
potranslator Documentation

Release 1.1.5

SekouD

28.07.2018

1	Potranslator	3
1.1	Unterstützte Sprachen	3
1.2	Quick Start for auto-translation with potranslator	6
1.3	Grundfunktionen	7
1.4	Optionale Funktionen	7
1.5	Installation	8
1.6	Befehle, Optionen, Umgebungsvariablen	8
1.7	Lizenz	9
1.8	Original	9
1.9	ÄNDERUNGEN	9
2	Installation	11
2.1	Stabile Version	11
2.2	Aus Quellen	11
3	Verwendung	13
3.1	Aus einem Python-Programm	13
3.2	Befehle, Optionen, Umgebungsvariablen	13
4	Paket Api Dokumentation für Potranslator	17
4.1	API-Referenz für die Klassen in potranslator.potranslator.py	17
5	Beitragend	19
5.1	Arten von Beiträgen	19
5.2	Loslegen!	20
5.3	Pull Request Richtlinien	21
5.4	Tipps	21
5.5	Bereitstellen	21
6	Kredite	23
6.1	Entwicklungsleiter	23
6.2	Mitwirkende	23
7	Geschichte	25
7.1	1.1.0 (2018-07-08)	25
7.2	1.0.5 (2018-07-06)	25
7.3	1.0.0 (2018-07-05)	25

7.4	0.1.0 (2018-06-27)	25
8	Indizes und Tabellen	27
	Python-Modulindex	29

Inhalt:

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](#).

Optional: unterstützt den Transifex-Collaboration-Service zum Hochladen der automatisch generierten Übersetzungen in [transifex](#) für kollaborative Korrekturen der Übersetzungen.

1.1 Unterstützte Sprachen

- Afrikaans af
- Albanisch sq
- Amharisch bin
- Arabisch ar
- Armenisch hy
- Aserbaidsschanische az
- Baskische eu
- Belarussisch sein
- Bengalisch
- Bosnisch bs
- Bulgarisch
- Katalanisch ca
- Cebuano ceb (ISO-639-2)

- Chinesisch (vereinfacht) zh-CN (BCP-47)
- Chinesisch (traditionell) zh-TW (BCP-47)
- Korsische co
- Kroatische hr
- Tschechische cs
- Dänisch da
- Holländische nl
- Englisch en
- Esperanto eo
- Estnisch et
- Finnische fi
- Französisch fr
- Friesisch fy
- Galizisch gl
- Georgischer Ka
- Deutsch de
- Griechischer el
- Gujarati gu
- Haitianische kreolische ht
- Hausa ha
- Hawaiianer (ISO-639-2)
- Hebräisch
- Hindi Hallo
- Hmong hmn (ISO-639-2)
- Ungarischer hu
- Isländisch ist
- Igbo ig
- Indonesische ID
- Irisches ga
- Italienisch es
- Japanisch ja
- Javanisch jw
- Kannada kn
- Kasachisch kk
- Khmer km
- Koreanisch ko

- Kurdisch ku
- Kyrgyz ky
- Lao lo
- Lateinisch la
- Lettisch lv
- Litauisch lt
- Luxemburgisch lb
- Mazedonische mk
- Madagassische mg
- Malaiische Frau
- Malayalam ml
- Maltesisch mt
- Maori mi
- Marathi Herr
- Mongolische mn
- Myanmar (Birmanisch) mein
- Nepali ne
- Norwegische Nr
- Nyanja (Chichewa) NY
- Paschtu ps
- Persische Fa
- Polnisch pl
- Portugiesisch (Portugal, Brasilien) pt
- Punjabi pa
- Rumänisch
- Russisch ru
- Samoanische Kl
- Schottisches Gälisch gd
- Serbisch sr
- Sesotho st
- Shona sn
- Sindhi sd
- Singhalesisch (Singhalesisch) si
- Slowakisch
- Slowenisch
- Somali so

- Spanisch es
- Sundanesische su
- Swahili sw
- Schwedisch sv
- Tagalog (Filipino) tl
- Tadschikisch tg
- Tamilisch ta
- Telugu te
- Thailändischer Th
- Türkische tr
- Ukrainisch uk
- Urdu ur
- Usbekische Uz
- Vietnamesisch vi
- Welsh cy
- Xhosa xh
- Jiddisch 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 ist erforderlich und *gettext_compact* ist 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
```

Erledigt. Sie haben diese Verzeichnisse, die PO-Dateien mit automatisch übersetzten Einträgen enthalten:

```
./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
```

Das ist alles!

1.3 Grundfunktionen

- 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 Anforderungen für die grundlegenden Funktionen

- Python 3.6, 3.5, 3.4, 2.7, Pypy.
- external libraries: `setuptools`, `six`, `babel`, `click`, `googletrans`, `polib`

1.4 Optionale Funktionen

Diese Funktionen benötigen die Bibliothek `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.
- Erstellen Sie MO-Dateien aus Po-Dateien im Gebietschema-Verzeichnis.

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

- `tx push -s`: Push-Pot (Übersetzungskataloge) nach transifex.
- `tx pull -l ja`: ziehe po (übersetzte Kataloge) von transifex.

1.4.1 Anforderungen für die optionalen Funktionen

- Ihr *transifex* Konto, wenn Sie po-Dateien von *transifex* hochladen / herunterladen möchten.
- externe Bibliothek: *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 Befehle, Optionen, Umgebungsvariablen

1.6.1 Befehle

Geben Sie *potranslator* ohne Argumente ein, um die Hilfeanweisungen anzuzeigen.

1.6.2 Setup-Umgebungsvariablen

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 Richten Sie *sphinx conf.py* ein

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 Setup 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 Lizenz

Lizenziert unter der BSD-Lizenz. Spezifische Bedingungen finden Sie in der LIZENZDATEI.

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 ÄNDERUNGEN

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

2.1 Stabile Version

Um den Potranslator zu installieren, führen Sie diesen Befehl in Ihrem Terminal aus:

```
$ pip install potranslator
```

Dies ist die bevorzugte Methode zur Installation von Potranslator, da immer die neueste stabile Version installiert wird.

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

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

Wenn Sie `pip` nicht installiert haben, kann dieses ‘Python Installationshandbuch’_ Sie durch den Prozess führen.

2.2 Aus Quellen

Die Quellen für den Potranslator können vom [Github Repo](#) heruntergeladen werden.

Sie können das öffentliche Repository entweder klonen:

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

Oder lade den [Tarball](#) herunter:

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

Sobald Sie eine Kopie der Quelle haben, können Sie sie installieren mit:

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


3.1 Aus einem Python-Programm

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 Befehle, Optionen, Umgebungsvariablen

3.2.1 Befehle

Geben Sie *potranslator* ohne Argumente ein, um die Hilfeanweisungen anzuzeigen.

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 Setup-Umgebungsvariablen

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 Richten Sie sphinx conf.py ein

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 Setup 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.

Paket Api Dokumentation für Potranslator

4.1 API-Referenz für die Klassen in potranslator.potranslator.py

Hauptmodul.

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

Dies ist die Hauptklasse dieser Bibliothek. Diese Klasse verwaltet alle Übersetzungsaufgaben.

Parameter

- **pot_dir** – Zeichenfolge. Pfad zum Pot-Verzeichnis.
- **locale_dir** – Zeichenfolge. Pfad zum Gebietsschema-Verzeichnis

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

Übersetzt die angegebene po-Datei in der angegebenen Zielsprache.

Parameter

- **file_name** – Zeichenfolge. Pfad zum Dateinamen der zu übersetzenden Datei.
- **target_lang** – Zeichenfolge. Zielsprache für die Übersetzung.
- **src_lang** – Zeichenfolge. Ausgangssprache für die Übersetzung.
- **encoding** – Zeichenfolge. Kodierung zum Speichern der PO-Dateien.
- **auto_save** – Bool. Schaltet die automatische Speicherfunktion um.
- **compiled** – Bool. Schaltet die Kompilierung auf MO-Dateien um.

Rückgabe Tupel. Ein Tupel, das die übersetzte Version des ursprünglichen Katalogs und den Status der POFile enthält.

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

Übersetzt alle PO-Dateien in den gefundenen Sprachen im Gebietsschema-Ordner.

Parameter

- **src_lang** – Zeichenfolge. Ausgangssprache für die Übersetzung.
- **encoding** – Zeichenfolge. Kodierung zum Speichern der PO-Dateien.
- **auto_save** – Bool. Schaltet die automatische Speicherfunktion um.
- **compiled** – Bool. Schaltet die Kompilierung auf MO-Dateien um.

Rückgabe Wörterbuch. Ein Wörterbuch von Po-Dateien.

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

Übersetzt die angegebene Topfdatei in die angegebenen Zielsprachen.

Parameter

- **filename** – Zeichenfolge. Pfad zum Dateinamen der zu übersetzenden Datei.
- **target_langs** – Sequenz von Strings. Zielsprache für die Übersetzung.
- **src_lang** – Zeichenfolge. Ausgangssprache für die Übersetzung.
- **encoding** – Zeichenfolge. Kodierung zum Speichern der PO-Dateien.
- **auto_save** – Bool. Schaltet die automatische Speicherfunktion um.
- **compiled** – Bool. Schaltet die Kompilierung auf MO-Dateien um.

Rückgabe Wörterbuch. Ein Wörterbuch von Po-Dateien.

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

Übersetzt alle Pot-Dateien im Pot-Ordner in den angegebenen Zielsprachen.

Parameter

- **target_langs** – Sequenz von Strings. Zielsprache für die Übersetzung.
- **src_lang** – Zeichenfolge. Ausgangssprache für die Übersetzung.
- **encoding** – Zeichenfolge. Kodierung zum Speichern der PO-Dateien.
- **auto_save** – Bool. Schaltet die automatische Speicherfunktion um.
- **compiled** – Bool. Schaltet die Kompilierung auf MO-Dateien um.

Rückgabe Wörterbuch. Ein Wörterbuch von Po-Dateien.

Beiträge sind willkommen, und sie werden sehr geschätzt! Jedes kleine bisschen hilft, und Kredit wird immer gegeben. Sie können auf viele Arten beitragen:

5.1 Arten von Beiträgen

5.1.1 Fehler melden

Melden Sie Fehler unter <https://github.com/SekouD/potranslator/issues>.

Wenn Sie einen Fehler melden, geben Sie Folgendes an:

- Ihr Betriebssystem Name und Version.
- Alle Details zu Ihrer lokalen Konfiguration, die bei der Fehlerbehebung hilfreich sein können.
- Detaillierte Schritte zum Reproduzieren des Fehlers.

5.1.2 Fehler beheben

Sehen Sie sich die GitHub-Probleme an. Alles, was mit „bug“ und „help wanted“ getaggt ist, steht jedem offen, der es implementieren will.

5.1.3 Funktionen implementieren

Sehen Sie sich die GitHub-Ausgaben für Funktionen an. Alles, was mit „Enhancement“ und „help wanted“ gekennzeichnet ist, steht jedem offen, der es implementieren möchte.

5.1.4 Dokumentation schreiben

Potranslator könnte immer mehr Dokumentation verwenden, sei es als Teil der offiziellen Potranslator-Dokumentation, in Docstrings oder sogar im Web in Blogposts, Artikeln und so weiter.

5.1.5 Feedback senden

Der beste Weg, Feedback zu senden, besteht darin, ein Problem unter <https://github.com/SekouD/potranslator/issues> zu melden.

Wenn Sie ein Feature vorschlagen:

- Erklären Sie im Detail, wie es funktionieren würde.
- Halten Sie den Umfang so eng wie möglich, um die Implementierung zu erleichtern.
- Denken Sie daran, dass dies ein freiwilliges Projekt ist und dass Beiträge willkommen sind :)

5.2 Loslegen!

Bereit zum Beitrag? Hier erfahren Sie, wie Sie den „Potranslator“ für die lokale Entwicklung einrichten.

1. Fork den *Potranslator* Repo auf GitHub.
2. Klonen Sie Ihre Gabel lokal:

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

3. Installiere deine lokale Kopie in ein virtualenv. Angenommen, Sie haben virtualenvwrapper installiert, so richten Sie Ihren fork für lokale Entwicklung ein:

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

4. Erstellen Sie eine Niederlassung für lokale Entwicklung

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

Jetzt können Sie Ihre Änderungen lokal vornehmen.

5. Wenn Sie mit den Änderungen fertig sind, überprüfen Sie, ob Ihre Änderungen flake8 und die Tests bestehen, einschließlich Tests anderer Python-Versionen mit tox

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

Um Flake8 und Tox zu erhalten, installiere sie einfach in dein virtualenv.

6. Übernehmen Sie Ihre Änderungen und schieben Sie Ihren Zweig zu GitHub

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

7. Senden Sie eine Pull-Anfrage über die GitHub-Website.

5.3 Pull Request Richtlinien

Bevor Sie eine Pull-Anforderung senden, überprüfen Sie, ob diese Richtlinien erfüllt sind:

1. Die Pull-Anforderung sollte Tests enthalten.
2. Wenn die Pull-Anforderung Funktionalität hinzufügt, sollten die Dokumente aktualisiert werden. Fügen Sie Ihre neue Funktionalität in eine Funktion mit einem Docstring ein und fügen Sie sie der Liste in README.rst hinzu.
3. Die Pull-Anfrage sollte für Python 2.7, 3.4, 3.5 und 3.6 und für PyPy funktionieren. Überprüfen Sie https://travis-ci.org/SekouD/potranslator/pull_requests und stellen Sie sicher, dass die Tests für alle unterstützten Python-Versionen gelten.

5.4 Tipps

So führen Sie eine Teilmenge von Tests aus

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

5.5 Bereitstellen

Eine Erinnerung für die Betreuer zur Bereitstellung. Stellen Sie sicher, dass alle Ihre Änderungen festgeschrieben sind (einschließlich eines Eintrags in HISTORY.rst). Dann renne:

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

Travis wird dann in PyPI bereitstellen, wenn Tests bestanden werden.

6.1 Entwicklungsleiter

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

6.2 Mitwirkende

Noch keine. Warum nicht der Erste sein?

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)

- Erster Release-Kandidat.
- Befehlszeilenschnittstelle hinzugefügt.

7.4 0.1.0 (2018-06-27)

- Erste Veröffentlichung auf PyPI.

Indizes und Tabellen

- : ref: *genindex*
- : ref: *modindex*
- : ref: *suchen*

p

`potranslator.potranslator`, 17

P

PoTranslator (Klasse in potranslator.potranslator), [17](#)
potranslator.potranslator (Modul), [17](#)

T

translate() (Methode von potransla-
tor.potranslator.PoTranslator), [17](#)
translate_all_locale() (Methode von
tor.potranslator.PoTranslator), [17](#)
translate_all_pot() (Methode von
tor.potranslator.PoTranslator), [18](#)
translate_from_pot() (Methode von
tor.potranslator.PoTranslator), [18](#)