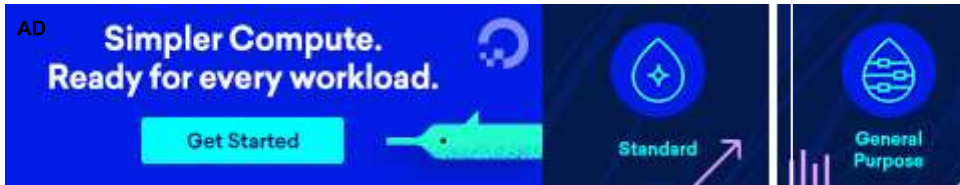


Text Translation with Google Translate API in Python

By  Guest Contributor (<https://twitter.com/stackabuse>) •

5 Comments (/text-translation-with-google-translate-api-in-python/#disqus_thread)



Unless you have been hiding under a rock, you have probably used Google Translate (<https://translate.google.com/>) on many occasions in your life. Whenever you try to translate a word or a sentence from a certain language to another, it is the Google Translate API which brings you the desired results in the background. Though you can translate anything by simply going to the Google Translate web page, you can also integrate Google Translate API (<https://pypi.org/project/googletrans/>) into your web applications or desktop programs. The best thing about the API is that it is extremely easy to set up and use.

You can actually do a lot of things with the help of the Google Translate API ranging from detecting languages to simple text translation, setting source and destination languages, and translating entire lists of text phrases. In this article, you will see how to work with the Google Translate API in the Python programming language.

Google Translate API Installation

Before you can work with the Google Translate API in Python, you will have to install it. There are two different methods of installing the API. The first method is straight forward. Simply go to terminal and use the `pip` installer to install the API, as you would for any other Python library. To do this, type the following command in your terminal:

```
$ pip install googletrans
```

Press `Enter` and the Python module for Google Translate API will be installed on your system.

If you have installed an Anaconda distribution of Python, you can install the API using the Anaconda Prompt. In this particular method, you will replace `pip` in the above command with `conda`, as shown in the following code snippet:

```
$ conda install googletrans
```

Now that you have installed the API, we will see it in action with the help of some examples.

Listing Supported Languages

The Google Translate API supports a variety of languages. To list all the supported languages, run the following script:

```
import googletrans

print(googletrans.LANGUAGES)
```

In the above example, you use the keyword `import` to import the `googletrans` module. Subsequently, you can list all the language names by printing the `LANGUAGES` attribute of the `googletrans` module.

When executed, the above piece of code will list all the supported languages names along with their shorthand notation. Here is how the output will look:



```
{'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', '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', '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'}
```

Basic Use

The most basic use of the Google Translate API is, of course, translating words or sentences from one language into another. To do so, we have to import the `Translator` class from `googletrans` module.

```
from googletrans import Translator
```

Next, you have to create an object of the `Translator` class.

```
translator = Translator()
```

Once the `Translator` class object is created, you will pass the text in source language as a parameter to the `translate()` method of the `Translator()` class object, as shown below:

```
result = translator.translate('Mitä sinä teet')
```

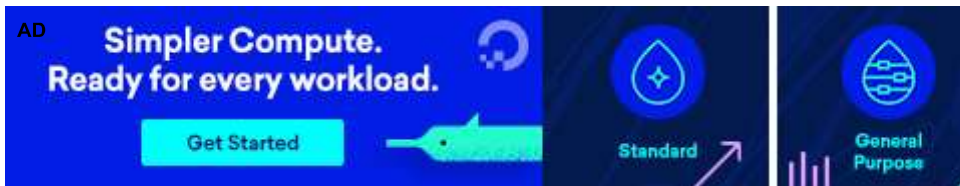
In the script above, we pass Finnish text to the `translate()` method.

The `translate()` method returns an object that contains information about the translated text, the source and destination languages and the pronunciation of the text. By default, the `translate()` method returns the English translation of the text passed to it. In our case, the object returned by the `translate()` method is stored in the `result` variable.

