
doschema Documentation

Release 1.0.0a2.dev20161018

CERN

Oct 18, 2016

1	User's Guide	3
1.1	Installation	3
1.2	Usage	3
2	API Reference	5
2.1	API Docs	5
3	Additional Notes	7
3.1	Contributing	7
3.2	Changes	9
3.3	License	9
3.4	Authors	9
	Python Module Index	11

JSON Schema utility functions and commands.

This is an experimental developer preview release.

- Free software: GPLv2 license
- Documentation: <https://doschema.readthedocs.io/>

This part of the documentation will show you how to get started in using Invenio-Base.

1.1 Installation

Invenio's DoSchema module is on PyPI so all you need is:

```
$ pip install doschema
```

1.2 Usage

JSON Schema utility functions and commands.

1.2.1 Compatibility Validation

It validates compatibility between different JSON schemas versions.

A schema is backward compatible if the fields' type remain the same in all JSON schemas declaring it and JSON schemas are type consistent within themselves too.

```
>>> import json
>>> from io import open
>>>
>>> import doschema.validation
>>> from doschema.utils import detect_encoding
>>>
>>> schemas = [
...     './examples/jsonschema_for_repetition.json',
...     './examples/jsonschema_repetition.json'
... ]
>>>
>>> schema_validator = doschema.validation.JSONSchemaValidator()
>>> for schema in schemas:
...     with open(schema, 'rb') as infile:
...         byte_file = infile.read()
...         encoding = detect_encoding(byte_file)
...         string_file = byte_file.decode(encoding)
```

```
...     json_schema = json.loads(string_file)
...     schema_validator.validate(json_schema, schema)
```

By default the index of “array” “items” are ignored. Thus all the values of an array should have the same type in order to be compatible. This behavior can be disabled by setting “ignore_index = False” in the validator’s constructor.

```
>>> import json
>>> from io import open
>>>
>>> import doschema.validation
>>> from doschema.utils import detect_encoding
>>>
>>> schemas = [
...     './examples/jsonschema_with_index_option.json'
... ]
>>>
>>> schema_validator = doschema.validation.JSONSchemaValidator(
...     ignore_index = False
... )
>>> for schema in schemas:
...     with open(schema, 'rb') as infile:
...         byte_file = infile.read()
...         encoding = detect_encoding(byte_file)
...         string_file = byte_file.decode(encoding)
...         json_schema = json.loads(string_file)
...         schema_validator.validate(json_schema, schema)
```

1.2.2 CLI usage

```
$ doschema validate jsonschema_for_repetition.json \
jsonschema_repetition.json
$ doschema validate jsonschema_with_index_option.json --with_index
```

API Reference

If you are looking for information on a specific function, class or method, this part of the documentation is for you.

2.1 API Docs

2.1.1 doschema

Additional Notes

Notes on how to contribute, legal information and changes are here for the interested.

3.1 Contributing

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

3.1.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/inveniosoftware/doschema/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.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/inveniosoftware/doschema/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 :)

3.1.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/doschema.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 doschema
$ cd doschema/
$ pip install -e .[all]
```

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 tests:

```
$ ./run-tests.sh
```

The tests will provide you with test coverage and also check PEP8 (code style), PEP257 (documentation), flake8 as well as build the Sphinx documentation and run doctests.

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

```
$ git add .
$ git commit -s -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.

3.1.3 Pull Request Guidelines

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

1. The pull request should include tests and must not decrease test coverage.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring.

3. The pull request should work for Python 2.7, 3.3, 3.4 and 3.5. Check https://travis-ci.com/inveniosoftware/doschema/pull_requests and make sure that the tests pass for all supported Python versions.

3.2 Changes

Version v1.0.0a1 (released 2016-10-12)

- Initial public release.

3.3 License

Invenio is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Invenio is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Invenio; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

In applying this license, CERN does not waive the privileges and immunities granted to it by virtue of its status as an Intergovernmental Organization or submit itself to any jurisdiction.

3.4 Authors

JSON Schema utility functions and commands.

- CERN <info@inveniosoftware.org>

d

`doschema`, [3](#)

D

`doschema` (module), [3](#)