure

Weblate 4.6

2021 4 19

The auto_translate management command has now a parameter for specifying translation mode.

Added date filtering when browsing changes.

Improved activity charts.

Sender for contact form e-mails can now be configured.

API

Docker API for creating components now automatically uses Weblate URL.

Simplified state indication while listing strings.

Renamed Argon2 to clarify the purpose.

XLIFF

Initial support for Scaling horizontally the Docker deployment.
Fixed possible loss of newly added strings on replace upload.

Weblate 4.5.2

Ignore format strings in the check.
Allow uploading screenshot from a translate page.
Added forced file synchronization to the repository maintenance.
Several performance improvements.

Weblate 4.5.1

Fixed editing of glossary flags in some corner cases.

TMX
API
Markdown
Improved bulk edit performance.
Fixed preserving "Needs editing" and "Approved" states for ODF files.
Weblate 4.5

2021 2 19

gettext PO lua-format

Fixed multiple unnamed variables check behavior with multiple format flags.
Dropped mailing list field on the project in favor of generic instructions for translators.

TermBase eXchange

Strings can now be added and removed in bilingual formats as well.

Amazon Translate

Java Java MessageFormat

Dropped specific API for glossaries as component API is used now.
Added simplified interface to toggle some of the flags.

Moved text direction toggle to get more space for the visual keyboard.

Added check whether translation matches the glossary.

Weblate 4.4.2

2021 1 14

Fixed corruption of one distributed MO file.

Weblate 4.4.1

2021 1 13

Fixed displaying help for project settings.

Fixed cleanup add-on behavior with HTML, ODF, IDML and Windows RC formats.
CSV

Use content compression for file downloads.

Improved user experience on importing from ZIP file.

Avoid duplicate pull requests on Pagure.
Reimplemented translation editor to use native browser textarea.

Added API for add-ons.

**Weblate 4.4**

2020 12 15

Weblate now requires Django 3.1.

**CodeMirror**

Syntax highlighting in translation editor for XML, HTML, Markdown and reStructuredText.

Improved support for non-standard language codes.

The user is now presented with a filtered list of languages when adding a new translation.

Improved search capabilities for changes in history.

Improved billing detail pages and Libre hosting workflow.

**API**

Added tasks API.

Improved display of user defined special characters.

Improved naming of ZIP downloads.

**Weblate 4.3.2**

2020 11 4

**Pagure**

**Markdown**

Simplified setup of Git repositories with different default branch than “master”.

Newly created internal repositories now use main as the default branch.

**reStructuredText**

Fixed CodeMirror display issues in some situations.

Renamed Template group to “Sources” to clarify its meaning.

Fixed GitLab pull requests on repositories with longer paths.
**Weblate 4.3.1**

2020 10 21

- Improve hooks compatibility with Bitbucket Server.
- Reduced memory usage.

**Weblate 4.3**

2020 10 15

- Include user stats in the API.
- Fixed component ordering on paginated pages.
- Rewritten support for GitHub and GitLab pull requests.
- Fixed configuration of enforced checks.
- Improve documentation about built-in backups.
- Vue 118n

- Generic placeholders check now supports regular expressions.
- Added support for interacting with multiple GitLab or GitHub instances.
- Extended API to cover project updates, unit updates and removals and glossaries.
- Unit API now properly handles plural strings.
- Component creation can now handle ZIP file or document upload.
- Consolidated API response status codes.
- Improved JSON, YAML and CSV formats compatibility.

- Improved performance of file downloads.
- Improved repository management view.
- Automatically enable java-format for Android.
- Python 3.9

- Fixed translating HTML files under certain conditions.
**Weblate 4.2.2**

2020 9 2

JSON

Fixed login redirect for some authentication configurations.
Fixed LDAP authentication with group sync.

Git

Fixed creating local VCS components using API.

---

**Weblate 4.2.1**

2020 8 21

Android

XLIFF

Allow setting up localization CDN in Docker image.

---

**Weblate 4.2**

2020 8 18

Improved user pages and added listing of users.
Dropped support for migrating from 3.x releases, migrate through 4.1 or 4.0.
Added exports into several monolingual formats.
Improved activity charts.
Number of displayed nearby strings can be configured.

Simplified main navigation (replaced buttons with icons).
Improved language code handling in Google Translate integration.
The Git squash add-on can generate Co-authored-by: trailers.
Improved query search parser.
Improved user feedback from format strings checks.
Improved performance of bulk state changes.
Added compatibility redirects after project or component renaming.
Added notifications for strings approval, component locking and license change.
Added support for ModernMT.
Allow to avoid overwriting approved translations on file upload.
Dropped support for some compatibility URL redirects.

ECMAScript

Removed leading dot from JSON unit keys.

Celery

Allow to configure Content-Security-Policy HTTP headers.
Added support for aliasing languages at project level.
New add-on to help with HTML or JavaScript localization, see JavaScript CDN.
The Weblate domain is now configured in the settings, see SITE_DOMAIN.

369
**Weblate 4.1.1**

2020  6  19

Fixed changing autofix or add-ons configuration in Docker.
Fixed possible crash in "About" page.

Fixed adding words to glossary.
Fixed keyboard shortcuts for machinery.
Removed debugging output causing discarding log events in some setups.
Fixed lock indication on project listing.
Fixed listing GPG keys in some setups.
Added option for which DeepL API version to use.
Added support for acting as SAML Service Provider, see SAML.

**Weblate 4.1**

2020  6  15

Added support for creating new translations with included country code.
Added support for searching source strings with screenshot.
Extended info available in the stats insights.
Improved search editing on "Translate" pages.
Improve handling of concurrent repository updates.

