
potranslator Documentation

Version 1.1.5

SekouD

nov. 01, 2018

Table des matières

1	potranslator	3
1.1	Langues supportées	3
1.2	Quick Start for auto-translation with potranslator	5
1.3	Fonctionnalités de base	6
1.4	Fonctionnalités optionnelles	6
1.5	Installation	7
1.6	Commandes, options, variables d'environnement	7
1.7	Licence	8
1.8	Original	8
1.9	CHANGEMENTS	8
2	Installation	9
2.1	Version stable	9
2.2	Depuis le code source	9
3	Utilisation	11
3.1	Depuis un programme Python	11
3.2	Commandes, options, et variables d'environnement	11
4	Documentation de l'API de potranslator	15
4.1	Référence de l'API pour les classes dans potranslator.potranslator.py	15
5	Contribuer	17
5.1	Types de contributions	17
5.2	Commencez !	18
5.3	Directives des Pull Request	19
5.4	Conseils	19
5.5	Déployer	19
6	Crédits	21
6.1	Responsable du développement	21
6.2	Contributeurs	21
7	Historique	23
7.1	1.1.0 (2018-07-08)	23
7.2	1.0.5 (2018-07-06)	23
7.3	1.0.0 (2018-07-05)	23

7.4 0.1.0 (2018-06-27)	23
8 Indices et tableaux	25
Index des modules Python	27

Contenu :

potranslator is a package to easily translate po and pot files generated by [Sphinx](#) or other tools in any language supported by Google Translate.

potranslator auto-detects the language in the original pot files and auto-translates the pot files into the supplied target languages.

The Command Line Interface of *potranslator* and its documentation are based on [sphinx-intl](#).

Facultatif : prend en charge le service collaboratif Transifex pour télécharger les traductions générées automatiquement vers [transifex](#) pour les corrections collaboratives des traductions.

1.1 Langues supportées

- Afrikaans af
- Albanais sq
- Amharic am
- Arabe ar
- Arménien hy
- Azerbaïdjanais az
- Basque eu
- Bélarusse be
- Bengali bn
- Bosniaque bs
- Bulgare bg
- Catalan ca
- Cebuano ceb (ISO-639-2)
- Chinois (simplifié) zh-CN (BCP-47)
- Chinois (traditionnel) zh-TW (BCP-47)
- Corse co
- Croate hr
- Tchèque cs
- Danish da

- Néerlandais nl
- Anglais en
- Esperanto eo
- Estonien et
- Finlandais fi
- Français fr
- Frisien fy
- Galicien gl
- Géorgien Ka
- Allemand de
- Grec el
- Gujarati gu
- Créole Haïtien ht
- Hausa ha
- Hawaïen haw (ISO-639-2)
- Hébreu iw
- Hindi hi
- Hmong hmh (ISO-639-2)
- Hongrois hu
- Islandais is
- Igbo ig
- Indonésien id
- Irlandais ga
- Italien it
- Japonais ja
- Javanais jw
- Kannada kn
- Kazakh kk
- Khmer km
- Coréen ko
- Kurde ku
- Kirghize ky
- Laotien lo
- Latin la
- Letton lv
- Lituanien lt
- Luxembourgeois lb
- Macédonien mk
- Malgache mg
- Malais ms
- Malayalam ml
- Maltais mt
- Maori mi
- Marathi mr
- Mongol mn
- Myanmar (Birmanie) my
- Nepali ne
- Norvégien no
- Nyanja (Chichewa) ny
- Pashtoun ps
- Persan fa
- Polonais pl
- Portugais (Portugal, Brésil) pt
- Punjabi pa

- Roumain ro
- Russe ru
- Samoan sm
- Écossais gaélique gd
- Serbe sr
- Sesotho st
- Shona sn
- Sindhi sd
- Sinhala (Sinhala) si
- Slovaque sk
- Slovène sl
- Somalien so
- Espagnol es
- Soudanais su
- Swahili sw
- Suédois sv
- Tagalog (Philippin) tl
- Tajik tg
- Tamil ta
- Telugu te
- Thai th
- Turc tr
- Ukrainien uk
- Ourdou ur
- Uzbek uz
- Vietnamien vi
- Gallois cy
- Xhosa xh
- Yiddish yi
- Yoruba yo
- Zulu zu

1.2 Quick Start for auto-translation with potranslator

This section describes how to translate documents generated by [Sphinx](#) with the *potranslator* command.

