The Open API Specifications Advanced Python Introspection library

Release 0.1.16

Mar 03, 2020

Contents

CHAPTER 1

Overview

docs	
tests	
package	

Python library for Web APIs leveraging OpenAPI/Swagger specification, enabling you to:

- validate an OAS 2.0 document
- prune an OAS 2.0 document of its unused elements
- filter endpoints of an OAS 2.0 document to generate a subset of the API
- [todo] detect changes between two OAS 2.0 documents (what's new, changed, deprecated, removed)
- [todo] merge multiple OAS 2.0 documents together
- [todo] rate/score quality of an OAS 2.0 document (documentation coverage, good practices,. . .)
- [todo] transform an OAS 2.0 document (add x-. . . items, rename paths, rewrite basePath, . . .)
- [todo] add/remove securityDefinitions on an OAS 2.0 document

Free software license: BSD 3-Clause License

1.1 Quickstart

Install oasapi from PyPI with:

```
pip install oasapi
```

You can also install the in-development version with:

```
pip install https://github.com/sdementen/oasapi/archive/master.zip
```

1.1.1 OAS Document validation

Validate an OAS 2.0 Document (in JSON or YAML format) with:

```
python -m oasapi validate samples/swagger_petstore.json
```

or if you prefer with the oasapi script:

```
oasapi validate samples/swagger_petstore.json
```

or with the `sdementen/oasapi` Docker image (available on Docker Hub):

```
docker run sdementen/oasapi validate http://petstore.swagger.io/v2/swagger.json
```

You can also pipe a swagger to the command (if oasapi cannot retrieve the file by itself):

```
type samples/swagger_petstore.json | oasapi validate -
type samples/swagger_petstore.json | docker run -i sdementen/oasapi validate -
curl http://petstore.swagger.io/v2/swagger.json | oasapi validate -
curl -s http://petstore.swagger.io/v2/swagger.json | docker run -i sdementen/oasapi␣
↪validate -
```

1.1.2 OAS Document filtering

Similarly, you can use the filter command to filter an OAS 2.0 document and keep only specific operations. For instance, to keep only operations with the tag ‘pet’, the security scope at most ‘read:pets’ and the operations matching the regepx “(GET|PUT) .*petid.*” (all paths containing ‘petid’ with verb GET or PUT), run:

```
oasapi filter http://petstore.swagger.io/v2/swagger.json -o new_swagger.json --tag␣
↪pet --security-scope read:pets --path "(GET|PUT) .*petid.*"
```

(you will see that the new_swagger.json has only one operation left `GET /pet/{petId}`).

1.1.3 OAS Document pruning

Another useful command is pruning, which takes an OAS 2.0 document and prune it of its unused elements as:

```
oasapi prune http://petstore.swagger.io/v2/swagger.json -o new_swagger.json
```

1.2 Documentation

https://oasapi.readthedocs.io/

1.3 Development

https://oasapi.readthedocs.io/en/latest/contributing.html#development

CHAPTER 2

Installation

At the command line:

```
pip install oasapi
```

CHAPTER 3

Usage with the CLI

3.1 Introduction

oasapi offers a command line interface (CLI) to run core operations:

```
$ python -m oasapi
Usage: python -m oasapi [OPTIONS] COMMAND [ARGS]...

  These are common operations offered by the oasapi library

Options:
  --help  Show this message and exit.

Commands:
  filter    Filter the SWAGGER operations based on tags, operation path or...
  prune     Prune from the SWAGGER unused global...
  validate  Validate the SWAGGER according to the specs.
```

All these operation are also available programmatically through the *oasapi* package.

Alternatively to the syntax herebove, you can call oasapi through the oasapi script:

```
$ oasapi
Usage: oasapi [OPTIONS] COMMAND [ARGS]...

  These are common operations offered by the oasapi library

Options:
  --help  Show this message and exit.

Commands:
  filter    Filter the SWAGGER operations based on tags, operation path or...
  prune     Prune from the SWAGGER unused global...
  validate  Validate the SWAGGER according to the specs.
```

And there is also a docker image `sdementen/oasapi` offering the same script through `docker run sdementen/oasapi`

Help is available with the `--help` option:

```
$ oasapi --help
Usage: oasapi [OPTIONS] COMMAND [ARGS]...

  These are common operations offered by the oasapi library

Options:
  --help  Show this message and exit.

Commands:
  filter    Filter the SWAGGER operations based on tags, operation path or...
  prune     Prune from the SWAGGER unused global...
  validate  Validate the SWAGGER according to the specs.
```

```
$ oasapi validate --help
Usage: oasapi validate [OPTIONS] SWAGGER

  Validate the SWAGGER according to the specs.

  SWAGGER is the path to the swagger file, in json or yaml format. It can be
  a file path, an URL or a dash (-) for the stdin

Options:
  -v, --verbose          Make the operation more talkative
  -s, --silent           Do not print the oasapi messages to stderr
  -o, --output FILENAME  Path to write the resulting swagger ('-' for stdout)
  --help                 Show this message and exit.
```

3.2 Specifying an OAS 2.0 (aka swagger) file

The *oasapi* commands will often require an OAS 2.0 Document (aka swagger). The swagger can be given in JSON or YAML format and can be a local file or a URL.