Include changes count in credits.
Fixed UI language selection in some cases.
Allow to whitelist registration methods with registrations closed.
Improved lookup of related terms in glossary.
Improved translation memory matches.
Group same machinery results.
Add direct link to edit screenshot from translate page.
Improved removal confirmation dialog.
Include templates in ZIP download.
Add support for Markdown and notification configuration in announcements.
Extended details in check listings.

Consistently use dismissed as state of dismissed checks.

Fixed editor keyboard shortcut to dismiss checks.
Improved machine translation of strings with placeholders.
Show ghost translation for user languages to ease starting them.
Improved language code parsing.
Show translations in user language first in the list.

Added new quality checks:

Consistently use dismissed as state of dismissed checks.

Fixed option to ignore source checks.
Added support for configuring different branch for pushing changes.
API HTTP

Added support for Google Translate V3 API (Advanced).
Added ability to restrict access on component level.
Added support for whitespace and other special chars in translation flags, see [API documentation](https://weblate.org/docs/api/translation/flags.html).
API now supports filtering of changes.
Added support for sharing glossaries between projects.

**Weblate 4.0.4**

2020 5 7

Fixed testsuite execution on some Python 3.8 environments.
Typo fixes in the documentation.
Fixed creating components using API in some cases.
Fixed JavaScript errors breaking mobile navigation.
Fixed crash on displaying some checks.
Fixed screenshots listing.
Fixed monthly digest notifications.
Fixed intermediate translation behavior with units non existing in translation.

**Weblate 4.0.3**

2020 4 27

Improved performance of translation stats.
Improved performance of changing labels.
Improved bulk edit performance.
Improved translation memory performance.
Fixed possible crash on component deletion.
Improved warning about too long celery queue.
Fixed possible false positives in the consistency check.
Fixed deadlock when changing linked component repository.
Included edit distance in changes listing and CSV and reports.
Avoid false positives of punctuation spacing check for Canadian French.
Fixed XLIFF export with placeholders.
Fixed false positive with zero width check.
Improved reporting of configuration errors.
Fixed bilingual source upload.
Automatically detect supported languages for DeepL machine translation.
Fixed progress bar display in some corner cases.
Fixed some checks triggering on non translated strings.

**Weblate 4.0.1**

2020  4  16

Fixed package installation from PyPI.

**Weblate 4.0**

2020  4  16

Weblate now requires Python 3.6 or newer.
Added management overview of component alerts.
Added component alert for broken repository browser URLs.
Improved sign in and registration pages.
Project access control and workflow configuration integrated to project settings.
Added check and highlighter for i18next interpolation and nesting.
Added check and highlighter for percent placeholders.
Record source string changes in history.
Upgraded Microsoft Translator to version 3 API.
Reimplemented translation memory backend.
Added support for several is: lookups in API.
Allow to make lookups avoid internal blacklist.
Improved comments extraction from monolingual po files.
Renamed whiteboard messages to announcements.
Fixed occasional problems with registration mails.
Fixed editing monolingual XLIFF source file.
Added support for exact matching in API.
Add support for source upload on bilingual translations.
Added support for intermediate language from developers.
Added support for source strings review.
Extended download options for platform wide translation memory.

**Weblate 3.x series**

**Weblate 3.11.3**

2020  3  11

Fixed searching for fields with certain priority.
Fixed predefined query for recently added strings.
Fixed searching returning duplicate matches.
Gmail
Fixed reverting changes from the history.
Added links to events in digest notifications.
Fixed email for account removal confirmation.
Added support for Slack authentication in Docker container.
Avoid sending notifications for not subscribed languages.
Include Celery queues in performance overview.

Raised bleach dependency to address CVE-2020-6802.
Fixed listing project level changes in history.
Fixed stats invalidation in some corner cases.
Fixed searching for certain string states.
Improved format string checks behavior on missing percent.
Fixed authentication using some third party providers.

**Weblate 3.11.2**

2020 2 22  

Fixed some strings wrongly reported as having no words.

**Weblate 3.11.1**

2020 2 20  

Documented Celery setup changes.
Improved filename validation on component creation.
Fixed minimal versions of some dependencies.
Fixed adding groups with certain Django versions.
Fixed manual pushing to upstream repository.
Improved glossary matching.

**Weblate 3.11**

2020 2 17  

Allow using VCS push URL during component creation via API.
Fixed links in notifications e-mails.
Improved look of plaintext e-mails.
Display ignored checks and allow to make them active again.

Recommend upgrade to new Weblate versions in the system checks.
Provide more detailed analysis for duplicate language alert.
Include more detailed license info on the project pages.
Automatically unshallow local copies if needed.
Fixed download of strings needing action.
New alert to warn about using the same file mask twice.
Improve XML placeables extraction.
The `SINGLE_PROJECT` can now enforce redirection to chosen project.
Added option to resolve comments.
Added bulk editing of flags.
Added support for labels.

Added option for...

Increased default validity of confirmation links.

Improved Matomo integration.

Fixed to correctly handle source string change.

Extended automatic updates configuration by AUTO_UPDATE.

LINGUAS

Weblate 3.10.3

2020 1 18

Support for translate-toolkit 2.5.0.

Weblate 3.10.2

2020 1 18

Add lock indication to projects.

Fixed CSS bug causing flickering in some web browsers.

Improved repository matching for GitHub and Bitbucket hooks.

Fixed data migration on some Python 2.7 installations.

Allow configuration of Git shallow cloning.

Improved background notification processing.

Fixed broken form submission when navigating back in web browser.

New add-on to configure YAML formatting.

Fixed same plurals check to not fire on single plural form languages.

Fixed regex search on some fields.

Weblate 3.10.1

2020 1 9

Extended API with translation creation.

Fixed several corner cases in data migrations.

Compatibility with Django 3.0.

Added support for customizable security.txt.

Improved breadcrumbs in changelog.

Improved translations listing on dashboard.

Improved HTTP responses for webhooks.

