Googletrans Documentation

Release 2.4.1

SuHun Han (ssut)

Apr 20, 2020

Contents

Googletrans is a **free** and **unlimited** python library that implemented Google Translate API. This uses the Google Translate Ajax API to make calls to such methods as detect and translate.

CHAPTER 1

Features

- Fast and reliable - it uses the same servers that translate.google.com uses
- Auto language detection
- Bulk translations
- Customizable service URL
- Connection pooling (the advantage of using requests.Session)
- HTTP/2 support

1.1 Note on library usage

- The maximum character limit on a single text is 15k.
- Due to limitations of the web version of google translate, this API does not guarantee that the library would work properly at all times. (so please use this library if you don't care about stability.)
- If you want to use a stable API, I highly recommend you to use Google's official translate API.
- If you get HTTP 5xx error or errors like #6, it's probably because Google has banned your client IP address.

CHAPTER 2

Quickstart

You can install it from PyPI:

```
$ pip install ggtrans
```

2.1 HTTP/2 support

This is a great deal for everyone! (up to 2x times faster in my test) If you want to get googletrans faster you should install hyper package. Googletrans will automatically detect if hyper is installed and if so, it will be used for http networking.

2.2 Basic Usage

If source language is not given, google translate attempts to detect the source language.

```
>>> from googletrans import Translator
>>> translator = Translator()
>>> translator.translate('.')
# <Translated src=ko dest=en text=Good evening. pronunciation=Good evening.>

>>> translator.translate('.', dest='ja')
# <Translated src=ko dest=ja text= pronunciation=Kon'nichiwa.>

>>> translator.translate('veritas lux mea', src='la')
# <Translated src=la dest=en text=The truth is my light pronunciation=The truth is my␣
↪light>
```

2.3 Customize service URL

You can use another google translate domain for translation. If multiple URLs are provided it then randomly chooses a domain.

```
>>> from googletrans import Translator
>>> translator = Translator(service_urls=[
      'translate.google.com',
      'translate.google.co.kr',
    ])
```

2.4 Advanced Usage (Bulk)

Array can be used to translate a batch of strings in a single method call and a single HTTP session. The exact same method shown above work for arrays as well.

```
>>> translations = translator.translate(['The quick brown fox', 'jumps over', 'the␣
↪lazy dog'], dest='ko')
>>> for translation in translations:
...    print(translation.origin, ' -> ', translation.text)
# The quick brown fox  ->
# jumps over  ->
# the lazy dog  ->
```

2.5 Language detection

The detect method, as its name implies, identifies the language used in a given sentence.

```
>>> translator.detect('    .')
# <Detected lang=ko confidence=0.27041003>
>>> translator.detect('')
# <Detected lang=ja confidence=0.64889508>
>>> translator.detect('This sentence is written in English.')
# <Detected lang=en confidence=0.22348526>
>>> translator.detect('Tiu frazo estas skribita en Esperanto.')
# <Detected lang=eo confidence=0.10538048>
```

CHAPTER 3

API Guide

CHAPTER 4

googletrans.Translator

class googletrans.**Translator**(*service_urls=None*, *user_agent='Mozilla/5.0 (Windows NT 10.0; Win64; x64)'*, *proxies=None*, *timeout=None*)

Google Translate ajax API implementation class

You have to create an instance of Translator to use this API

Parameters

- **service_urls** (*a sequence of strings*) – google translate url list. URLs will be used randomly. For example `['translate.google.com', 'translate.google.co.kr']`
- **user_agent** (`str`) – the User-Agent header to send when making requests.
- **proxies** (*dictionary*) – proxies configuration. Dictionary mapping protocol or protocol and host to the URL of the proxy For example `{'http': 'foo.bar:3128', 'http://host.name': 'foo.bar:4012'}`
- **timeout** (*number or a double of numbers*) – Definition of timeout for Requests library. Will be used by every request.

