
epubcheck

Release 0.4.2

Aug 07, 2019

Contents

1	Python wrappers for EpubCheck	1
2	Introduction	3
3	Installation	5
4	Quickstart	7
4.1	Command line usage examples	7
4.2	Using epubcheck as a python library	7
5	Documentation	9
6	Development	11
7	Credits	13
8	Installation	15
9	Usage	17
10	Reference	19
10.1	epubcheck	19
11	Contributing	21
11.1	Bug reports	21
11.2	Documentation improvements	21
11.3	Feature requests and feedback	21
11.4	Development	22
12	Authors	23
13	Changelog	25
13.1	0.4.2 (2019-08-07)	25
13.2	0.3.1 (2016-04-20)	25
13.3	0.3.0 (2016-04-10)	25
13.4	0.2.0 (2016-04-03)	25
13.5	0.1.0 (2016-04-01)	25

14 Indices and tables	27
Index	29

CHAPTER 1

Python wrappers for EpubCheck

CHAPTER 2

Introduction

The original [EpubCheck](#) is the standard Java based validation tool for EPUB maintained by [DAISY Consortium](#) on behalf of the [W3C](#), originally developed by the [IDPF](#).

This package provides a Python library and command line tool for convenient validation of EPUB files by wrapping the original [EpubCheck 4.2.2](#).

- Free software: BSD license

CHAPTER 3

Installation

If you have Python on your system you can do the usual:

```
pip install epubcheck
```

You must have Python & Java installed on your system. The original Java EpubCheck command line client itself is bundled in the [PyPi](#) package.

This package is tested with Python 2.7, 3.4, 3.5, 3.6, 3.7 on Linux and Windows. It should also work with PyPy.

4.1 Command line usage examples

Validata all epub files in the current directory:

```
$ epubcheck
```

Validate a single EPUB file:

```
$ epubcheck /path/to/book.epub
```

Validate all files in /epubfolder and create a detailed Excel report:

```
$ epubcheck /path/epubfolder --xls report.xls
```

Show command line help:

```
$ epubcheck -h
```

4.2 Using epubcheck as a python library

```
>>> from epubcheck import EpubCheck
>>> result = EpubCheck('src/epubcheck/samples/invalid.epub')
>>> print(result.valid)
>>> print(result.messages)
```


CHAPTER 5

Documentation

<https://epubcheck.readthedocs.org/>

CHAPTER 6

Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	<pre>set PYTEST_ADDOPTS=--cov-append tox</pre>
Other	<pre>PYTEST_ADDOPTS=--cov-append tox</pre>

CHAPTER 7

Credits

EpubCheck is a project coordinated by [IDPF](#). Most of the EpubCheck functionality comes from the schema validation tool [Jing](#) and schemas that were developed by [IDPF](#) and [DAISY](#). Initial EpubCheck development was largely done at [Adobe Systems](#).

CHAPTER 8

Installation

At the command line:

```
pip install epubcheck
```


CHAPTER 9

Usage

To use epubcheck in a project:

```
import epubcheck
```


10.1 epubcheck

class `epubcheck.EpubCheck` (*infile*, *lang=u'en'*, *profile=u'default'*, *autorun=True*)
Wraps an epubcheck task and provides results as native python objects.

Parameters

- **infile** (*str*) – path to epubfile to be checked
- **lang** (*str*) – set language for generated messages
- **profile** (*str*) – name of epubcheck profile to use
- **autorun** (*bool*) – whether to run the checking process on instantiation.

class `epubcheck.models.Checker`
Checker related information from epubcheck json data.

Parameters

- **path** (*str*) – Relative path to checked epub
- **filename** (*str*) – Filename of checked epub
- **checkerVersion** (*str*) – Version string of epubcheck
- **checkDate** (*str*) – When the epub was checked
- **elapsedTime** (*int*) – processing time
- **nFatal** (*int*) – number of fatal errors
- **nError** (*int*) – number of errors
- **nWarning** (*int*) – number of warnings
- **nUsage** (*int*) – number of usage messages

class `epubcheck.models.Meta`
EPUB metadata from *publication* key in epubcheck json data.

Parameters

- **publisher** (*str*) – name of publisher
- **title** (*str*) – title of ebook
- **creator** (*list[str]*) – list of creators
- **date** (*str*) – date of ebook
- **subject** (*list[str]*) – list of ebook subjects
- **description** (*str*) – description of ebook
- **rights** (*str*) –
- **identifier** (*str*) –
- **language** (*str*) – language of ebook
- **nSpines** (*int*) –
- **checksum** (*int*) –
- **renditionLayout** (*str*) –
- **renditionSpread** (*str*) –
- **ePubVersion** (*str*) –
- **isScripted** (*bool*) –
- **hasFixedFormat** (*bool*) –
- **isBackwardCompatible** (*bool*) –
- **hasAudio** (*bool*) –
- **charsCount** (*int*) –
- **embeddedFonts** (*list[str]*) –
- **refFonts** (*list[str]*) –
- **hasEncryption** (*bool*) –
- **hasSignatures** (*bool*) –
- **contributors** (*list[str]*) –

class epubcheck.models.**Message**

A Validation message representing a single error condition.

Parameters

- **id** (*str*) – Error type id (ex: “OPF-049”)
- **level** (*str*) – Severity of message (ex: “ERROR”)
- **location** (*str*) – Location of error (ex: <file>:<line>:<column>)
- **message** (*str*) – Description of the error condition
- **suggestion** (*str*) – How to resolve error condition

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

11.1 Bug reports

When [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.

11.2 Documentation improvements

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

11.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/titusz/epubcheck/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 code contributions are welcome :)

11.4 Development

To set up *epubcheck* for local development:

1. Fork [epubcheck](#) (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/epubcheck.git
```

3. Create a branch for local development:

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

Now you can make your changes locally.

4. When you’re done making changes, run all the checks, doc builder and spell checker with `tox` one command:

```
tox
```

5. 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
```

6. Submit a pull request through the GitHub website.

11.4.1 Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)¹.
2. Update documentation when there’s new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

11.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

¹ If you don’t have all the necessary python versions available locally you can rely on Travis - it will [run the tests](#) for each change you add in the pull request.
It will be slower though ...

CHAPTER 12

Authors

- Titusz Pan - <https://github.com/titusz>
- Sean Quinn - <https://github.com/swquinn>

13.1 0.4.2 (2019-08-07)

- Update the epubcheck.jar to v4.2.2 (see: <https://github.com/w3c/epubcheck/releases/tag/v4.2.2>)

13.2 0.3.1 (2016-04-20)

- Added custom PY2/PY3 compat module and removed dependancy on six

13.3 0.3.0 (2016-04-10)

- Add commandline support with Excel batch reporting
- Moved development status from Alpha to Beta

13.4 0.2.0 (2016-04-03)

- EpubCheck results as native python objects
- More documentation

13.5 0.1.0 (2016-04-01)

- First release on PyPI.

CHAPTER 14

Indices and tables

- `genindex`
- `modindex`
- `search`

C

Checker (*class in epubcheck.models*), [19](#)

E

EpubCheck (*class in epubcheck*), [19](#)

M

Message (*class in epubcheck.models*), [20](#)

Meta (*class in epubcheck.models*), [19](#)