Added support for GitLab merge requests in Docker container.
**Weblate 3.10**

2019 12 20

Improved application user interface.
Added doublespace check.
Fixed creating new languages.
Avoid sending auditlog notifications to deleted e-mails.

Added support for Markdown in comments.
Allow placing translation instruction text in project info.

Improved support for Mercurial.
Improved Git repository fetching performance.
Add search lookup for age of string.

Show context for nearby strings.
Added support for notifications on repository operations.
Improved translation listings.
Extended search capabilities.
Added support for automatic translation strings marked for editing.
Avoid sending duplicate notifications for linked component alerts.
Improve default merge request message.
Better indicate string state in Zen mode.
Added support for more languages in Yandex Translate.
Improved look of notification e-mails.
Provide choice for translation license.

**Weblate 3.9.1**

2019 10 28

Remove some unneeded files from backups.
Fixed potential crash in reports.
Fixed cross database migration failure.
Added support for force pushing Git repositories.
Reduced risk of registration token invalidation.

Added search based on priority.
Fixed possible crash on adding strings to JSON file.
Safe HTML check and fixup now honor source string markup.
Avoid sending notifications to invited and deleted users.
Fix SSL connection to redis in Celery in Docker container.
Weblate 3.9

2019 10 15

Include Weblate metadata in downloaded files.
Indicate missing strings in format checks.
Separate check for French punctuation spacing.
Add support for fixing some of quality checks errors.
Add separate permission to create new projects.
Extend stats for char counts.
Improve support for Java style language codes.
Added new generic check for placeholders.
Added support for WebExtension JSON placeholders.
Added support for flat XML format.
Extended API with project, component and translation removal and creation.
Added support for Gitea and Gitee webhooks.
Added new custom regex based check.
Allow to configure contributing to shared translation memory.
Added ZIP download for more translation files.
Make XLIFF standard compliant parsing of maxwidth and font.
Added new check and fixer for safe HTML markup for translating web applications.
Add component alert on unsupported configuration.
Display add-on parameters on overview.
Sentry is now supported through modern Sentry SDK instead of Raven.
Changed example settings to be better fit for production environment.
Added automated backups using BorgBackup.
Split cleanup add-on for RESX to avoid unwanted file updates.
Added advanced search capabilities.
Allow users to download their own reports.
Added localization guide to help configuring components.
Added support for GitLab merge requests.
Improved display of repository status.
Perform automated translation in the background.

Weblate 3.8

2019 8 15

Added support for simplified creating of similar components.
Added support for parsing translation flags from the XML based file formats.
Log exceptions into Celery log.
Improved look of notification e-mails.
Improved performance on most of translation pages.
Fixed listing of languages not known to Weblate.
Add support for replacing file content with uploaded.
Add support for translating non VCS based content.
Added OpenGraph widget image to use on social networks.
Added support for animated screenshots.
Improved handling of monolingual XLIFF files.
Avoid sending multiple notifications for single event.
Add support for filtering changes.
Extended predefined periods for reporting.
Added webhook support for Azure Repos.
New opt-in notifications on pending suggestions or untranslated strings.
Add one click unsubscribe link to notification e-mails.
Fixed false positives with Has been translated check.
New management interface for admins.
String priority can now be specified using flags.
Added language management views.
Add checks for Qt library and Ruby format strings.
Added configuration to better fit single project installations.
Notify about new string on source string change on monolingual translations.
Added separate view for translation memory with search capability.

**Weblate 3.7.1**

2019 6 28

Documentation updates.
Fixed some requirements constraints.
Updated language database.
Localization updates.
Various user interface tweaks.
Improved handling of unsupported but discovered translation files.
More verbosely report missing file format requirements.

**Weblate 3.7**

2019 6 21

Added separate Celery queue for notifications.
Use consistent look with application for API browsing.
Include approved stats in the reports.
Report progress when updating translation component.
Allow to abort running background component update.
Extend template language for filename manipulations.
Use templates for editor link and repository browser URL.
Indicate max length and current characters count when editing translation.
Refreshed landing page for new contributors.
msgmerge

Delay opening SMTP connection when sending notifications.
Improved error logging.
Allow custom location in MO generating add-on.
Added add-ons to cleanup old suggestions or comments.
Added option to enable horizontal mode in the Zen editor.
Improved import performance with many linked components.
Fixed examples installation in some cases.

Added new horizontal stats widget.
Improved format strings check on plurals.
Added font management tool.

Added support for subtitle formats.
Include overall completion stats for languages.
Added reporting at project and global scope.
Improved user interface when showing translation status.
New Weblate logo and color scheme.
New look of bitmap badges.

**Weblate 3.6.1**

2019 4 26

Improved handling of monolingual XLIFF files.
Fixed digest notifications in some corner cases.
Fixed add-on script error alert.
Fixed generating MO file for monolingual PO files.
Fixed display of uninstalled checks.
Indicate administered projects on project listing.
Allow update to recover from missing VCS repository.

**Weblate 3.6**

2019 4 20

Add support for downloading user data.

Cleanup add-on is now compatible with app store metadata translations.
Configurable language code syntax when adding new translations.
Warn about using Python 2 with planned termination of support in April 2020.
Extract special characters from the source string for visual keyboard.
Extended contributor stats to reflect both source and target counts.
Admins and consistency add-ons can now add translations even if disabled for users.
Fixed description of toggle disabling `Language-Team` header manipulation.
Notify users mentioned in comments.
Removed file format autodetection from component setup.
Fixed generating MO file for monolingual PO files.
Added digest notifications.
Added support for muting component notifications.
Added notifications for new alerts, whiteboard messages or components.
Notifications for administered projects can now be configured.
Improved handling of three letter language codes.

