drf-swagger-customization Documentation
Release 0.1.6

Release 0.1.6

Andres Rojano Ruiz

Sep 03, 2017

Contents

Contents:

drf-swagger-customization

This is a django app which you can modify and improve your autogenerated swagger documentation from your drf API.

Documentation

The full documentation is at https://drf-swagger-customization.readthedocs.io.

Quickstart

Install drf-swagger-customization:

```
pip install drf-swagger-customization
```

Add these global variables to your settings.py:

```
EXTENSION_PATH = os.path.join(PROJECT_DIR, 'docs/doc_extension.json') # Path to your
↪extension file
EXTERNAL_DOC_FOLDER = os.path.join(PROJECT_DIR, 'docs/external/') # Path to your
↪external documentation folder
```

Add drf-swagger-customization's URL patterns:

```python
from drf_swagger_customization.views import get_swagger_view

schema_view = get_swagger_view(title='Pastebin API')

urlpatterns = [
    ...
    url(r'^docs/$', schema_view),
    ...
]
```

Usage

With this package we can increase the auto-generated documentation from django-swagger. That way, we can add documentation from external APIs or add more information to our drf API methods such as fields, remove endpoints, update attributes, and so on.

In order to add information to our EXTENSION_PATH json file, we have available these operations:

Create

```
{
    "action": "create",
    "operation": {
        "paths|/v1/travels/|get|parameters": [
            {
              "name": "Field1",
              "in": "query",
              "required": true,
              "type": "string"
            },
            {
              "name": "Field2",
              "in": "path",
              "required": true,
              "type": "integer"
            }
        ]
    }
}
```

Update

```
{
    "action": "update",
    "operation": {
        "paths|/v1/travels/|get|parameters|field1": {
            "name": "Field1",
            "in": "query",
            "required": true,
            "type": "string"
        }
    }
}
```

Remove

```
{
    "action": "delete",
```

```
    "operation": "paths|/v1/travels/|get|parameters|field1"
}
```

Rename

```
{
    "action": "rename",
    "operation": {
        "paths|/v1/travels/": "/travels/",
        "paths|/v1/travels/{id}/": "/travels/{id}/"
    }
}
```

Completed Sample

```
[
  {
    "action": "create",
    "operation": {
      "paths|/v1/travels/|get|parameters": [
        {
          "name": "Field1",
          "in": "query",
          "required": true,
          "type": "string"
        },
        {
          "name": "Field2",
          "in": "path",
          "required": true,
          "type": "integer"
        }
      ]
    }
  },
  {
    "action": "update",
    "operation": {
      "paths|/v1/travels/|get|parameters|field1": {
        "name": "Field1",
        "in": "query",
        "required": true,
        "type": "string"
      }
    }
  },
  {
    "action": "delete",
    "operation": "paths|/v1/travels/|get|parameters|field1"
  },
  {
    "action": "rename",
    "operation": {
```

```
            "paths|/v1/travels/": "/travels/",
            "paths|/v1/travels/{id}/": "/travels/{id}/"
        }
    }
]
```

Future Work

- Add an example

- Command for doc_file customization, in order to remove redundant executions.

- Improve the settings options.

Installation

At the command line:

```
$ easy_install drf-swagger-customization
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv drf-swagger-customization
$ pip install drf-swagger-customization
```

Usage

With this package we can increase the auto-generated documentation from django-swagger. That way, we can add documentation from external APIs or add more information to our drf API methods such as fields, remove endpoints, update attributes, and so on.

Add these global variables to your settings.py:

```python
EXTENSION_PATH = os.path.join(PROJECT_DIR, 'docs/doc_extension.json') # Path to your
↪extension file
EXTERNAL_DOC_FOLDER = os.path.join(PROJECT_DIR, 'docs/external/') # Path to your
↪external documentation folder
```

Add drf-swagger-customization's URL patterns:

```python
from drf_swagger_customization.views import get_swagger_view

schema_view = get_swagger_view(title='Pastebin API')

urlpatterns = [
    ...
    url(r'^docs/$', schema_view),
    ...
]
```

Building the JSON File

In order to add information to our EXTENSION_PATH json file, we have available these operations:

Create:

```json
{
    "operation": "create",
    "swagger-data": {
        "paths|/v1/travels/|get|parameters": [
            {
                "name": "Field1",
                "in": "query",
                "required": true,
                "type": "string"
            },
            {
                "name": "Field2",
                "in": "path",
                "required": true,
                "type": "integer"
            }
        ]
    }
}
```

Update:

```json
{
    "operation": "update",
    "swagger-data": {
        "paths|/v1/travels/|get|parameters|field1": {
            "name": "Field1",
            "in": "query",
            "required": true,
            "type": "string"
        }
    }
}
```

Remove:

```json
{
    "action": "delete",
    "operation": "paths|/v1/travels/|get|parameters|field1"
}
```

Rename

```json
{
    "action": "rename",
    "operation": {
        "paths|/v1/travels/": "/travels/",
        "paths|/v1/travels/{id}/": "/travels/{id}/"
    }
}
```

Completed Sample

```json
[
  {
    "action": "create",
    "operation": {
      "paths|/v1/travels/|get|parameters": [
        {
          "name": "Field1",
          "in": "query",
          "required": true,
          "type": "string"
        },
        {
          "name": "Field2",
          "in": "path",
          "required": true,
          "type": "integer"
        }
      ]
    }
  },
  {
    "action": "update",
    "operation": {
      "paths|/v1/travels/|get|parameters|field1": {
        "name": "Field1",
        "in": "query",
        "required": true,
        "type": "string"
      }
    }
  },
  {
    "action": "delete",
    "operation": "paths|/v1/travels/|get|parameters|field1"
  },
  {
    "action": "rename",
    "operation": {
      "paths|/v1/travels/": "/travels/",
      "paths|/v1/travels/{id}/": "/travels/{id}/"
    }
  }
]
```

CHAPTER 4

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/androiz/drf-swagger-customization/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

drf-swagger-customization could always use more documentation, whether as part of the official drf-swagger-customization 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/androiz/drf-swagger-customization/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 *drf-swagger-customization* for local development.

1. Fork the *drf-swagger-customization* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/drf-swagger-customization.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 drf-swagger-customization
$ cd drf-swagger-customization/
$ 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 drf_swagger_customization tests
$ python setup.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 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/androiz/drf-swagger-customization/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ python -m unittest tests.test_drf_swagger_customization
```

Credits

Development Lead

- Andres Rojano Ruiz <androiz10@gmail.com>

Contributors

None yet. Why not be the first?

History

0.1.6 (2017-09-03)

- Fix in documentation

0.1.5 (2017-05-31)

- Added new action "Rename"

0.1.4 (2017-05-30)

- Bugfix with documentation

0.1.3 (2017-05-30)

- Bugfix with documentation

0.1.2 (2017-05-30)

- Improved documentation in order to explain how to build the json file

0.1.1 (2017-05-25)

- Fix in methods add, update and remove documentation.

0.1.0 (2017-05-24)

• First release on PyPI.