Example of usage (YAML file)

```
$ oasapi validate samples/swagger_petstore.yaml
The swagger is valid.
```

Example of usage (JSON file):

```
$ oasapi validate samples/swagger_petstore.json
The swagger is valid.
```

Example of usage (JSON URL)

```
$ oasapi validate http://petstore.swagger.io/v2/swagger.json
The swagger is valid.
```

3.3 Pipelining commands

When using oasapi in pipes, the `-` denotes the stdin/stdout.

To pipe a swagger in a command, you replace the path/URL of the swagger with a `-`:

```
$ curl -s http://petstore.swagger.io/v2/swagger.json | oasapi validate -
The swagger is valid.
```

To send the output swagger of a command to stdout, you replace the path for the `--output` argument with a `-` (when using stdout, the format is always YAML).

If you want to silence the commands (ie not sending their message to stderr), you can add the silent argument (`-s`)

For instance, the following command will:

1. get a swagger with curl
2. filter it to keep only the operations with the tag ‘pet’
3. prune it of any unused elements
4. validate it and send it to stdout

```
$ curl -s http://petstore.swagger.io/v2/swagger.json | oasapi filter -s --tag pet - --
↪output - | oasapi prune -s - --output - | oasapi validate -s - -o -
swagger: '2.0'
info:
  description: 'This is a sample server Petstore server.  You can find out more about
    Swagger at [http://swagger.io](http://swagger.io) or on [irc.freenode.net,
↪#swagger](http://swagger.io/irc/).  For
    this sample, you can use the api key `special-key` to test the authorization␣
↪filters.'
  version: 1.0.3
  title: Swagger Petstore
  termsOfService: http://swagger.io/terms/
  contact:
    email: apiteam@swagger.io
...
```

3.4 Validating an OAS 2.0 Document

Validating is an operation that will check the swagger for errors:

- structural errors, i.e. errors coming from the swagger not complying with the swagger JSON schema
- semantic errors, i.e. errors beyond the structural ones (e.g. duplicate operationIds)

You can validate a document with the `validate` command:

```
$ oasapi validate --help
Usage: oasapi validate [OPTIONS] SWAGGER

  Validate the SWAGGER according to the specs.

  SWAGGER is the path to the swagger file, in json or yaml format. It can be
  a file path, an URL or a dash (-) for the stdin

```

(continues on next page)

(continued from previous page)

```
Options:
  -v, --verbose          Make the operation more talkative
  -s, --silent           Do not print the oasapi messages to stderr
  -o, --output FILENAME  Path to write the resulting swagger ('-' for stdout)
  --help                 Show this message and exit.
```

```
$ oasapi validate samples/swagger_petstore.json
The swagger is valid.
```

```
$ oasapi validate samples/swagger_petstore_with_errors.json
The swagger is not valid. Following 6 errors have been detected:
- Duplicate operationId @ 'paths./pet/findByStatus.get.operationId' -> the␣
↪operationId 'updatePet' is already used in an endpoint.
- Json schema validator error @ 'info' -> 'notvalidinfo' does not match any of the␣
↪regexes: '^x-'
- Json schema validator error @ 'paths./pet.post' -> 'responses' is a required␣
↪property
- Json schema validator error @ 'paths./pet/findByStatus.get.security.0.petstore_auth
↪' -> 1 is not of type 'array'
- Json schema validator error @ 'schemes.1' -> 'ftp' is not one of ['http', 'https',
↪'ws', 'wss']
- Security scope not found @ 'paths./pet.put.security.[0].petstore_auth.think:pets' ->
↪ scope think:pets is not declared in the scopes of the securityDefinitions
↪'petstore_auth'
```

3.5 Filtering an OAS 2.0 Document

Filtering is an operation that will keep from the swagger only the operations that do match criteria:

- tags: the operation should have at least one tag from a given list of tags (e.g. [“pet”, “store”])
- operations: the VERB + PATH should match a regexp from a list (e.g. [“POST /pet”, “(GET|PUT) /pet/{petId}”])
- security scopes: the operation should be accessible only with the scopes in a given list of scopes (e.g. [“read:pets”])