The object returned by the `translate()` method has the following attributes:

- `src` : The source language
- `dest` : Destination language, which is set to English (en)
- `origin` : Original text, that is 'Mitä sinä teet' in our example
- `text` : Translated text, that will be 'what are you doing?' in our case
- `pronunciation` : Pronunciation of the translated text

Let's print all the above attributes and see what output we get:



```
print(result.src)
print(result.dest)
print(result.origin)
print(result.text)
print(result.pronunciation)
```

Output:

```
fi
en
Mitä sinä teet
What are you doing
What are you doing
```

The output shows that the source language is Finnish (fi) and the destination language is English (en). The translated sentence can be printed via the `text` attribute.

In the above example, we did not specify the source language. Therefore, Google Translate API tries to detect source language itself. Similarly, we did not specify any destination language as well and thus, the API translated the source language into the default language that is English. But, what if you want to specify both the source and destination languages?

Specifying Source and Destination Languages

It is in fact, very easy to specify both the destination and source languages in the Google Translate API. Here is the code you'll use to pass only the source language:

```
result = translator.translate('Mikä on nimesi', src='fi')
```

For adding destination language only, you have to add `dest` attribute, followed by the language code:

```
result = translator.translate('Mikä on nimesi', dest='fr')
```

You can also pass the source and destination languages at the same time:

```
result = translator.translate('Mikä on nimesi', src='fi', dest='fr')
```

Let's now translate a Finnish sentence into French and then print the source and destination languages, as well as the translated text. This time we will specify the source and destination languages.

Subscribe to our Newsletter

Get occasional tutorials, guides, and jobs in your inbox. No spam ever.

Ad

Unsubscribe at any time.

Subscribe

```
from googletrans import Translator

translator = Translator()
result = translator.translate('Mikä on nimesi', src='fi', dest='fr')

print(result.src)
print(result.dest)
print(result.text)
```

The above piece of code will produce the following result.

```
fi
fr
Quel est votre nom
```

Translating List of Phrases

It is also possible to translate a *list* of textual phrases with the help of the Google Translate API. The basic process is the same as discussed above. You just have to pass the list containing the phrases as a parameter to the `translate()` method. This is useful for having a batch of phrases translated separately, but all in one API call.

Let's create a list of strings containing some phrases from the French language.

```
sentences = ['Bienvenu', 'Comment allez-vous', 'je vais bien']
```

Now it is time to call the `translate()` method and pass the list, the source language, and the destination language as parameters.

^

```
result = translator.translate(sentences, src='fr', dest='sw')
```

Ad

In the above script, the source language is French and the destination language is Swahili.

The `translate()` method returns a list of objects if you pass a list of phrases to it. Each object in the list returned by `translate()` method corresponds to each phrase in the input list that has to be translated. The best way to find the translation of each input phrase in the list is to iterate over the list of output objects. You can then use the `text`, `origin`, `src`, and other attributes of the individual objects to see the translation of individual phrases in the input list.

In the script below, we iterate over the list of objects returned by the `translate()` method and then print the origin and translated text:

```
for trans in result:  
    print(f'{trans.origin} -> {trans.text}')
```



The following will be the result displayed on the screen.

```
Bienvenu -> karibu  
Comment allez-vous -> Vipi wewe  
je vais bien -> Niko sawa
```

Translating Text Documents

You can also translate text documents via Google Translate API. All you have to do is to read the text file in Python using the `open` method, read the text and pass it to the `translate()` method.

The first step is to open the file in the "read" mode:

Ad

```
f = open('C:\\Users\\Khurram\\Desktop\\test.txt', 'r')
```

You can also check whether or not the file is in "read" mode using the `mode` property:

```
if f.mode == 'r':
```

Next, you can use the `f.read()` method to read the contents of the file. The contents of the file can be stored in any variable. In our case, the name of the variable will be `contents`.

