
Weblate Documentation

Versión 0.5

Michal Čihař

20 de julio de 2020

1. About Weblate	3
1.1. Project goals	3
1.2. Project name	3
1.3. Project website	3
1.4. Authors	3
2. Usage guide	5
2.1. Registration	5
2.2. Profile information	5
2.3. Projects structure	5
2.4. Translation links	6
2.5. Translating	6
2.6. Suggestions	6
2.7. Machine translation	6
3. Changes	7
3.1. weblate 0.5	7
3.2. weblate 0.4	7
3.3. weblate 0.3	8
3.4. weblate 0.2	8
3.5. weblate 0.1	8
4. Installation instructions	9
4.1. Requirements	9
4.2. Installation	9
4.3. Running server	9
5. Configuration	11
6. Administration	13
6.1. Adding new resources	13
6.2. Project	13
6.3. Subproject	14
6.4. Updating repositories	14
6.5. Interacting with others	14
7. Management commands	15

8. License	17
9. Indices and tables	19
Índice	21

Contents:

1.1 Project goals

Minimalistic web based translation with direct commit to git on each translation made. There is no plan in heavy conflict resolution as these should be primarily handled on git side.

1.2 Project name

The project is named as mixture of words web and translate.

1.3 Project website

You can find project website at <http://weblate.org/>, there is also demonstration server at <http://demo.weblate.org/>. This documentation can be browsed on <http://weblate.readthedocs.org/>.

1.4 Authors

This tool was written by Michal Čihař michal@cihar.com.

This document briefly covers how to translate application using Weblate.

2.1 Registration

While everybody can browse projects, view translations or suggest them, only registered users are allowed to actually save changes and are credited for every translation made.

You can register following two simple steps:

1. Fill out the registration form with your credentials
2. Activate registration by following in email you receive
3. Possibly adjust your profile to choose which languages you know

2.2 Profile information

User profile contains your preferences, name and email. Name and email are being used in Git commits, so keep this information accurate.

In preferences, you can choose user interface language, languages which you prefer to translate (list of these will be offered to you on main page) and secondary languages, whose translations will be shown to you while translating.

2.3 Projects structure

Each project can contain various subprojects. The reason for this structure is that all subprojects in a project are expected to have a lot in common. Whenever translation is made in single subproject, it is automatically propagated to others within same project (this is especially useful when translating more version of same project).

2.4 Translation links

Once you navigate to translation, you will be shown set of links which lead to translation. These are results of various checks, like untranslated or fuzzy strings. Should no other checks fire, there will be still link to all translations. Alternatively you can use search field to find translation you need to fix.

2.5 Translating

On translate page, you are shown source string and edit area for translating. Should the translation be plural, multiple source strings and edit areas are shown, each described with label for plural form.

There are various extra information which can be shown on this page. Most of them are coming from the project source code (like context, comments or where the message is being used). When you configure secondary languages in your preferences, translation to these languages will be shown.

Bellow translation can be also shown suggestions from other users, which you can accept or delete.

2.6 Suggestions

As an anonymous user, you have no other choice than making a suggestion. However if you are logged in you can still decide to make only a suggestion instead of saving translation, for example in case you are unsure about the translation and you want somebody else to review it.

2.7 Machine translation

Based on configuration and your language, Weblate provides buttons for following machine translation tools.

2.7.1 MyMemory

Huge translation memory with machine translation.

Ver también:

<http://mymemory.translated.net/>

2.7.2 Apertium

A free/open-source machine translation platform providing translation to limited set of lanugages.

Ver también:

<http://www.apertium.org/>

2.7.3 Microsoft Translator

Machine translation service. Weblate is currently using deprecated v1 API, which might stop working in future.

Ver también:

<http://www.microsofttranslator.com/>

3.1 weblate 0.5

Released on February 12th 2012.

- **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).

3.2 weblate 0.4

Released on February 8th 2012.

- Added usage guide to documentation.
- Fixed API hooks not to require CSRF protection.

3.3 weblate 0.3

Released on February 8th 2012.

- Better display of source for plural translations.
- New documentation in Sphinx format.
- Displays secondary languages while translating.
- Improved error page to give list of existing projects.
- New per language stats.

3.4 weblate 0.2

Released on February 7th 2012.

- Improved validation of several forms.
- Warn users on profile upgrade.
- Remember URL for login.
- Naming of text areas while entering plural forms.
- Automatic expanding of translation area.

3.5 weblate 0.1

Released on February 6th 2012.

- Initial release.