**Weblate 3.5.1**

2019 3 10

Fixed Celery systemd unit example.
Fixed notifications from HTTP repositories with login.
Fixed race condition in editing source string for monolingual translations.
Include output of failed add-on execution in the logs.
Improved validation of choices for adding new language.
Allow to edit file format in component settings.
Update installation instructions to prefer Python 3.
Performance and consistency improvements for loading translations.
Microsoft Terminology service Zeep

Localization updates.

**Weblate 3.5**

2019 3 3

Improved performance of built-in translation memory.
Added interface to manage global translation memory.
Improved alerting on bad component state.
Added user interface to manage whiteboard messages.
Add-on commit message now can be configured.
Reduce number of commits when updating upstream repository.
Fixed possible metadata loss when moving component between projects.
Improved navigation in the Zen mode.
Added several new quality checks (Markdown related and URL).
Added support for app store metadata files.
Added support for toggling GitHub or Gerrit integration.
Kashida

Added option to squash commits based on authors.
Improved support for XLSX file format.
Compatibility with Tesseract 4.0.
Billing add-on now removes projects for unpaid billings after 45 days.

**Weblate 3.4**

2019 1 22

Added support for XLIFF placeholders.
Celery can now utilize multiple task queues.
Added support for renaming and moving projects and components.
Include characters counts in reports.
Added guided adding of translation components with automatic detection of translation files.
Customizable merge commit messages for Git.
Added visual indication of component alerts in navigation.
Improved performance of loading translation files.  
Changed default merge style to rebase and made that configurable.  
Better handle private use subtags in language code.  
Improved performance of fulltext index updates.  
Extended file upload API to support more parameters.

**Weblate 3.3**

2018 11 30

Added support for component and project removal.  
Improved performance for some monolingual translations.  
Added translation component alerts to highlight problems with a translation.  
Expose XLIFF string resname as context when available.  
Added support for XLIFF states.  
Added check for non writable files in DATA_DIR.  
Improved CSV export for changes.

**Weblate 3.2.2**

2018 10 20

Remove no longer needed Babel dependency.  
Updated language definitions.  
Improve documentation for add-ons, LDAP and Celery.  
Fixed enabling new dos-eol and auto-java-messageformat flags.  
Fixed running setup.py test from PyPI package.  
Improved plurals handling.  
Fixed translation upload API failure in some corner cases.  
Fixed updating Git configuration in case it was changed manually.

**Weblate 3.2.1**

2018 10 10

Document dependency on backports.csv on Python 2.7.  
Fix running tests under root.  
Improved error handling in gitexport module.  
Fixed progress reporting for newly added languages.  
Correctly report Celery worker errors to Sentry.  
Fixed creating new translations with Qt Linguist.  
Fixed occasional fulltext index update failures.  
Improved validation when creating new components.  
Added support for cleanup of old suggestions.
**Weblate 3.2**

2018 10 6

Add install_addon management command for automated add-on installation.

Added support for export and import of Excel files.

Improve component cleanup in case of multiple component discovery add-ons.

Weblate now uses Celery to offload some processing.

Improved search capabilities and added regular expression search.

Added support for Youdao Zhiyun API machine translation.

Added support for Baidu API machine translation.

Integrated maintenance and cleanup tasks using Celery.

Improved performance of loading translations by almost 25%.

Removed support for merging headers on upload.

Removed support for custom commit messages.

Configurable editing mode (zen/full).

Added support for error reporting to Sentry.

Added support for automated daily update of repositories.

Added support for creating projects and components by users.

Users and projects can import their existing translation memories.

Better management of related strings for screenshots.

Added support for checking Java MessageFormat.

See 3.2 milestone on GitHub for detailed list of addressed issues.

**Weblate 3.1.1**

2018 7 27

Fix testsuite failure on some setups.

**Weblate 3.1**

2018 7 27

Upgrades from older version than 3.0.1 are not supported.

Allow to override default commit messages from settings.

Improve webhooks compatibility with self hosted environments.

Added support for Amazon Translate.

Compatibility with Django 2.1.

Django system checks are now used to diagnose problems with installation.

Removed support for soon shutdown libravatar service.

Improved calculation of string similarity in translation memory matches.

Added support by signing Git commits by GnuPG.
Weblate 3.0.1

2018 6 10 10

Fixed possible migration issue from 2.20.
Localization updates.
Removed obsolete hook examples.

Weblate 3.0

2018 6 1 15

Several code cleanups that lead to moved and renamed modules.
import_project
Windows RC
PO

The per component hook scripts are removed, use add-ons instead.

Add support for providing additional textual context.

Weblate 2.x

Weblate 2.20

2018 4 4 15

Subversion

New add-on to configure gettext output wrapping.
New add-on to configure JSON formatting.

Added support for authentication in API using RFC 6750 compatible Bearer authentication.
Added support for automatic translation using machine translation services.
Added support for HTML markup in whiteboard messages.
Added support for mass changing state of strings.
Translate-toolkit at least 2.3.0 is now required, older versions are no longer supported.

Added support for DeepL machine translation service.
Machine translation results are now cached inside Weblate.
**Weblate 2.19.1**

2018 2 20

Fixed migration issue on upgrade from 2.18.
Improved file upload API validation.

**Weblate 2.19**

2018 2 15

Fixed imports across some file formats.
Display human friendly browser information in audit log.
Added TMX exporter for files.
Various performance improvements for loading translation files.
Added option to disable access management in Weblate in favor of Django one.
Improved glossary lookup speed for large strings.
Compatibility with django_auth_ldap 1.3.0.
Configuration errors are now stored and reported persistently.
Honor ignore flags in whitespace autofixer.
Improved compatibility with some Subversion setups.

Added support for SAP Translation Hub service.
Microsoft Terminology service

Removed support for advertisement in notification e-mails.
Improved translation progress reporting at language level.
Improved support for different plural formulas.
Added support for Subversion repositories not using stdlayout.