We will also print the `contents` variable to check whether or not Python is properly reading the text file:

```
contents = f.read()
print(contents)
```

Here is the output of the file contents:

```
We are going to translate this text file using Python.
Subsequently, we will also translate it into French.
```

Make sure you have the above content in your text file if you want to follow along with our example.

We have ascertained that Python is accessing and reading the text file. Now, we will translate the result by importing the same old `Translate` class from before.

```
from googletrans import Translator

file_translate = Translator()
```

The next step is to pass the `contents` variable containing the input text to the `translate()` function. Finally, print the `text` attribute of the object returned by the `translate()` method and you will get the translated string.

^


```
result = translator.translate(contents, dest='fr')
print(result.text)
```

Ad

The output should look like the following:

```
Nous allons traduire ce fichier texte en Python.
Par la suite, nous le traduirons également en français.
```

To write the translated text to the same file, or a different text file, you will simply open the file in the write mode ("w"). Next, you need to call the `write()` method and pass it your translated text, as shown below:

```
with open('C:\\Users\\Khurram\\Desktop\\test_2.txt', 'w') as f:
    f.write(result.text)
```

In the above example, we have used the context manager (`/python-context-managers/`) with `with` to automatically open and close the write stream. Secondly, we have opened the file in the write mode. Lastly, we have used the `write()` method to write the translated string to a new file.

Conclusion

Google Translate is a tool with an API that helps you perform a lot of different translation-related functions. We have only scratched the surface with the above examples. You are encouraged to practice the API, as well as learn and understand how to use it in real-life applications.

 [python \(/tag/python/\)](#), [tool \(/tag/tool/\)](#)

20Google%20Translate%20API%20in%20Python&url=https://stackabuse.com/text-
translation-with-google-translate-api-in-python/)

rer/sharer.php?u=https://stackabuse.com/text-translation-with-google-translate-api-

Article?mini=true%26url=https://stackabuse.com/text-translation-with-google-
translate-api-in-python/&https://stackabuse.com)

^

Ad

AD



**Humiliated Indian Casino CEO
Fired After This Huge Mistake**

If you live in India and want to earn
some extra money easily, then this...

TodayPosts

[READ MORE](#)

Follow Us



(<https://twitter.com/StackAbuse>)



(<https://www.facebook.com/stackabuse/>)



(<https://stackabuse.com/rss/>)

Newsletter

Subscribe to our newsletter! Get occasional tutorials, guides, and reviews in your inbox.

No spam ever. Unsubscribe at any time.





Want a remote job?

Upcoming Paid Research Study Opportunities fo...

User Research International 14 hours ago

\$0K - \$0K

(<https://hireremote.io/remote-job/2337-upcoming-paid-research-study-opportunities-for-python-developers-at-user-research-international>)

[python](https://hireremote.io/remote-python-jobs) (<https://hireremote.io/remote-python-jobs>)

Analytics Engineer

Loom 1 day ago (<https://hireremote.io/remote-job/2323-analytics-engineer-at-loom>)

[python](https://hireremote.io/remote-python-jobs) (<https://hireremote.io/remote-python-jobs>) [data science](https://hireremote.io/remote-data-science-jobs) (<https://hireremote.io/remote-data-science-jobs>) [testing](https://hireremote.io/remote-testing-jobs) (<https://hireremote.io/remote-testing-jobs>) [healthcare](https://hireremote.io/remote-healthcare-jobs) (<https://hireremote.io/remote-healthcare-jobs>)

Senior Data Engineer II

Auth0 1 day ago (<https://hireremote.io/remote-job/2324-senior-data-engineer-ii-at-auth0>)

[senior](https://hireremote.io/remote-senior-jobs) (<https://hireremote.io/remote-senior-jobs>) [python](https://hireremote.io/remote-python-jobs) (<https://hireremote.io/remote-python-jobs>) [big data](https://hireremote.io/remote-big-data-jobs) (<https://hireremote.io/remote-big-data-jobs>) [aws](https://hireremote.io/remote-aws-jobs) (<https://hireremote.io/remote-aws-jobs>)

 [More jobs \(https://hireremote.io\)](https://hireremote.io)

Jobs via HireRemote.io (<https://hireremote.io>)

Prepping for an interview?