1. Create your document(s) by using Sphinx :

```
$ sphinx-build -b html /path/to/docs path/to/docs/_build
```

2. Optionally add the settings to your *conf.py* if you have one :

```
locale_dirs = ['locale/'] #path is an example but this is the_  
↪recommended path.  
gettext_compact = False #optional.
```

locale_dirs est requis et 'gettext_compact' est optionnel.

3. Extract the document's translatable messages into pot files (make sure you are in the folder containing *make.bat* and *Makefile* if you are on windows) :

```
$ make gettext
```

4. Translate/Update your documents in German and Japanese :

```
$ potranslator update -p _build/gettext -l de -l ja
```

Terminé. Vous avez ces répertoires qui contiennent des fichiers po avec des entrées traduites automatiquement :

```
./locale/de/LC_MESSAGES/  
./locale/ja/LC_MESSAGES/
```

5. Translate/Update your documents in Japanese, build the compiled mo files and generate the translated html documents :

Command line (for Unix systems) :

```
$ potranslator build  
$ make -e SPHINXOPTS="-D language='ja'" html
```

Command line (for Windows cmd.exe) :

```
> set SPHINXOPTS=-D language=de  
> potranslator build  
> .\make.bat html
```

Command line (for PowerShell) :

```
> Set-Item env:SPHINXOPTS "-D language=de"  
> potranslator build  
> .\make.bat html
```

C'est tout !

1.3 Fonctionnalités de base

- Translate from pot files or update existing po files with auto-generated translation.
- Build mo files from translated po or pot files.

1.3.1 Exigences pour les fonctionnalités de base

- Python 3.6, 3.5, 3.4, 2.7, pypy.
- external libraries : [setuptools](#), [six](#), [babel](#), [click](#), [googletrans](#), [polib](#)

1.4 Fonctionnalités optionnelles

Ces fonctionnalités nécessitent la bibliothèque [transifex-client](#).

- create a `.transifexrc` file from an environment variable, without interactive input.
- create a `.tx/config` file without interactive input.
- update a `.tx/config` file from locale/pot files automatically.
- Compiler des fichiers mo à partir de fichiers po dans le répertoire locale.

You need to use the `tx` command to use the following features :

- `tx push -s` : push pot (catalogues de traduction) vers transifex.
- `tx pull -l ja` : tire po (catalogues traduits) de transifex.

1.4.1 Exigences pour les fonctionnalités optionnelles

- Votre compte [transifex](#) si vous voulez télécharger des fichiers po vers transifex.
- librairie externe : `transifex-client`

1.5 Installation

It is strongly recommended to use virtualenv for this procedure :

```
$ pip install potranslator
```

If you want to use the *Optional Features*, you need install this additional library :

```
$ pip install potranslator[transifex]
```

1.6 Commandes, options, variables d'environnement

1.6.1 Commandes

Tapez `potranslator` sans arguments pour afficher les instructions d'aide.

1.6.2 Configurer les variables d'environnement

All command-line options can be set with environment variables using the format `POTRANSLATOR_<UPPER_LONG_NAME>`.

Dashes (-) have to be replaced with underscores (_).

For example, to set the target languages :

```
$ export POTRANSLATOR_LANGUAGE=de,ja
```

On the Windows command line :

```
> set POTRANSLATOR_LANGUAGE=de,ja
```

This is the same as passing the option to `potranslator` directly :

```
$ potranslator <command> --language=de --language=ja
```

1.6.3 Configurer sphinx conf.py

Add the following settings to your sphinx document's `conf.py` if it exists :

```
locale_dirs = ['locale/'] #for example
gettext_compact = False #optional
```

1.6.4 Configurer Makefile / make.bat

make gettext will generate pot files into the *_build/gettext* directory, however pot files can be generated in the *locale/pot* directory if convenient.

You can do that by replacing *_build/gettext* with *locale/pot* in your *Makefile* and/or *make.bat* that was generated by sphinx-quickstart.

