
Weblate Documentation

Versión 0.3

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. Changes	5
2.1. weblate 0.3	5
2.2. weblate 0.2	5
2.3. weblate 0.1	5
3. Installation instructions	7
3.1. Requirements	7
3.2. Installation	7
4. Configuration	9
5. Administration	11
5.1. Adding new resources	11
5.2. Project	11
5.3. Subproject	12
5.4. Interacting with others	12
6. Management commands	13
7. Planned features list	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.

2.1 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.

2.2 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.

2.3 weblate 0.1

Released on February 6th 2012.

- Initial release.

Installation instructions

3.1 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/>

3.2 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.

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.

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.

5.1 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>

5.2 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.

5.3 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.

5.4 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.

Planned features list

Possible features for future releases

- Auto translate using some online service
- **Checking of messages**
 - n at beginning/end
 - php-format
 - python-format
- Glossary of frequently used term (per project)
- **Cleanup of database**
 - not usable suggestions

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, 13
loadpo, 13
setuplang, 13
updategit, 13

C

checkgit
 `./manage.py` opción en línea de comandos, 13

G

GIT_ROOT, 12

L

loadpo
 `./manage.py` opción en línea de comandos, 13

S

setuplang
 `./manage.py` opción en línea de comandos, 13

U

updategit
 `./manage.py` opción en línea de comandos, 13

V

variables de entorno
 COMMIT_MESSAGE, 9
 ENABLE_HOOKS, 9
 GIT_ROOT, 9, 12
 SITE_TITLE, 9