Quran Tafseer Python Package Documentation

Release 0.1.1

Emad Mokhtar

Apr 19, 2020

CONTENTS:

CHAPTER

ONE

QURAN TAFSEER PYTHON PACKAGE

Python Package (wrapper) for Quran Tafseer REST APIs

- Free software: MIT license
- Documentation: https://pytafseer.readthedocs.io.

1.1 Features

- Get list of Quran Tafseer/translation.
- Get list of Quran Tafseer/translation filtered by language.
- Get one verse tafseer/translation in a chapter.
- Get Multiple verses tafseer/translation in a chapter.

1.2 Settings (Environment Variable)

- `WEB_API_URL` -> Quran Tafseer API URL, default http://api.quran-tafseer.com/

1.3 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER

TWO

INSTALLATION

2.1 Stable release

To install Tafseer Python Package, run this command in your terminal:

```
$ pip install pytafseer
```

This is the preferred method to install Tafseer Python Package, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for Tafseer Python Package can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/emadmokhtar/pytafseer
```

Or download the tarball:

```
$ curl  -OL https://github.com/emadmokhtar/pytafseer/tarball/master
```

Once you have a copy of the source, you can install it with:

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

CHAPTER

THREE

USAGE

To use Quran Tafseer Python Package in a project

```
import pytafseer
```

3.1 List of Tafseer books

To get a list of tafseer books available at the REST APIs

```
from pytafseer import QuranTafseer

books = QuranTafseer.get_tafseer_books()
print(books)
>>>
[{'author': '  ',
  'book_name': ' ',
  'id': 1,
  'language': 'ar',
  'name': ' '},
 {'author': '    ',
  'book_name': ' ',
  'id': 2,
  'language': 'ar',
  'name': ' '},
 {'author': '         ',
  'book_name': '      ',
  'id': 3,
  'language': 'ar',
  'name': ' '},
 {'author': '       ',
  'book_name': '  ',
  'id': 4,
  'language': 'ar',
  'name': '  '},
 {'author': '  ',
  'book_name': '   ',
  'id': 5,
  'language': 'ar',
  'name': '  '},
 {'author': '     ',
  'book_name': ' ',
  'id': 6,
```

(continues on next page)

(continued from previous page)

```
  'language': 'ar',
  'name': ' '},
 {'author': '        ',
  'book_name': '   ',
  'id': 7,
  'language': 'ar',
  'name': ' '},
 {'author': '    ',
  'book_name': '     ',
  'id': 8,
  'language': 'ar',
  'name': ' '},
 {'author': 'A. J. Arberry',
  'book_name': 'The Koran Interpreted',
  'id': 9,
  'language': 'en',
  'name': 'Arberry'},
 {'author': 'Abdullah Yusuf Ali',
  'book_name': 'The Meaning of the Glorious Koran',
  'id': 10,
  'language': 'en',
  'name': 'Yusuf Ali'},
 {'author': 'Salomo Keyzer',
  'book_name': 'De Koran, voorafgegaan door het leven van Mahomet',
  'id': 11,
  'language': 'nl',
  'name': 'Keyzer'},
 {'author': 'Fred Leemhuis',
  'book_name': 'De Koran: Een weergave van de betekenis van de Ara',
  'id': 12,
  'language': 'nl',
  'name': 'Leemhuis'},
 {'author': 'Sofian S. Siregar',
  'book_name': 'De Edele Koran, en een vertaling van betekenissen',
  'id': 13,
  'language': 'nl',
  'name': 'Siregar'}]
```

3.2 List of Tafseer books for one language

To get a list of tafseer books for one language available at the REST APIs

```
from pytafseer import QuranTafseer

books = QuranTafseer.get_tafseer_books(language='en')
print(books)
>>>
 [{'author': 'A. J. Arberry',
'book_name': 'The Koran Interpreted',
'id': 9,
'language': 'en',
'name': 'Arberry'},
{'author': 'Abdullah Yusuf Ali',
'book_name': 'The Meaning of the Glorious Koran',
```

(continues on next page)

(continued from previous page)

```
'id': 10,
'language': 'en',
'name': 'Yusuf Ali'}]
```

3.3 One verse Tafseer text

To get tafseer for one verse in a chapter.

1. You should active one of the tafseer books.
2. Get the verse tafseer text from the book.

```
tafseer = QuranTafseer(book_id=10)  # activate tafseer book
verse_tafseer = tafseer.get_verse_tafseer(chapter_number=1,
                                          verse_number=1)
print(verse_tafseer)
>>>
{'ayah_number': 1,
'ayah_url': '/quran/1/1',
'tafseer_id': 10,
'tafseer_name': 'Yusuf Ali',
'text': 'In the name of Allah, Most Gracious, Most Merciful.'}
```

3.4 Range of verses in a chapter Tafseer text

To get tafseer range of verses in a chapter.

1. You should active one of the tafseer books.
2. Get multiple verses tafseer text from the book.