**Weblate 2.18**

2017 12 15

Extended contributor stats.
Improved configuration of special characters virtual keyboard.
Added support for DTD file format.
Changed keyboard shortcuts to less likely collide with browser/system ones.
Improved support for approved flag in XLIFF files.
Added support for not wrapping long strings in gettext PO files.
Added button to copy permalink for current translation.
Dropped support for Django 1.10 and added support for Django 2.0.
Removed locking of translations while translating.
Added support for adding new strings to monolingual translations.
Added support for translation workflows with dedicated reviewers.
Weblate 2.17.1
2017 10 13
Fixed running testsuite in some specific situations.

Weblate 2.17
2017 10 13
Weblate by default does shallow Git clones now.
Improved performance when updating large translation files.
Added support for blocking certain e-mails from registration.
Users can now delete their own comments.
Added preview step to search and replace feature.
Client side persistence of settings in search and upload forms.
Extended search capabilities.
More fine grained per project ACL configuration.
Default value of BASE_DIR has been changed.
Added two step account removal to prevent accidental removal.
Project access control settings is now editable.
Added optional spam protection for suggestions using Akismet.

Weblate 2.16
2017 8 11
Various performance improvements.
Added support for WebExtension JSON format.
Fixed git exporter authentication.
Improved CSV import in certain situations.
Improved look of Other translations widget.
The max-length checks is now enforcing length of text in form.
Make the commit_pending age configurable per component.
Various user interface cleanups.
Fixed component/project/site wide search for translations.

Weblate 2.15
2017 6 30
Show more related translations in other translations.
Add option to see translations of current string to other languages.
Use 4 plural forms for Lithuanian by default.
Fixed upload for monolingual files of different format.
Improved error messages on failed authentication.
Keep page state when removing word from glossary.
Added Perl format quality check.
Added support for rejecting reused passwords.
Extended toolbar for editing RTL languages.

**Weblate 2.14.1**

2017 5 24

Fixed possible error when paginating search results.
Fixed migrations from older versions in some corner cases.
Fixed possible CSRF on project watch and unwatch.
The password reset no longer authenticates user.
Fixed possible CAPTCHA bypass on forgotten password.

**Weblate 2.14**

2017 5 17

Add glossary entries using AJAX.
The logout now uses POST to avoid CSRF.
The API key token reset now uses POST to avoid CSRF.
Weblate sets Content-Security-Policy by default.
The local editor URL is validated to avoid self-XSS.
The password is now validated against common flaws by default.
Notify users about important activity with their account such as password change.
The CSV exports now escape potential formulas.
Various minor improvements in security.

Suggestion content is stored in the history.
Store important account activity in audit log.
Ask for password confirmation when removing account or adding new associations.
Show time when suggestion has been made.
There is new quality check for trailing semicolon.
Ensure that search links can be shared.
Included source string information and screenshots in the API.
Allow to overwrite translations through API upload.

**Weblate 2.13.1**

2017 4 12

Fixed listing of managed projects in profile.
Fixed migration issue where some permissions were missing.
Fixed listing of current file format in translation download.
Return HTTP 404 when trying to access project where user lacks privileges.
Weblate 2.13

2017 4 12

Fixed quality checks on translation templates.
Added quality check to trigger on losing translation.
Add option to view pending suggestions from user.

Default dashboard for unauthenticated users can be configured.
Add option to browse 25 random strings for review.
History now indicates string change.
Better error reporting when adding new translation.
Added per language search within project.
Group ACLs can now be limited to certain permissions.
The per project ACLs are now implemented using Group ACL.
Added more fine grained privileges control.
Various minor UI improvements.

Weblate 2.12

2017 3 3

Improved admin interface for groups.
Added support for Yandex Translate API.
Improved speed of site wide search.
Added project and component wide search.
Added project and component wide search and replace.

Added support for opening source files in local editor.
Added support for configuring visual keyboard with special characters.
Improved screenshot management with OCR support for matching source strings.
Default commit message now includes translation information and URL.
Added support for Joomla translation format.
Improved reliability of import across file formats.

Weblate 2.11

2017 1 31

Include language detailed information on language page.
Mercurial backend improvements.
Added option to specify translation component priority.
More consistent usage of Group ACL even with less used permissions.
Added WL_BRANCH variable to hook scripts.
Improved developer documentation.
Better compatibility with various Git versions in Git exporter add-on.
Included per project and component stats.
Added language code mapping for better support of Microsoft Translate API.
Moved fulltext cleanup to background job to make translation removal faster.
Fixed displaying of plural source for languages with single plural form.
Improved error handling in import_project.
Various performance improvements.

**Weblate 2.10.1**

2017 1 20

Do not leak account existence on password reset form (CVE-2017-5537).

**Weblate 2.10**

2016 12 15

Added quality check to check whether plurals are translated differently.
Fixed GitHub hooks for repositories with authentication.
Added optional Git exporter module.
Support for Microsoft Cognitive Services Translator API.
Simplified project and component user interface.
Added automatic fix to remove control characters.
Added per language overview to project.
Added support for CSV export.
Added CSV download for stats.
Added basic API for changes and strings.
Added support for Apertium APy server for machine translations.

**Weblate 2.9**

2016 11 4

Extended parameters for createadmin management command.
Extended import_json to be able to handle with existing components.
Added support for YAML files.
Project owners can now configure translation component and project details.
Use "Watched" instead of "Subscribed" projects.
Projects can be watched directly from project page.
Added multi language status widget.
Record suggestion deletion in history.
Improved UX of languages selection in profile.
Fixed showing whiteboard messages for component.
Show source string comment more prominently.
Automatically install Gettext PO merge driver for Git repositories.
Added search and replace feature.
Weblate 2.8

2016 8 31

Translations.
Updated bundled JavaScript libraries.
Added list_translators management command.
Django 1.8 is no longer supported.
Fixed compatibility with Django 1.10.
Added Subversion support.
Separated XML validity check from XML mismatched tags.
Fixed API to honor HIDE_REPO_CREDENTIALS settings.
Show source change in Zen mode.
Alt+PageUp/PageDown/Home/End now works in Zen mode as well.
Add tooltip showing exact time of changes.
Add option to select filters and search from translation page.
Added UI for translation removal.
Improved behavior when inserting placeables.
Fixed auto locking issues in Zen mode.