translate(*text*, *dest='en'*, *src='auto'*)

Translate text from source language to destination language

Parameters

- **text** (UTF-8 `str`; `unicode`; string sequence (list, tuple, iterator, generator)) – The source text(s) to be translated. Batch translation is supported via sequence input.
- **dest** – The language to translate the source text into. The value should be one of the language codes listed in `googletrans.LANGUAGES` or one of the language names listed in `googletrans.LANGCODES`.
- **dest** – `str`; `unicode`
- **src** – The language of the source text. The value should be one of the language codes listed in `googletrans.LANGUAGES` or one of the language names listed in

googletrans.LANGCODES. If a language is not specified, the system will attempt to identify the source language automatically.

- **src** – str; unicode

Return type *Translated*

Return type list (when a list is passed)

Basic usage:

```
>>> from googletrans import Translator
>>> translator = Translator()
>>> translator.translate('.')
<Translated src=ko dest=en text=Good evening. pronunciation=Good evening.>
>>> translator.translate('.', dest='ja')
<Translated src=ko dest=ja text= pronunciation=Kon'nichiwa.>
>>> translator.translate('veritas lux mea', src='la')
<Translated src=la dest=en text=The truth is my light pronunciation=The␣
↪truth is my light>
```

Advanced usage:

```
>>> translations = translator.translate(['The quick brown fox', 'jumps␣
↪over', 'the lazy dog'], dest='ko')
>>> for translation in translations:
...     print(translation.origin, ' -> ', translation.text)
The quick brown fox  ->
jumps over  ->
the lazy dog  ->
```

detect(*text*)

Detect language of the input text

Parameters **text** (UTF-8 str; unicode; string sequence (list, tuple, iterator, generator)) – The source text(s) whose language you want to identify. Batch detection is supported via sequence input.

Return type *Detected*

Return type list (when a list is passed)

Basic usage:

```
>>> from googletrans import Translator
>>> translator = Translator()
>>> translator.detect('    .')
<Detected lang=ko confidence=0.27041003>
>>> translator.detect('')
<Detected lang=ja confidence=0.64889508>
>>> translator.detect('This sentence is written in English.')
<Detected lang=en confidence=0.22348526>
>>> translator.detect('Tiu frazo estas skribita en Esperanto.')
<Detected lang=eo confidence=0.10538048>
```

Advanced usage:

```
>>> langs = translator.detect(['', '', 'English', 'le français'])
>>> for lang in langs:
...     print(lang.lang, lang.confidence)
ko 1
ja 0.92929292
en 0.96954316
fr 0.043500196
```

CHAPTER 5

googletrans.models

class googletrans.models.**Translated**(*src*, *dest*, *origin*, *text*, *pronunciation*, *extra_data=None*)

Translate result object

Parameters

- **src** – source langauge (default: auto)
- **dest** – destination language (default: en)
- **origin** – original text
- **text** – translated text
- **pronunciation** – pronunciation

class googletrans.models.**Detected**(*lang*, *confidence*)

Language detection result object

Parameters

- **lang** – detected language
- **confidence** – the confidence of detection result (0.00 to 1.00)

CHAPTER 6

googletrans.gtoken

Hint: This is for internal use only to generate a valid token to access translate.google.com ajax API.

class googletrans.gtoken.**TokenAcquirer**(*tkk='0'*, *session=None*, *host='translate.google.com'*)

Google Translate API token generator

translate.google.com uses a token to authorize the requests. If you are not Google, you do have this token and will have to pay for use. This class is the result of reverse engineering on the obfuscated and minified code used by Google to generate such token.

The token is based on a seed which is updated once per hour and on the text that will be translated. Both are combined - by some strange math - in order to generate a final token (e.g. 744915.856682) which is used by the API to validate the request.

This operation will cause an additional request to get an initial token from translate.google.com.