1.7 Licence

Licencié sous la licence BSD. Voir le fichier LICENCE pour des termes spécifiques.

1.8 Original

The Command Line Interface and the *transifex* integration of *potranslator* are adapted from *sphinx-intl*.

— <https://pypi.org/project/sphinx-intl>

1.9 CHANGEMENTS

See : <https://github.com/SekouD/potranslator/blob/master/HISTORY.rst>

2.1 Version stable

Pour installer potranslator, exécutez cette commande dans votre terminal :

```
$ pip install potranslator
```

C'est la méthode préférée pour installer potranslator, car elle installera toujours la dernière version stable.

If you want to use the [Optional Features](#), you need to install this additional library [transifex-client](#) :

```
$ pip install potranslator[transifex]
```

Si vous n'avez pas installé [pip](#), ce ' [Guide d'installation Python](#) ' peut vous guider tout au long du processus.

2.2 Depuis le code source

Le code source de Potranslator peut être téléchargé à partir du [Github repo](#).

Vous pouvez soit cloner le référentiel public :

```
$ git clone git://github.com/SekouD/potranslator
```

Ou téléchargez l' [archive](#) :

```
$ curl -OL https://github.com/SekouD/potranslator/tarball/master
```

Une fois que vous avez une copie du code source, vous pouvez l'installer avec :

```
$ python setup.py install
```


3.1 Depuis un programme Python

To use potranslator in a python project :

```
from potranslator import PoTranslator

languages = ('fr', 'es', 'it')
translator = PoTranslator(pot_dir='path/to/pot_dir', locale_dir='path/to/locale_dir')

results = translator.translate_all_pot(src_lang='en', target_langs=languages, auto_
→save=False)
```

3.2 Commandes, options, et variables d'environnement

3.2.1 Commandes

Tapez *potranslator* sans arguments pour afficher les instructions d'aide.

3.2.2 Basic Usage

This section describes how to translate documents generated by *Sphinx* with the *potranslator* command.

1. Create your document(s) by using Sphinx :

```
$ sphinx-build -b html /path/to/docs path/to/docs/_build
```

2. Optionally add the settings to your *conf.py* if you have one :

```
locale_dirs = ['locale/'] #path is an example but this is the
↳recommended path.
gettext_compact = False #optional.
```

`locale_dirs` is required and `gettext_compact` is optional.

3. Extract the document's translatable messages into pot files (make sure you are in the folder containing `make.bat` and `Makefile` if you are on windows) :

```
$ make gettext
```

4. Translate/Update your documents in German and Japanese :

```
$ potranslator update -p _build/gettext -l de -l ja
```

Done. You got these directories that contain po files with auto-translated entries :

```
./locale/de/LC_MESSAGES/
./locale/ja/LC_MESSAGES/
```

5. Translate/Update your documents in Japanese, build the compiled mo files and generate the translated html documents :

Command line (for Unix systems) :

```
$ potranslator build
$ make -e SPHINXOPTS="-D language='ja'" html
```

Command line (for Windows cmd.exe) :

```
> set SPHINXOPTS=-D language=de
> potranslator build
> .\make.bat html
```

Command line (for PowerShell) :

```
> Set-Item env:SPHINXOPTS "-D language=de"
> potranslator build
> .\make.bat html
```

That's all !

3.2.3 Configurer les variables d'environnement

All command-line options can be set with environment variables using the format `POTRANSLATOR_<UPPER_LONG_NAME>` .

Dashes (-) have to be replaced with underscores (_).

For example, to set the languages :

```
$ export POTRANSLATOR_LANGUAGE=de,ja
```

On the Windows command line :

```
> set POTRANSLATOR_LANGUAGE=de,ja
```

This is the same as passing the option to `potranslator` directly :


```
$ potranslator <command> --language=de --language=ja
```

3.2.4 Configurer sphinx conf.py

Add the following settings to your sphinx document's conf.py if it exists :

```
locale_dirs = ['locale/'] #for example
gettext_compact = False #optional
```

3.2.5 Configurer Makefile / make.bat

make gettext will generate pot files into *_build/gettext* directory, however pot files can be generated in the *locale/pot* if convenient.

You can do that by replacing *_build/gettext* with *locale/pot* in your *Makefile* and/or *make.bat* that was generated by sphinx-quickstart.