(<https://stackabu.se/daily-coding-problem>)

- Improve your skills by solving one coding problem every day
- Get the solutions the next morning via email
- Practice on **actual problems** asked by top companies, like:

`</>` Daily Coding Problem (<https://stackabu.se/daily-coding-problem>)

Recent Posts

Calculating Variance and Standard Deviation in Python (</calculating-variance-and-standard-deviation-in-python/>)

Automating AWS EC2 Management with Java (</automating-aws-ec2-management-with-java/>)

What's New in Tensorflow 2.0? (</whats-new-in-tensorflow-2/>)

Tags

[ai \(/tag/ai/\)](/tag/ai/)

[algorithms \(/tag/algorithms/\)](/tag/algorithms/)

[amqp \(/tag/amqp/\)](/tag/amqp/)

[angular \(/tag/angular/\)](/tag/angular/)

[announcements \(/tag/announcements/\)](/tag/announcements/)

[apache \(/tag/apache/\)](/tag/apache/)

[apache commons \(/tag/apache-commons/\)](/tag/apache-commons/)

[api \(/tag/api/\)](/tag/api/)

[arduino \(/tag/arduino/\)](/tag/arduino/)

[artificial intelligence \(/tag/artificial-intelligence/\)](/tag/artificial-intelligence/)

Follow Us



(<https://twitter.com/StackAbuse>)



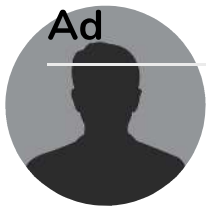
(<https://www.facebook.com/stackabuse>)



(<https://stackabuse.com/rss/>)

[Disclosure \(/disclosure\)](#) • [Privacy Policy \(/privacy-policy\)](#) • [Terms of Service \(/terms-of-service\)](#)

Ad



Ad

(/author/guest/)

About Guest Contributor (/author/guest/)

Twitter (<https://twitter.com/stackabuse>)

Subscribe to our Newsletter

Get occasional tutorials, guides, and jobs in your inbox. No spam ever. Unsubscribe at any time.

ALSO ON STACKABUSE

Binary Search in Java

4 months ago • 3 comments

Binary Search is a really simple yet effective searching algorithm. In ...

A Beginner-Level Introduction to ...

5 months ago • 2 comments

A Beginner-Level Introduction to MongoDB with Node.js Introduction ...

Variable-Length Arguments in ...

6 months ago • 1 comment

With a variable number of arguments in a function, we offer a flexible API to ...

5 Comments

StackAbuse

 Disqus' Privacy Policy

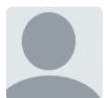
 Login ▾

 Recommend 1

 Tweet

 Share

Sort by Best ▾



LOG IN WITH

OR SIGN UP WITH DISQUS 



Name

Ad

**Raahim Khan** • 6 months ago

Thank you so much for this. I'm a beginner programmer and I'm learning about API's and i really helped. Thank you so much

2 ^ | v • Reply • Share ›

**Jonas zB** → Raahim Khan • 3 months ago

Same. This guid helped a lot. Any chance you've written an article about another google API?

^ | v • Reply • Share ›

**Ellie** • 2 months ago

Thanks for the explanations, I run my simple example and got the translation, but now the same code throws this error:

```
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
```

1 ^ | v • Reply • Share ›

**Vaishnavi** • 2 months ago

Sir, i have used this code to convert a telugu document to english but its not converting it into english and the file is in telugu itself.

^ | v • Reply • Share ›

**Pavan Kumar Konjeti** → Vaishnavi • a month ago

I am also having the same problem, after print(contexts) .. I am getting a blank space as output and the file stays as it is.

^ | v • Reply • Share ›

◀ Previous Post (/using-sqlalchemy-with-flask-and-postgresql/)

Next Post ▶ (/text-classification-with-bert-tokenizer-and-tf-2-0-in-python/)

Ad