Weblate 2.7

2016 7 10

Removed Google web translate machine translation.
Improved commit message when adding translation.
Fixed Google Translate API for Hebrew language.
Compatibility with Mercurial 3.8.
Added import_json management command.
Correct ordering of listed translations.
Show full suggestion text, not only a diff.
Extend API (detailed repository status, statistics, …).
Testsuite no longer requires network access to test repositories.

Weblate 2.6

2016 4 28

Fixed validation of components with language filter.
Improved support for XLIFF files.
Fixed machine translation for non English sources.
Added REST API.
Django 1.10 compatibility.
Added categories to whiteboard messages.
Weblate 2.5

2016 3 10

Fixed automatic translation for project owners.
Improved performance of commit and push operations.

Added support for merging comments on file upload.
Added support for some GNU extensions to C printf format.

Added support for generating translator credits.
Added support for generating contributor stats.
Site wide search can search only in one language.
Improve quality checks for Armenian.
Support for starting translation components without existing translations.
Support for adding new translations in Qt TS.
Improved support for translating PHP files.
Performance improvements for quality checks.

Improved support for XLIFF files.
Extended list of options for import_project.
Improved targeting for whiteboard messages.
Support for automatic translation across projects.
Optimized fulltext search index.
Added management command for auto translation.
Added placeables highlighting.
Added keyboard shortcuts for placeables, checks and machine translations.
Improved translation locking.
Added quality check for AngularJS interpolation.
Added extensive group based ACLs.

Support for Python 3.
Dropped support for Django 1.7.
Dropped dependency on msginit for creating new gettext PO files.
Added configurable dashboard views.
Improved notifications on parse errors.
Added option to import components with duplicate name to import_project.
Improved support for translating PHP files.
Added XLIFF export for dictionary.
Added XLIFF and gettext PO export for all translations.

Added support for configurable automatic group assignments.
Improved adding of new translations.
Weblate 2.4

2015 9 20

Improved support for PHP files.
Ability to add ACL to anonymous user.
Improved configurability of import_project command.
Added CSV dump of history.
Avoid copy/paste errors with whitespace characters.
Added support for Bitbucket webhooks.
Tighter control on fuzzy strings on translation upload.
Several URLs have changed, you might have to update your bookmarks.
Hook scripts are executed with VCS root as current directory.
Hook scripts are executed with environment variables describing current component.
Add management command to optimize fulltext index.
Added support for error reporting to Rollbar.
Projects now can have multiple owners.
Project owners can manage themselves.
Added support for javascript-format used in gettext PO.
Support for adding new translations in XLIFF.
Improved file format autodetection.
Extended keyboard shortcuts.
Improved dictionary matching for several languages.
Improved layout of most of pages.
Support for adding words to dictionary while translating.
Added support for filtering languages to be managed by Weblate.
Added support for translating and importing CSV files.
Rewritten handling of static files.
Direct login/registration links to third-party service if that’s the only one.
Commit pending changes on account removal.
Add management command to change site name.
Add option to configure default committer.
Add hook after adding new translation.
Add option to specify multiple files to add to commit.

Weblate 2.3

2015 5 22

Dropped support for Django 1.6 and South migrations.
Support for adding new translations when using Java Property files.
Allow to accept suggestion without editing.
Improved support for Google OAuth 2.0.
Added support for Microsoft .resx files.
Tuned default robots.txt to disallow big crawling of translations.
Simplified workflow for accepting suggestions.
Added project owners who always receive important notifications.
Allow to disable editing of monolingual template.
More detailed repository status view.
Direct link for editing template when changing translation.
Allow to add more permissions to project owners.

Weblate 2.2
2015 2 19
Fulltext search on location and comments fields.
New SVG/JavaScript-based activity charts.
Support for Django 1.8.
Support for deleting comments.
Added own SVG badge.
Added support for Google Analytics.
Improved handling of translation filenames.
Added support for monolingual JSON translations.
Record component locking in a history.
Support for editing source (template) language for monolingual translations.
Added basic support for Gerrit.

Weblate 2.1
2014 12 5
Added support for Mercurial repositories.
Replaced Glyphicon font by Awesome.
Added icons for social authentication services.
Better consistency of button colors and icons.
Various bugfixes.
Automatic hiding of columns in translation listing for small screens.
Changed configuration of filesystem paths.
Improved SSH keys handling and storage.
Improved repository locking.
Customizable quality checks per source string.
Allow to hide completed translations from dashboard.

Weblate 2.0
2014 11 6
New responsive UI using Bootstrap.
Rewritten VCS backend.
Added whiteboard for site wide messages.
Configurable strings priority.
Added support for JSON file format.
Fixed generating mo files in certain cases.
Added support for GitLab notifications.
Added support for disabling translation suggestions.
Django 1.7 support.
ACL projects now have user management.
Extended search possibilities.
Give more hints to translators about plurals.
Fixed Git repository locking.
Compatibility with older Git versions.
Improved ACL support.
Added buttons for per language quotes and other special characters.
Support for exporting stats as JSONP.

**Weblate 1.x [Link]**

**Weblate 1.9**

2014 5 6
Django 1.6 compatibility.
No longer maintained compatibility with Django 1.4.
Management commands for locking/unlocking translations.
Improved support for Qt TS files.
Users can now delete their account.
Avatars can be disabled.
Merged first and last name attributes.
Avatars are now fetched and cached server side.
Added support for shields.io badge.