4.1 Référence de l'API pour les classes dans potranslator.potranslator.py

Module principal

class potranslator.potranslator.**PoTranslator** (*pot_dir=None, locale_dir=None*)

C'est la classe principale de cette librairie. Cette classe gère toutes les tâches de traduction.

Paramètres

- **pot_dir** – string. Chemin vers le répertoire contenant les fichiers *.pot.
- **locale_dir** – string. Chemin d'accès au répertoire locale.

translate (*file_name, target_lang='auto', src_lang='auto', encoding='utf-8', auto_save=False, compiled=False*)

Traduit le fichier po donné dans la langue cible spécifiée.

Paramètres

- **file_name** – string. Chemin vers le nom de fichier du fichier à traduire.
- **target_lang** – string. Langue cible pour la traduction.
- **src_lang** – string. Langue source pour la traduction.
- **encoding** – string. Encodage pour enregistrer les fichiers po.
- **auto_save** – bool. Active la fonction de sauvegarde automatique.
- **compiled** – bool. Active la compilation en fichiers mo.

Renvoie tuple. Un tuple contenant la version traduite du catalogue original et l'état du POFile.

translate_all_locale (*src_lang='auto', encoding='utf-8', auto_save=False, compiled=False*)

Traduit tous les fichiers po dans les langues trouvées dans le dossier locale.

Paramètres

- **src_lang** – string. Langue source pour la traduction.
- **encoding** – string. Encodage pour enregistrer les fichiers po.
- **auto_save** – bool. Active la fonction de sauvegarde automatique.
- **compiled** – bool. Active la compilation en fichiers mo.

Renvoie Dictionnaire. Un dictionnaire de fichiers po.

translate_from_pot (*filename, status, target_langs, src_lang='auto', encoding='utf-8', auto_save=False, compiled=False*)

Traduit le fichier de pot donné dans les langues cibles spécifiées.

Paramètres

- **filename** – string. Chemin vers le nom de fichier du fichier à traduire.
- **target_langs** – séquence de strings. Langues cible pour la traduction.
- **src_lang** – string. Langue source pour la traduction.
- **encoding** – string. Encodage pour enregistrer les fichiers po.
- **auto_save** – bool. Active la fonction de sauvegarde automatique.
- **compiled** – bool. Active la compilation en fichiers mo.

Renvoie Dictionnaire. Un dictionnaire de fichiers po.

translate_all_pot (*target_langs, src_lang='auto', encoding='utf-8', auto_save=False, compiled=False*)

Traduit tous les fichiers *.pot dans le dossier dans les langues cibles spécifiées.

Paramètres

- **target_langs** – séquence de strings. Langues cible pour la traduction.
- **src_lang** – string. Langue source pour la traduction.
- **encoding** – string. Encodage pour enregistrer les fichiers po.
- **auto_save** – bool. Active la fonction de sauvegarde automatique.
- **compiled** – bool. Active la compilation en fichiers mo.

Renvoie Dictionnaire. Un dictionnaire de fichiers po.

Les contributions sont les bienvenues, et elles sont grandement appréciées ! Chaque petit coup de pouce aide, et le crédit sera toujours donné.

Vous pouvez contribuer de plusieurs façons :

5.1 Types de contributions

5.1.1 Signaler les bugs

Rapportez les bugs sur <https://github.com/SekouD/potranslator/issues>.

Si vous signalez un bug, veuillez inclure :

- Le nom et la version de votre système d'exploitation.
- Tous les détails concernant votre configuration locale qui pourraient être utiles pour le dépannage.
- Étapes détaillées pour reproduire le bug.

5.1.2 Corriger les bugs

Regardez sur le site GitHub du projet pour les bugs. Tout ce qui est marqué « bug » et « help wanted » est ouvert à quiconque veut l'implémenter.

5.1.3 Implémenter de nouvelles fonctionnalités

Regardez sur la page GitHub du projet pour les fonctionnalités. Tout ce qui est marqué « amélioration » et « help wanted » est ouvert à quiconque veut l'implémenter.

5.1.4 Écrire de la documentation

potranslator pourrait toujours utiliser plus de documentation, que ce soit dans le cadre de la doc officielle de potranslator, dans les docstrings, ou même sur le web dans des articles de blog, et autres.