Installation instructions

4.1 Requirements

Django <https://www.djangoproject.com/>

Translate-toolkit <http://translate.sourceforge.net/wiki/toolkit/index>

GitPython (>= 0.3) <http://gitorious.org/projects/git-python/>

Django-registration <https://bitbucket.org/ubernostrum/django-registration/>

4.2 Installation

Install all required components (see above), adjust `settings.py` and then run `./manage.py syncdb` to create database structure. Now you should be able to create translation projects using admin interface. You probably also want to run `./manage.py setuplang` to get default list of languages.

4.3 Running server

Running Weblate is not different from running any other Django based application.

Ver también:

<https://docs.djangoproject.com/en/1.3/howto/deployment/>

All settings are stored in `settings.py` (as usual for Django).

COMMIT_MESSAGE

Message used on each commit Weblate does.

ENABLE_HOOKS

Whether to enable anonymous remote hooks.

Ver también:

[Interacting with others](#)

GIT_ROOT

Path where Weblate will store cloned Git repositories. Defaults to `repos` subdirectory.

MT_APERTIUM_KEY

API key for Apertium Web Service, you can register at <http://api.apertium.org/register.jsp>

MT_MICROSOFT_KEY

API key for Microsoft Translator service, you can register at <http://www.bing.com/developers/createapp.aspx>

SITE_TITLE

Site title to be used in website and emails as well.

Ver también:

<https://docs.djangoproject.com/en/1.3/ref/settings/>

Administration of Weblate is done through standard Django admin interface, which is available under `/admin/` URL.

6.1 Adding new resources

All translation resources need to be available as Git repositories and are organized as project/subproject structure.

Weblate supports wide range of translation formats supported by translate toolkit, for example:

- GNU Gettext
- XLIFF
- Java properties
- Windows RC files
- Qt Linguist .ts
- Symbian localization files
- CSV
- INI

Ver también:

<http://translate.sourceforge.net/wiki/toolkit/formats>

6.2 Project

To add new resource to translate, you need to create translation project first. The project is sort of shelf, in which real translations are folded. All subprojects in same project share suggestions and dictionary, also the translations are automatically propagated through the all subproject in single project.

6.3 Subproject

Subproject is real resource for translating. You enter Git repository location and file mask which files to translate and Weblate automatically fetches the Git and finds all translated files.

Nota: As setup of translation project includes fetching Git repositories, you might want to preseed these, repos are stored in path defined by `GIT_ROOT` in `settings.py` in `<project>/<subproject>` directories.

6.4 Updating repositories

You should set up some way how backend repositories are updated from their source. You can either use hooks (see *Interacting with others*) or just regularly run `./manage.py updategit --all`.

With Gettext po files, you might be often bitten by conflict in PO file headers. To avoid it, you can use shipped merge driver (`scripts/git-merge-gettext-po`). To use it just put following configuration to your `.gitconfig`:

```
[merge "merge-gettext-po"]
  name = merge driver for gettext po files
  driver = /path/to/weblate/scripts/git-merge-gettext-po %O %A %B
```

And enable it's use by defining proper attributes in given repository (eg. in `.git/info/attribute`):

```
*.po merge=merge-gettext-po
```

Nota: This merge driver assumes the changes in POT files always are done in brach we're trying to merge.

Ver también:

<http://www.no-ack.org/2010/12/writing-git-merge-driver-for-po-files.html>

6.5 Interacting with others

You can trigger update of underlying git repository for every subproject by accessing URL `/hooks/p/project/subproject/update/`. This can be used for example as as Post-Receive URLs on Github.

Management commands

The `./manage.py` is extended with following commands:

checkgit

Prints current state of backend git repository.

You can either define which subproject to check (eg. `weblate/master`) or use `--all` to check all existing subprojects.

loadpo

Reloads translations from disk (eg. in case you did some updates in Git repository).

setuplang

Setups list of languages (it has own list and all defined in `translate-toolkit`).

updategit

Fetches remote Git repositories and updates internal cache.

You can either define which subproject to update (eg. `weblate/master`) or use `--all` to update all existing subprojects.

License

Copyright (C) 2012 Michal Čihař <michal@cihar.com>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

CAPÍTULO 9

Indices and tables

- genindex
- modindex
- search

Símbolos

`./manage.py` opción en línea de comandos
checkgit, 15
loadpo, 15
setuplang, 15
updategit, 15

C

checkgit
 `./manage.py` opción en línea de comandos, 15

G

GIT_ROOT, 14

L

loadpo
 `./manage.py` opción en línea de comandos, 15

S

setuplang
 `./manage.py` opción en línea de comandos, 15

U

updategit
 `./manage.py` opción en línea de comandos, 15

V

variables de entorno
 COMMIT_MESSAGE, 11
 ENABLE_HOOKS, 11
 GIT_ROOT, 11, 14
 MT_APERTIUM_KEY, 11
 MT_MICROSOFT_KEY, 11
 SITE_TITLE, 11