**Weblate 1.8**

2013 11 7
Please check manual for upgrade instructions.
Nicer listing of project summary.
Better visible options for sharing.
More control over anonymous users privileges.
Supports login using third party services, check manual for more details.
Users can login by e-mail instead of username.
Improved source strings review.
Searching across all strings.
Better tracking of source strings.
Captcha protection for registration.
Weblate 1.7

2013 10 7

Please check manual for upgrade instructions.
Support for checking Python brace format string.
Per component customization of quality checks.
Detailed per translation stats.
Changed way of linking suggestions, checks and comments to strings.
Users can now add text to commit message.
Support for subscribing on new language requests.
Support for adding new translations.

Pango + Cairo Pillow

Add status badge widget.

Changes in dictionary are now logged in history.

Weblate 1.6

2013 7 25

Nicer error handling on registration.
Browsing of changes.
Fixed sorting of machine translation suggestions.
Improved support for MyMemory machine translation.
Added support for Amagama machine translation.
Various optimizations on frequently used pages.
Highlights searched phrase in search results.
Support for automatic fixups while saving the message.
Tracking of translation history and option to revert it.
Added support for Google Translate API.
Added support for managing SSH host keys.
Various form validation improvements.
Various quality checks improvements.
Performance improvements for import.
Added support for voting on suggestions.
Cleanup of admin interface.

Weblate 1.5

2013 4 16

Please check manual for upgrade instructions.
Added public user pages.
Better naming of plural forms.
Added support for TBX export of glossary.
Added support for Bitbucket notifications.
Activity charts are now available for each translation, language or user.
Extended options of import_project admin command.
Compatible with Django 1.5.
Avatars are now shown using libravatar.
Added possibility to pretty print JSON export.
Various performance improvements.

- Added support for custom pre-commit hooks and committing additional files.
- Rewritten search for better performance and user experience.
- New interface for machine translations.
- Added support for monolingual po files.
- Extend amount of cached metadata to improve speed of various searches.
- Now shows word counts as well.

**Weblate 1.4**

2013 1 23

- Fixed deleting of checks/comments on string deletion.
- Added option to disable automatic propagation of translations.
- Added option to subscribe for merge failures.
- Correctly import on projects which needs custom ttkit loader.
- Added sitemaps to allow easier access by crawlers.
- Provide direct links to string in notification e-mails or feeds.
- Various improvements to admin interface.
- Provide hints for production setup in admin interface.
- Added per language widgets and engage page.
- Improved translation locking handling.

- More options for formatting commit message.
- Fixed error handling with machine translation services.
- Improved automatic translation locking behaviour.
- Support for showing changes from previous source string.
- Added support for substring search.
- Various quality checks improvements.
- Support for per project ACL.
- Basic code coverage by unit tests.

**Weblate 1.3**

2012 11 16

- Compatibility with PostgreSQL database backend.
- Removes languages removed in upstream git repository.
- Improved quality checks processing.

- Support for optional rebasing instead of merge.
- Possibility to relocate Weblate (for example to run it under /weblate path).
- Support for manually choosing file type in case autodetection fails.
- Better support for Android resources.
- Support for generating SSH key from web interface.
More visible data exports.

New buttons to enter some special characters.

Support for exporting dictionary.

Support for locking down whole Weblate installation.

Checks for source strings and support for source strings review.

Support for user comments for both translations and source strings.

Better changes log tracking.

Changes can now be monitored using RSS.

Improved support for RTL languages.

**Weblate 1.2**

2012 8 14

Weblate now uses South for database migration, please check upgrade instructions if you are upgrading.

Fixed minor issues with linked git repos.

New introduction page for engaging people with translating using Weblate.

Added widgets which can be used for promoting translation projects.

Added option to reset repository to origin (for privileged users).

Project or component can now be locked for translations.

Possibility to disable some translations.

Configurable options for adding new translations.

Configuration of git commits per project.

Better layout of main page.

Support for automatically pushing changes on every commit.

Support for e-mail notifications of translators.

Improved handling of not known languages when importing project.

Support for locking translation by translator.

Optionally maintain Language-Team header in po file.

Include some statistics in about page.

Supports (and requires) django-registration 0.8.

Checking of requirements during setup.

**Weblate 1.1**

2012 7 4

Improved several translations.

Better validation while creating component.

Added support for shared git repositories across components.

Do not necessary commit on every attempt to pull remote repo.

Added support for offloading indexing.
Weblate 1.0

2012 5 10
Improved validation while adding/saving component.
Experimental support for Android component files (needs patched ttkit).
Updates from hooks are run in background.
Improved installation instructions.
Improved navigation in dictionary.

Weblate 0.x series

Weblate 0.9

2012 4 18
Fixed import of unknown languages.
Improved listing of nearby messages.
Improved several checks.
Documentation updates.
Added definition for several more languages.
Various code cleanups.

Changed file layout.
Update helper scripts to Django 1.4.
Improved navigation while translating.
Better handling of po file renames.
Better validation while creating component.
Integrated full setup into syncdb.
Added list of recent changes to all translation pages.
Check for untranslated strings ignores format string only messages.

Weblate 0.8

2012 4 3
Replaced own full text search with Whoosh.
Various fixes and improvements to checks.
New command updatechecks.
Lot of translation updates.
Added dictionary for storing most frequently used terms.
Added /admin/report/ for overview of repositories status.
Machine translation services no longer block page loading.
Management interface now contains also useful actions to update data.
Records log of changes made by users.
Ability to postpone commit to Git to generate less commits from single user.

Automatic translation using already translated strings.
New about page showing used versions.
Django 1.4 compatibility.
Ability to push changes to remote repo from web interface.
Added review of translations done by others.

