Module (`gtts`)

- gTTS (`gtts.gTTS`)
- Languages (`gtts.lang`)
- Examples
- Playing sound directly
- Logging

gTTS (`gtts.gTTS`)

class **gtts.tts.gTTS**(*text, tld='com', lang='en', slow=False, lang_check=True, pre_processor_funcs=[<function tone_marks>, <function end_of_line>, <function abbreviations>, <function word_sub>], tokenizer_func=<bound method Tokenizer.run of re.compile('(?<=\\?).|(?<=!).|(?<=?).|(?<=!).|(?<!\\.[a-z])\\. |(?<!\\.[a-z]), |(?<!\\d):|\\)|—|\\(|\\)|。|¿|\\\n|¡|;|：|、|،|..|…|\\[|，', re.IGNORECASE) from: [<function tone_marks>, <function period_comma>, <function colon>, <function other_punctuation>]>*) [source]

gTTS – Google Text-to-Speech.

An interface to Google Translate's Text-to-Speech API.

Parameters

- **text** (*string*) – The text to be read.
- **tld** (*string*) – Top-level domain for the Google Translate host, i.e *https://translate.google.<tld>*. This is useful when `google.com` might be blocked within a network but a local or different Google host (e.g. `google.cn`) is not. Default is `com`.

- **lang** (*string, optional*) – The language (IETF language tag) to read the text in. Default is `en`.
- **slow** (*bool, optional*) – Reads text more slowly. Defaults to `False`.
- **lang_check** (*bool, optional*) – Strictly enforce an existing `lang`, to catch a language error early. If set to `True`, a `ValueError` is raised if `lang` doesn't exist. Setting `lang_check` to `False` skips Web requests (to validate language) and therefore speeds up instanciation. Default is `True`.
- **pre_processor_funcs** (*list*) – A list of zero or more functions that are called to transform (pre-process) text before tokenizing. Those functions must take a string and return a string. Defaults to:

```
[
    pre_processors.tone_marks,
    pre_processors.end_of_line,
    pre_processors.abbreviations,
    pre_processors.word_sub
]
```

- **tokenizer_func** (*callable*) – A function that takes in a string and returns a list of string (tokens). Defaults to:

```
Tokenizer([
    tokenizer_cases.tone_marks,
    tokenizer_cases.period_comma,
    tokenizer_cases.colon,
    tokenizer_cases.other_punctuation
]).run
```

See also

Pre-processing and tokenizing

Raises

- **AssertionError** – When `text` is `None` or empty; when there's nothing left to speak after pre-precessing, tokenizing and cleaning.
- **ValueError** – When `lang_check` is `True` and `lang` is not supported.
- **RuntimeError** – When `lang_check` is `True` but there's an error loading the languages dictionary.

`get_bodies`() [source]

Get TTS API request bodies(s) that would be sent to the TTS API.

Returns A list of TTS API request bodiess to make.

Return type list

`get_urls()` [source]

Get TTS API request URL(s) that would be sent to the TTS API.

Returns A list of TTS API request URLs to make.

This is particularly useful to get the list of URLs generated by `gTTS` but not yet fullfilled, for example to be used by an external program.

Return type list

`save(savefile)` [source]

Do the TTS API request and write result to file.

Parameters **savefile** (*string*) – The path and file name to save the `mp3` to.

Raises gTTSError – When there's an error with the API request.

`write_to_fp(fp)` [source]

Do the TTS API request(s) and write bytes to a file-like object.

Parameters **fp** (*file object*) – Any file-like object to

write the `mp3` to.

Raises

- **gTTSError** – When there's an error with the API request.
- **TypeError** – When `fp` is not a file-like object that takes bytes.

exception `gtts.tts.gTTSError`(*msg=None*, ***kwargs*) [source]

Exception that uses context to present a meaningful error message

`infer_msg`(*tts*, *rsp=None*) [source]

Attempt to guess what went wrong by using known information (e.g. http response) and observed behaviour

Languages (`gtts.lang`)

Note

The easiest way to get a list of available languages is to print them with `gtts-cli --all`

`gtts.lang.tts_langs`() [source]

Languages Google Text-to-Speech supports.

Returns

A dictionary of the type *{ '<lang>': '<name>'}*

Where *<lang>* is an IETF language tag such as *en* or *pt-br*, and *<name>* is the full English name of the language, such as *English* or *Portuguese (Brazil)*.

Return type dict

The dictionary returned combines languages from two origins:

- Languages fetched from Google Translate
- Languages that are undocumented variations that were observed to work and present different dialects or accents.

Examples

Write ‘hello’ in English to `hello.mp3`:

```
>>> from gtts import gTTS
>>> tts = gTTS('hello', lang='en')
>>> tts.save('hello.mp3')
```

Write ‘hello bonjour’ in English then French to `hello_bonjour.mp3`:

```
>>> from gtts import gTTS
>>> tts_en = gTTS('hello', lang='en')
>>> tts_fr = gTTS('bonjour', lang='fr')
>>>
>>> with open('hello_bonjour.mp3', 'wb') as f:
...     tts_en.write_to_fp(f)
...     tts_fr.write_to_fp(f)
```

Instead of writing to disk, get URL for ‘hello’ in English:

```
>>> from gtts import gTTS
>>> tts = gTTS('hello', lang='en')
>>> tts.get_urls()
['https://translate.google.com/translate_tts?ie=UTF-
8&q=hello&tl=en&ttsspeed=1&total=1&idx=0&client=tw-
ob&textlen=5&tk=316070.156329']
```

Playing sound directly

There’s quite a few libraries that do this. Write ‘hello’ to a file-like object to do further manipulation::

```
>>> from gtts import gTTS
>>> from io import BytesIO
>>>
>>> mp3_fp = BytesIO()
>>> tts = gTTS('hello', lang='en')
>>> tts.write_to_fp(mp3_fp)
>>>
>>> # Load `mp3_fp` as an mp3 file in
>>> # the audio library of your choice
```

Note

See Issue #26 for a discussion and examples of direct playback using various methods.

Note

Starting with `gTTS` 2.1.0, the `gtts.tts.gTTS.get_urls` method can be used to obtain the list of generated URLs requests (whithout fullfilling them) which could be used for playback in another program. See Examples above.

Logging

`gtts` does logging using the standard Python logging module. The following loggers are available:

`gtts.tts`
: Logger used for the `gTTS` class

`gtts.lang`
: Logger used for the `lang` module (language fetching)

`gtts`
: Upstream logger for all of the above