You can filter a document with the `filter` command:

```
$ oasapi filter --help
Usage: oasapi filter [OPTIONS] SWAGGER

  Filter the SWAGGER operations based on tags, operation path or security
  scopes.

  SWAGGER is the path to the swagger file, in json or yaml format. It can be
  a file path, an URL or a dash (-) for the stdin

Options:
  -v, --verbose               Make the operation more talkative
  -s, --silent                Do not print the oasapi messages to stderr
  -o, --output FILENAME       Path to write the resulting swagger ('-' for
                              stdout)
  -t, --tag TEXT              A tag to keep
```

(continues on next page)

(continued from previous page)

```
  -p, --path TEXT              A path to keep
  -sc, --security-scope TEXT  A security scope to keep
  --help                       Show this message and exit.
```

```
$ oasapi filter samples/swagger_petstore.json -t pet -t store -sc read:pets -p "POST /
↪pet" -p "(GET|PUT) /pet/{petId}" -o swagger_filtered.yaml
The swagger has filtered or removed the following 19 operations:
- Operation removed as no filter matched. @ 'paths./pet.post' -> The operation has␣
↪been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./pet.put' -> The operation has␣
↪been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./pet/findByStatus.get' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./pet/findByTags.get' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./pet/{petId}.delete' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./pet/{petId}.post' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./pet/{petId}/uploadImage.post' ->␣
↪The operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./store/inventory.get' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./store/order.post' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./store/order/{orderId}.delete' ->␣
↪The operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./store/order/{orderId}.get' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./user.post' -> The operation has␣
↪been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./user/createWithArray.post' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./user/createWithList.post' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./user/login.get' -> The operation␣
↪has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./user/logout.get' -> The operation␣
↪has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./user/{username}.delete' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./user/{username}.get' -> The␣
↪operation has been removed as it does not match any filter.
- Operation removed as no filter matched. @ 'paths./user/{username}.put' -> The␣
↪operation has been removed as it does not match any filter.
```

As the `filter` command may remove operations, it is a good idea to follow it with a `prune` command to remove any elements of the swagger that would not be used anymore.

For instance, the following command filter the swagger to keep only operations with the tag ‘weird’ and prune the resulting swagger afterwards. As no operation has the tag ‘weird’, the filtering leads to a swagger with no more paths and the pruning will clean the swagger showing at the end an almost empty swagger.

```
$ oasapi filter samples/swagger_petstore.json -t weird -o - 2> filter_messages |␣
↪oasapi prune - -o - 2> prune_messages
swagger: '2.0'
```

(continues on next page)

(continued from previous page)

```
info:
  description: 'This is a sample server Petstore server.  You can find out more about
    Swagger at [http://swagger.io](http://swagger.io) or on [irc.freenode.net,
↪#swagger](http://swagger.io/irc/).  For
    this sample, you can use the api key `special-key` to test the authorization␣
↪filters.'
  version: 1.0.0
  title: Swagger Petstore
  termsOfService: http://swagger.io/terms/
  contact:
    email: apiteam@swagger.io
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
host: petstore.swagger.io
basePath: /v2
schemes:
- https
- http
paths: {}
externalDocs:
  description: Find out more about Swagger
  url: http://swagger.io
```

The operation must match all the three different filter criteria (tags, security scopes and operations regexp) when given. If you want to apply more advanced filter (like "(tag='pet' AND security-scope='read:pets') or (tag='store')"), you can call the filter method directly from python and pass these filters (see `oasapi.filter()`).

3.6 Pruning an OAS 2.0 Document

Pruning is an operation that will 'clean' the swagger by removing any unused elements:

- global definitions not referenced
- global parameters not referenced
- global responses not referenced
- securityDefinitions not used
- securityDefinitions oauth2 scopes not used
- tags not used
- empty paths (endpoints with no verbs attached)

You can prune a document with the `prune` command:

```
$ oasapi prune --help
Usage: oasapi prune [OPTIONS] SWAGGER

  Prune from the SWAGGER unused global definitions/responses/parameters,
  unused securityDefinition/scopes, unused tags and unused paths.

  SWAGGER is the path to the swagger file, in json or yaml format. It can be
  a file path, an URL or a dash (-) for the stdin

```

(continues on next page)