Example usage:

```
>>> from googletrans.gtoken import TokenAcquirer
>>> acquirer = TokenAcquirer()
>>> text = 'test'
>>> tk = acquirer.do(text)
>>> tk
950629.577246
```

CHAPTER 7

googletrans.LANGUAGES

Hint: iso639-1 language codes for supported languages for translation. Some language codes also include a country code, like zh-CN or zh-TW.

```
LANGUAGES = {
    'af': 'afrikaans',
    'sq': 'albanian',
    'am': 'amharic',
    'ar': 'arabic',
    'hy': 'armenian',
    'az': 'azerbaijani',
    'eu': 'basque',
    'be': 'belarusian',
    'bn': 'bengali',
    'bs': 'bosnian',
    'bg': 'bulgarian',
    'ca': 'catalan',
    'ceb': 'cebuano',
    'ny': 'chichewa',
    'zh-cn': 'chinese (simplified)',
    'zh-tw': 'chinese (traditional)',
    'co': 'corsican',
    'hr': 'croatian',
    'cs': 'czech',
    'da': 'danish',
    'nl': 'dutch',
    'en': 'english',
    'eo': 'esperanto',
    'et': 'estonian',
    'tl': 'filipino',
    'fi': 'finnish',
    'fr': 'french',
    'fy': 'frisian',
```

(continues on next page)

(continued from previous page)

```
    'gl': 'galician',
    'ka': 'georgian',
    'de': 'german',
    'el': 'greek',
    'gu': 'gujarati',
    'ht': 'haitian creole',
    'ha': 'hausa',
    'haw': 'hawaiian',
    'iw': 'hebrew',
    'hi': 'hindi',
    'hmn': 'hmong',
    'hu': 'hungarian',
    'is': 'icelandic',
    'ig': 'igbo',
    'id': 'indonesian',
    'ga': 'irish',
    'it': 'italian',
    'ja': 'japanese',
    'jw': 'javanese',
    'kn': 'kannada',
    'kk': 'kazakh',
    'km': 'khmer',
    'ko': 'korean',
    'ku': 'kurdish (kurmanji)',
    'ky': 'kyrgyz',
    'lo': 'lao',
    'la': 'latin',
    'lv': 'latvian',
    'lt': 'lithuanian',
    'lb': 'luxembourgish',
    'mk': 'macedonian',
    'mg': 'malagasy',
    'ms': 'malay',
    'ml': 'malayalam',
    'mt': 'maltese',
    'mi': 'maori',
    'mr': 'marathi',
    'mn': 'mongolian',
    'my': 'myanmar (burmese)',
    'ne': 'nepali',
    'no': 'norwegian',
    'ps': 'pashto',
    'fa': 'persian',
    'pl': 'polish',
    'pt': 'portuguese',
    'pa': 'punjabi',
    'ro': 'romanian',
    'ru': 'russian',
    'sm': 'samoan',
    'gd': 'scots gaelic',
    'sr': 'serbian',
    'st': 'sesotho',
    'sn': 'shona',
    'sd': 'sindhi',
    'si': 'sinhala',
    'sk': 'slovak',
    'sl': 'slovenian',
```

(continues on next page)

(continued from previous page)

```
    'so': 'somali',
    'es': 'spanish',
    'su': 'sundanese',
    'sw': 'swahili',
    'sv': 'swedish',
    'tg': 'tajik',
    'ta': 'tamil',
    'te': 'telugu',
    'th': 'thai',
    'tr': 'turkish',
    'uk': 'ukrainian',
    'ur': 'urdu',
    'uz': 'uzbek',
    'vi': 'vietnamese',
    'cy': 'welsh',
    'xh': 'xhosa',
    'yi': 'yiddish',
    'yo': 'yoruba',
    'zu': 'zulu',
    'fil': 'Filipino',
    'he': 'Hebrew'
}

LANGCODES = dict(map(reversed, LANGUAGES.items()))
```

Python Module Index

g

Index