# JSON:API ORM Documentation

***Release 0.1.8***

**Mislav Cimperšak**

**Apr 10, 2018**

# Contents:


**1 JSON:API ORM** **1**
- 1.1 How To . . . 1
- 1.2 Caveats . . . 2

**2 Indices and tables** **7**

**Python Module Index** **9**

CHAPTER 1

# JSON:API ORM

Quick and dirty ORM that maps JSON:API responses to object attributes.

- Free software: BSD license
- Documentation: https://jsonapi-orm.readthedocs.io.
- Requirements: Python 3.5+

## 1.1 How To

Use Requests or (if you are a masochist) Python's built-in urllib modules to make the request to your JSON:API service and from there pass the response to JSON:API ORM.

So, first install requests and this lib:

```
pip install requests
pip install jsonapi-orm
```

Switch to your Python code and use the magic!

```
import requests
from jsonapi_orm import response_to_obj


# list of items
r = requests.get('https://raw.githubusercontent.com/mislavcimpersak/jsonapi-orm/
↪master/tests/responses/example_list.json')
obj = response_to_obj(r.json())

print('LIST OF ITEMS:')
for item in obj.data:
    print(item.title)
    # author is defined as a relationship
```

(continues on next page)

(continued from previous page)

```
    print(item.author.twitter)


# single item
r = requests.get('https://raw.githubusercontent.com/mislavcimpersak/jsonapi-orm/
↪master/tests/responses/example_single.json')
obj = response_to_obj(r.json())

print('SINGLE ITEM')
print(obj.data.title)
# author is defined as a relationship
print(obj.data.author.id)
print(obj.data.author.twitter)
```

## 1.2 Caveats

- Since Python object attribute names have certain rules like not starting with a number or not containing "-" char, all such attributes can be accessed using `.get()` method. Ie. `obj.data.author.get('first-name')`.
- If relationship is not described in more detail in the `included` part of the response matching fails silently.
- For now, this lib does not lazily follow relationship links or anything like that. You can of course make a new request to the given link and pass that response to JSON:API ORM.
- For now, there is no check if response is a valid JSON:API response. But you'll probably get that you are trying to parse an invalid response when things start to break.
- And last, this lib requires Python 3.5 or newer.

### 1.2.1 Installation

#### Stable release

To install JSON:API ORM, run this command in your terminal:

```
$ pip install jsonapi-orm
```

This is the preferred method to install JSON:API ORM, 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.

#### From sources

The sources for JSON:API ORM can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/mislavcimpersak/jsonapi-orm
```

Or download the tarball:

```
$ curl  -OL https://github.com/mislavcimpersak/jsonapi-orm/tarball/master
```

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

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

### 1.2.2 jsonapi_orm

#### jsonapi_orm package

#### Submodules

#### jsonapi_orm.jsonapi_orm module

Main module.

jsonapi_orm.jsonapi_orm.**mapped_included**(*inc: dict*) → dict

Makes included more easily searchable by creating a dict that has (type,id) pair as key.

Also, returned value for the key can be used for relationships node.

ie. {

**('people', '12'): {** 'id': '12, 'type': 'people', 'attributes': {

'name': 'John'

}

}, ('tag', '11'): {

...

}

}

jsonapi_orm.jsonapi_orm.**resolve**(*response: dict*) → dict

jsonapi_orm.jsonapi_orm.**resolve_single_data_item**(*data: dict*) → dict

jsonapi_orm.jsonapi_orm.**response_to_obj**(*response: dict*) → munch.Munch

#### Module contents

Top-level package for JSON:API ORM.

### 1.2.3 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:

#### Types of Contributions

#### Report Bugs

Report bugs at https://github.com/mislavcimpersak/jsonapi-orm/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" and "help wanted" is open to whoever wants to implement it.

### Implement Features

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

### Write Documentation

JSON:API ORM could always use more documentation, whether as part of the official JSON:API ORM 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/mislavcimpersak/jsonapi-orm/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 :)

### Get Started!

Ready to contribute? Here's how to set up *jsonapi-orm* for local development.

1. Fork the *jsonapi-orm* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/jsonapi-orm.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 jsonapi-orm
$ cd jsonapi-orm/
$ 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 jsonapi_orm 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.

### 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 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/mislavcimpersak/jsonapi-orm/pull_requests and make sure that the tests pass for all supported Python versions.

### Tips

To run a subset of tests:

```
$ py.test tests.test_jsonapi_orm
```

### 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.

### 1.2.4 History

#### 0.1.8 (2018-04-06)

- Added test config and basic tests.

#### 0.1.7 (2018-03-25)

- Using readme.rst for frontpage of the docs.

#### 0.1.7 (2018-03-25)

- Fixed pip install.

#### 0.1.6 (2018-03-25)

- Readme fix for pypi.

#### 0.1.5 (2018-03-25)

- Readme fix for pypi.

#### 0.1.4 (2018-03-25)

- Readme examples using slighty modified examples from jsonapi.org which are located in this repo.

#### 0.1.3 (2018-03-25)

- Fixed a bug when child data is non-existent.

#### 0.1.2 (2018-03-25)

- Fixed a bug when included is not present in response.

#### 0.1.0 (2018-03-24)

- First release on PyPI.

CHAPTER 2

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

## j

jsonapi_orm, 3
jsonapi_orm.jsonapi_orm, 3

# Index

## J

jsonapi_orm (module), 3
jsonapi_orm.jsonapi_orm (module), 3

## M

mapped_included() (in module jsonapi_orm.jsonapi_orm), 3

## R

resolve() (in module jsonapi_orm.jsonapi_orm), 3
resolve_single_data_item() (in module jsonapi_orm.jsonapi_orm), 3
response_to_obj() (in module jsonapi_orm.jsonapi_orm), 3