(continued from previous page)

```
Options:
  -v, --verbose          Make the operation more talkative
  -s, --silent           Do not print the oasapi messages to stderr
  -o, --output FILENAME  Path to write the resulting swagger ('-' for stdout)
  --help                 Show this message and exit.
```

```
$ oasapi prune samples/swagger_petstore.json
The swagger had no unused elements.
```

```
$ oasapi prune samples/swagger_petstore_unused_elements.json
The swagger has been pruned of 5 elements:
- Oauth2 scope removed @ 'securityDefinitions.petstore_auth.scopes.write:pets' ->␣
↪oauth2 scope not used
- Path is empty @ 'paths./store/inventory' -> path '/store/inventory' has no␣
↪operations defined
- Reference filtered out @ 'definitions.User' -> reference not used
- Security definition removed @ 'securityDefinitions.api_key' -> security definition␣
↪not used
- Tag definition removed @ 'tags.[2]' -> tag definition for 'user' not used
```

CHAPTER 4

Reference

4.1 oasapi

The oasapi packages

`oasapi.validate(swagger)`

Validate a swagger specification.

The validations checks the following points:

- validate against re. OAS 2.0 schema
- no missing reference
- unicity of operationId
- no missing securityDefinition
- consistency of parameters (default value vs type)

Parameters **swagger** (`Dict`) – the swagger spec

Return type `Tuple[Dict, List[ValidationError]]`

Returns a set of errors

`oasapi.prune(swagger)`

Prune a swagger specification.

The pruning removed from the swagger the following elements:

- unused global definitions/responses/parameters
- unused securityDefinition/scopes
- unused tags
- empty paths (i.e. endpoints with no verbs)

Parameters **swagger** (Dict) – the swagger spec

Return type Tuple[Dict, List[FilterAction]]

Returns pruned swagger, a set of actions

oasapi.**filter**(*swagger*, *mode='keep_only'*, *conditions=None*)

Filter endpoints of a swagger specification.

The endpoints can be filtered according to two modes:

- keep_only: it will keep only the operations matching any of the conditions
- remove: it will remove only the operations matching any of the conditions (TO BE IMPLEMENTED)

The conditions parameter is a list of FilterCondition objects containing each:

- tags: the operation is kept only if it has at least one tag in the tags
- operations: the operation is kept only if its VERB + PATH matches at least one operation in the operations
- security_scopes: the operation is kept only if it requires no security or if some of its security items only requires the scopes in the security_scopes

Any of these fields can be None to avoid matching on the field criteria.

Parameters

- **mode** –
- **conditions** (Optional[List[FilterCondition]]) –
- **swagger** (Dict) – the swagger spec

Return type Tuple[Dict, List[FilterAction]]

Returns filtered swagger, a set of actions

class oasapi.events.**Event**(*path*, *reason*, *type*)

Bases: object

Base class for an event (an error, an action, . . .).

static format_path(*path*)

Format a path to a JSON Path alike string

Return type str

path = None

the path in the dictionary to which the even relates

reason = None

the reason of the event

type = None

the string representation of the type of event

class oasapi.events.**Error**(*path*, *reason*, *type*)

Bases: *oasapi.events.Event*

Base class for an error

class oasapi.events.**ValidationError**(*path*, *reason*, *type*)

Bases: *oasapi.events.Error*

Base class for a validation error (used in the swagger validation)

class oasapi.events.**ParameterDefinitionValidationError**(*path*, *reason*, *parameter_name*, *type='Parameter definition error'*)

Bases: *oasapi.events.ValidationError*

An error on a parameter definition

class oasapi.events.**ReferenceNotFoundValidationError**(*path*, *reason*, *type='Reference not found'*)

Bases: *oasapi.events.ValidationError*

An error on a reference used but not found

class oasapi.events.**SecurityDefinitionNotFoundValidationError**(*path*, *reason*, *type='Security definition not found'*)

Bases: *oasapi.events.ValidationError*

An error on a securityDefinition used but not found

class oasapi.events.**OAuth2ScopeNotFoundInSecurityDefinitionValidationError**(*path*, *reason*, *type='Security scope not found'*)

Bases: *oasapi.events.ValidationError*

An error on an OAuth2 scope used but not found

class oasapi.events.**DuplicateOperationIdValidationError**(*path*, *reason*, *operationId*, *path_already_used*, *type='Duplicate operationId'*)

Bases: *oasapi.events.ValidationError*

An error on two operations using the same operationId

operationId = None

the name of the duplicate operationId

path_already_used = None

the path of the first operation using the operationId

class oasapi.events.**JsonSchemaValidationError**(*path*, *reason*, *type='Json schema validator error'*)

Bases: *oasapi.events.ValidationError*

An error due to an invalid schema

CHAPTER 5

Contributing

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

5.1 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.