**Weblate 0.7**

2012 2 16

Direct support for GitHub notifications.
Added support for cleaning up orphaned checks and translations.
Displays nearby strings while translating.
Displays similar strings while translating.
Improved searching for string.

**Weblate 0.6**

2012 2 14

Added various checks for translated messages.
Tunable access control.
Improved handling of translations with new lines.
Added client side sorting of tables.
Please check upgrading instructions in case you are upgrading.

**Weblate 0.5**

2012 2 12

Support for machine translation using following online services:
- Apertium
- Microsoft Translator
- MyMemory
Several new translations.
Improved merging of upstream changes.
Better handle concurrent git pull and translation.
Propagating works for fuzzy changes as well.
Propagating works also for file upload.
Fixed file downloads while using FastCGI (and possibly others).

**Weblate 0.4**

2012 2 8

CSRF API
Weblate 0.3

2012  2  8  8

Sphinx

Weblate 0.2

2012  2  7  8

URL

Weblate 0.1

2012  2  6  8
w
wlc.??
wlc.config.??
wlc.main.??
ANY /.*\/api

GET /api/.*\?\?\api/addons
GET /api/addons/.*\?\?\nGET /api/addons/(int:id)/.*\?\?\nPUT /api/addons/(int:id)/.*\?\?\nDELETE /api/addons/(int:id)/.*\?\?\nPATCH /api/addons/(int:id)/.*\?\?\api/changes
GET /api/changes/.*\?\?\nGET /api/changes/(int:id)/.*\?\?\api/component-lists
GET /api/components/.*\?\?\nGET /api/components/(string:project)/(string:component)/.*\?\?\nGET /api/components/(string:project)/(string:component)/changes/.*\?\?\nGET /api/components/(string:project)/(string:component)/file/.*\?\?\nGET /api/components/(string:project)/(string:component)/links/.*\?\?\nGET /api/components/(string:project)/(string:component)/lock/.*\?\?\nGET /api/components/(string:project)/(string:component)/monolingual_base/.*\?\?\nGET /api/components/(string:project)/(string:component)/new_template/.*\?\?\nGET /api/components/(string:project)/(string:component)/repository/.*\?\?\nGET /api/components/(string:project)/(string:component)/screenshots/.*\?\?\nGET /api/components/(string:project)/(string:component)/statistics/.*\?\?\nGET /api/components/(string:project)/(string:component)/translations/.*\?\?\nPOST /api/components/(string:project)/(string:component)/addons/.*\?\?\nPOST /api/components/(string:project)/(string:component)/links/.*\?\?\nPOST /api/components/(string:project)/(string:component)/lock/.*\?\?\nPOST /api/components/(string:project)/(string:component)/repository/.*\?\?\nPOST /api/components/(string:project)/(string:component)/translations/.*\?\?\nPUT /api/components/(string:project)/(string:component)/.*\?\?\nDELETE /api/components/(string:project)/(string:component)/.*\?\?\DELETE /api/components/(string:project)/(string:component)/links/(string:project_slug)/.*\?\?\nPATCH /api/components/(string:project)/(string:component)/.*\?\?\api/groups
GET /api/groups/.*\?\?\nGET /api/groups/(int:id)/.*\?\?\nPOST /api/groups/.*\?\?\nPOST /api/groups/(int:id)/componentlists/.*\?\?\nPOST /api/groups/(int:id)/components/.*\?\?\nPOST /api/groups/(int:id)/languages/.*\?\?\nPOST /api/groups/(int:id)/projects/.*\?\?400
GET /api/tasks/,
GET /api/tasks/(str:uuid)/, /api/translations
GET /api/translations/,
GET /api/translations/(string:project)/(string:component)/(string:language)/,
GET /api/translations/(string:project)/(string:component)/(string:language)/changes/,
GET /api/translations/(string:project)/(string:component)/(string:language)/file/,
GET /api/translations/(string:project)/(string:component)/(string:language)/repository/,
GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/,
GET /api/translations/(string:project)/(string:component)/(string:language)/units/,
POST /api/translations/(string:project)/(string:component)/(string:language)/autotranslate/,
POST /api/translations/(string:project)/(string:component)/(string:language)/file/,
POST /api/translations/(string:project)/(string:component)/(string:language)/repository/,
POST /api/translations/(string:project)/(string:component)/(string:language)/units/,
DELETE /api/translations/(string:project)/(string:component)/(string:language)/,
/api/units
GET /api/units/,
GET /api/units/(int:id)/,
PUT /api/units/(int:id)/,
DELETE /api/units/(int:id)/,
PATCH /api/units/(int:id)/, /api/users
GET /api/users/,
GET /api/users/(str:username)/,
GET /api/users/(str:username)/notifications/,
GET /api/users/(str:username)/notifications/(int:subscription_id)/,
GET /api/users/(str:username)/statistics/,
POST /api/users/,
POST /api/users/(str:username)/groups/,
POST /api/users/(str:username)/notifications/,
PUT /api/users/(str:username)/,
PUT /api/users/(str:username)/notifications/(int:subscription_id)/,
DELETE /api/users/(str:username)/,
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/,
PATCH /api/users/(str:username)/,
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, /exports
GET /exports/rss/,
GET /exports/rss/(string:project)/,
GET /exports/rss/(string:project)/(string:component)/,
GET /exports/rss/(string:project)/(string:component)/(string:language)/,
GET /exports/rss/language/(string:language)/,
GET /exports/stats/(string:project)/(string:component)/,?? /hooks
GET /hooks/update/(string:project)/,??
GET /hooks/update/(string:project)/(string:component)/,??
POST /hooks/azure/,??
POST /hooks/bitbucket/,??
POST /hooks/gitea/,??
POST /hooks/gitlab/,??
POST /hooks/gitee/,??
POST /hooks/github/,??
POST /hooks/pagure/,??