```
tafseer = QuranTafseer(book_id=10)  # activate tafseer book
verses_tafseer = tafseer.get_verses_tafseer(chapter_number=1,
                                            verse_number_from=1,
                                            verse_number_to=2)
print(verses_tafseer)
>>>
[{'ayah_number': 1,
'ayah_url': '/quran/1/1',
'tafseer_id': 10,
'tafseer_name': 'Yusuf Ali',
'text': 'In the name of Allah, Most Gracious, Most Merciful.'},
{'ayah_number': 2,
'ayah_url': '/quran/1/2',
'tafseer_id': 10,
'tafseer_name': 'Yusuf Ali',
'text': 'Praise be to Allah, the Cherisher and Sustainer of the worlds;'}]
```

3.5 Verse Tafseer text with its Quran text

Sometimes you want to get the verse Quran text with its tafseer text.

1. You should active one of the tafseer books.
2. Get the verse tafseer from the book, but pass extra argument.

```
tafseer = QuranTafseer(book_id=10)  # activate tafseer book
verse_tafseer = tafseer.get_verse_tafseer(chapter_number=1,
                                          verse_number=1,
                                          with_verse_text=True)
print(verse_tafseer['verse_text'])
>>>
{'ayah_number': 1,
'ayah_url': '/quran/1/1',
'tafseer_id': 10,
'tafseer_name': 'Yusuf Ali',
'text': 'In the name of Allah, Most Gracious, Most Merciful.',
'verse_text': '    '}
```

It's also working with getting the range.

1. You should active one of the tafseer books.
2. Get multiple verses tafseer from the book.

```
tafseer = QuranTafseer(book_id=10)  # activate tafseer book
verses_tafseer = tafseer.get_verses_tafseer(chapter_number=1,
                                            verse_number_from=1,
                                            verse_number_to=2,
                                            with_verse_text=True)
print(verses_tafseer[0]['verse_text'])
>>>
```

CHAPTER

FOUR

PYTAFSEER

4.1 pytafseer package

4.1.1 Submodules

4.1.2 pytafseer.settings module

4.1.3 Module contents

Top-level package for PyTafseer Python Package.

class pytafseer.**QuranTafseer**(*book_id: int*)
Bases: object

property book_details
Get the details of the tafseer book

classmethod get_tafseer_books(*language=''*) → list
Gets the list of available tafseer

Parameters **language** – filter the list of tafseer based on language, defaults, ISO 639-1 language optional.

Raises

- **ValueError** – raise Value error if the JSON return form the services is invalid. #noqa
- **Timeout** – if the server didn't return any response.
- **HTTPError** – if the server returned unsuccessful response.

Returns list of dictionary with tafseer attributes ['id', 'name', 'language', 'author', 'book_name']

get_verse_tafseer(*chapter_number: int*, *verse_number: int*, *with_verse_text: bool = False*) → dict
Gets the tafseer text for one verse

Parameters

- **with_verse_text** – Whether to load the verse Quran text or not.
- **chapter_number** – Chapter number.
- **verse_number** – Verse number or a start range.

get_verses_tafseer(*chapter_number: int*, *verse_number_from: int*, *verse_number_to: int*, *with_verse_text: bool = False*) → list

Gets the tafseer text for a range of verses

Parameters

- **with_verse_text** – Whether to load the verse Quran text or not.
- **chapter_number** – Chapter number.
- **verse_number_from** – Verse number start range.
- **verse_number_to** – Verse number end range.

CHAPTER

FIVE

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/Quran-Tafseer/pytafseer/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

5.1.4 Write Documentation

Tafseer Python Package could always use more documentation, whether as part of the official Tafseer Python Package docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/Quran-Tafseer/pytafseer/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *pytafseer* for local development.

1. Fork the *pytafseer* repo on GitHub.
2. Clone your fork locally:

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

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

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

4. Create a branch for local development:

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

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

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

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

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

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.4, 3.5 3.6 and 3.7, and for PyPy. Check https://travis-ci.org/emadmokhtar/pytafseer/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_pytafseer
```

5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

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

Travis will then deploy to PyPI if tests pass.

CHAPTER

SIX

CREDITS

6.1 Development Lead

- Emad Mokhtar <emad.m.habib@gmail.com>

6.2 Contributors

None yet. Why not be the first?

CHAPTER

SEVEN

HISTORY

7.1 0.1 (2019-01-20)

- First release on PyPI.

7.2 0.1.1 (2019-01-20)

- Fix missing requirements.

CHAPTER

EIGHT

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX

B

G

P

Q