5.2 Documentation improvements

The Open API Specifications Advanced Python Introspection library could always use more documentation, whether as part of the official The Open API Specifications Advanced Python Introspection library docs, in docstrings, or even on the web in blog posts, articles, and such.

5.3 Feature requests and feedback

The best way to send feedback is to file an issue at https://github.com/sdementen/oasapi/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 :)

5.4 Development

To set up *oasapi* for local development:

1. Fork oasapi (look for the “Fork” button).
2. Clone your fork locally:

   ```
   git clone git@github.com:sdementen/oasapi.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 and docs builder 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.

5.4.1 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`.

5.4.2 Tips

To install a minimal virtual environment with tox (see https://docs.python.org/3/tutorial/venv.html#creating-virtual-environments)

```
cd path-to-your-oasapi-folder
python -m venv .env
' activate your .env virtualenv
python -m pip install -r requirements-dev.txt
```

To install the git pre-commit scripts:

[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 . . .

```
pre-commit install
```

To run the tests locally:

```
cd path-to-your-oasapi-folder
' activate your .env virtualenv
tox
```

To build the documentation locally (available in the folder `docs/dist`, entry point `docs/dist/index.html`):

```
cd path-to-your-oasapi-folder
' activate your .env virtualenv
tox -e docs
```

To recreate the tox environments (e.g. if you add a dependency in the setup.py):

```
tox --recreate
tox --recreate -e py36              '(only the py36 environment)
```

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

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

```
detox
```

5.4.3 Tips with PyCharm

To run tox within PyCharm, right click on `tox.ini` and choose `Run` (see https://www.jetbrains.com/help/pycharm/tox-support.html)

5.4.4 Tips to deploy (for the maintainers)

To bump the version:

```
' update/commit first all your changes including the changelog
bump2version patch --tag --commit
```

To build the source distribution:

```
' clean first the /dist folder
python setup.py sdist
```

To upload on PyPI Test:

```
python -m twine upload --repository-url https://test.pypi.org/legacy/ dist/*
```

To upload on PyPI:

```
python -m twine upload dist/*
```

For the setup of the deploy to PyPI step on Travis, the information on https://docs.travis-ci.com/user/deployment/pypi/ (with the online encrypt tool on https://travis-encrypt.github.io/) were useful.

CHAPTER 6

Authors

- Sebastien de Menten - https://github.com/sdementen

CHAPTER 7

Changelog

7.1 dev

- fix integer response code raising exception (closes #14)

7.2 0.1.16 (2020-02-08)

- add documentation of filter in prune + cleanup doc

7.3 0.1.15 (2020-02-08)

- add filtering of swagger
- support outputting swagger in yaml format
- support silencing the CLI
- [dev] refactor CLI

7.4 0.1.13 (2020-01-25)

- add validation of parameters required/default
- add validation of parameters default value wrt type/format

7.5 0.1.12 (2020-01-17)

- add pruning of unused items (definitions, responses, parameters, securityDefinitions/scopes, tags)

- rename validate_swagger function to validate, add prune function

7.6 0.1.11 (2020-01-16)

- [dev] use jsonpath_ng to walk the swagger
- add timing of validation in CLI if verbose

7.7 0.1.10 (2020-01-10)

- [dev] fix tag name to remove “v” (for readthedocs latest build)
- [dev] fix travis ci python version for doc

7.8 0.1.9 (2020-01-10)

- Improve validation of array parameters
- Improve documentation

7.9 0.1.8 (2020-01-09)

- Push the sdementen/oasapi docker image to Docker Hub on each release
- Update doc on Docker image use and pipeing a swagger to oasapi

7.10 0.1.7 (2020-01-09)

- Support URL and stdin (with -) as SWAGGER for the CLI

7.11 0.1.6 (2020-01-08)

- [dev] PyPI deployment through Travis CI

7.12 0.1.5 (2020-01-08)

- Fix script form of the cli (`oasapi` instead of `python -m oasapi`)

7.13 0.1.4 (2020-01-08)

- Explicit support only for python >= 3.6 (no py35 as use of f-string)
- [dev] Move local build of docs from dist/docs to docs/dist to avoid cluttering dist

7.14 0.1.3 (2020-01-08)

- [dev] Add git pre-commit hooks for black and flake
- [dev] Do not use isort
- [dev] Fix black + flake8 issues

7.15 0.1.2 (2020-01-08)

- Fix model & reporting of duplicate operationIds
- Add documentation on the CLI Usage

7.16 0.1.1 (2020-01-08)

- Add pyyaml dependencies (to support OAS in yaml format)

7.17 0.1.0 (2020-01-08)

- First release on PyPI.
- Implementation of the validation of an OAS 2.0 (aka swagger) file

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index