5.1.5 Soumettre des commentaires

La meilleure façon d'envoyer vos commentaires est de déposer un problème sur <https://github.com/SekouD/potranslator/issues>.

Si vous proposez une fonctionnalité :

- Expliquez en détail comment cela fonctionnerait.
- Gardez la portée aussi étroite que possible, pour rendre la fonctionnalité plus facile à mettre en œuvre.
- Rappelez-vous que c'est un projet dirigé par des bénévoles, et que les contributions sont les bienvenues :)

5.2 Commencez !

Prêt à contribuer ? Voici comment configurer *potranslator* pour le développement local.

1. Forkez le repo *potranslator* sur GitHub.
2. Cloner votre fork localement

```
$ git clone git@github.com:your_name_here/potranslator.git
```

3. Installez votre copie locale dans un virtualenv. En supposant que virtualenvwrapper soit installé, voici comment configurer votre fork pour le développement local :

```
$ mkvirtualenv potranslator
$ cd potranslator/
$ python setup.py develop
```

4. Créer une branche pour le développement local

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Vous pouvez maintenant effectuer vos modifications localement.

5. Lorsque vous avez fini de faire des changements, vérifiez que vos changements passent flake8 et les tests, y compris le test d'autres versions de Python avec tox

```
$ flake8 potranslator tests
$ python setup.py test or py.test
$ tox
```

Pour obtenir flake8 et tox, il suffit de les installer dans votre virtualenv.

6. Validez vos changements et faites un push de votre branche vers GitHub

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Soumettez une pull request sur le site Web de GitHub.

5.3 Directives des Pull Request

Avant de soumettre une pull request, vérifiez qu'elle respecte les consignes suivantes :

1. La pull request doit inclure des tests.
2. Si la pull request ajoute une fonctionnalité, la documentation doit être mise à jour. Mettez votre nouvelle fonctionnalité dans une fonction avec une docstring et ajoutez la fonctionnalité à la liste dans README.rst.
3. La requête pull devrait fonctionner pour Python 2.7, 3.4, 3.5 et 3.6, et pour PyPy. Vérifiez https://travis-ci.org/SekouD/potranslator/pull_requests et assurez-vous que les tests réussissent pour toutes les versions Python supportées.

5.4 Conseils

Pour exécuter un sous-ensemble des tests

```
$ py.test tests.test_potranslator
```

5.5 Déployer

Un rappel pour les mainteneurs sur la façon de déployer. Assurez-vous que toutes vos modifications sont validées (y compris une entrée dans HISTORY.rst). Puis exécutez

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis déploiera ensuite le package sur PyPI si les tests réussissent.

6.1 Responsable du développement

— SekouD <sekoud.python@gmail.com> GPG key ID : B51D1046EF63C50B

6.2 Contributeurs

Aucun pour l'instant. Pourquoi ne pas être le premier ?

7.1 1.1.0 (2018-07-08)

- Now uses `importlib_ressources` for faster startup from CLI.
- Updated the command line usability.
- Added Type Annotation compliant with PEP 561.
- Updated Documentation.

7.2 1.0.5 (2018-07-06)

- Updated Documentation.
- Translated the documentation in French, Spanish, Italian, German, Italian, Japanese and Chinese.
- More detailed updates to the po files meta-data.

7.3 1.0.0 (2018-07-05)

- Première release candidate.
- Ajout de l'interface de ligne de commande.

7.4 0.1.0 (2018-06-27)

- Première version sur PyPI.

CHAPITRE 8

Indices et tableaux

- : ref : *genindex*
- : ref : *modindex*
- : ref : *search*

p

`potranslator.potranslator`, 15

P

PoTranslator (classe dans potranslator.potranslator), 15
potranslator.potranslator (module), 15

T

translate()	(méthode tor.potranslator.PoTranslator), 15	potransla-
translate_all_locale()	(méthode tor.potranslator.PoTranslator), 15	potransla-
translate_all_pot()	(méthode tor.potranslator.PoTranslator), 16	potransla-
translate_from_pot()	(méthode tor.potranslator.PoTranslator), 15	potransla-