
Durga Documentation

Release 0.1.0

transcode

January 05, 2015

1	Installation	3
2	Usage	5
3	durga	7
3.1	durga package	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	10
4.4	Tips	11
5	Authors	13
5.1	Development Lead	13
5.2	Contributors	13
6	Changelog	15
6.1	0.1.0 (2014-12-09)	15
7	Indices and tables	17
	Python Module Index	19

Contents:

Installation

At the command line:

```
$ pip install durga
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv durga  
$ pip install durga
```

Usage

To use Durga in a project define a class that extends `durga.resource.Resource`. This example uses the Flickr API `flickr.photos.search`:

```
1 import durga
2
3
4 class FlickrResource(durga.Resource):
5     base_url = 'https://api.flickr.com/services'
6     name = 'rest'
7     results_path = ('photos', 'photo')
8     schema = durga.schema.Schema({
9         'farm': durga.schema.Use(int),
10        'id': durga.schema.Use(int),
11        'isfamily': durga.schema.Use(bool),
12        'isfriend': durga.schema.Use(bool),
13        'ispublic': durga.schema.Use(bool),
14        'owner': durga.schema.And(basestring, len),
15        'secret': durga.schema.And(basestring, len),
16        'server': durga.schema.Use(int),
17        'title': durga.schema.And(basestring, len),
18    })
19     query = {
20         'method': 'flickr.photos.search',
21         'api_key': 'a33076a7ae214c0d12931ae8e38e846d',
22         'format': 'json',
23         'nojsoncallback': 1,
24     }
```

Note: For convenience `durga.resource.Resource` and the `schema` library are available at the top module level.

Now you can search for the first 10 cat images:

```
FlickrResource().collection.filter(text='Cat', per_page=10)
```

This will return a `durga.collection.Collection` with a `durga.element.Element` for each result.

3.1 durga package

3.1.1 Submodules

3.1.2 durga.collection module

```
class durga.collection.Collection (url, resource)  
    Bases: object  
  
    all ()  
    count ()  
    create (data)  
    delete ()  
    filter (*args, **kwargs)  
    get (*args, **kwargs)  
    get_element (data)  
    order_by ()  
    update (data)  
    validate (data)
```

3.1.3 durga.element module

```
class durga.element.Element (resource, data)  
    Bases: object  
  
    delete ()  
    get_raw ()  
    get_resource ()  
    get_url ()  
    update (data)
```

3.1.4 durga.resource module

```
class durga.resource.Resource
    Bases: object
    get_url()
```

3.1.5 Module contents

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:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at the [GitHub issue tracker](#).

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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at the [GitHub issue tracker](#).

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 :)

4.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/durga.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 durga
$ cd durga/
$ pip install -e ./
```

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 pep8, pyflakes and the tests, including testing other Python versions with tox:

```
$ make test
$ make test-all
```

To get *pytest*, *pep8*, *pyflakes* and *tox*, just pip install them into your virtualenv:

```
$ pip install -r requirements.txt
```

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.

4.3 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 and 3.4. Check [Travis CI](#) and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ make test TEST_ARGS='-k <EXPRESSION>'
```

Authors

5.1 Development Lead

- Markus Zapke-Gründemann

5.2 Contributors

None yet. Why not be the first?

Changelog

6.1 0.1.0 (2014-12-09)

- First release on PyPI.

Indices and tables

- *genindex*
- *modindex*
- *search*

d

`durga`, 8
`durga.collection`, 7
`durga.element`, 7
`durga.resource`, 8

A

`all()` (`durga.collection.Collection` method), 7

C

`Collection` (class in `durga.collection`), 7

`count()` (`durga.collection.Collection` method), 7

`create()` (`durga.collection.Collection` method), 7

D

`delete()` (`durga.collection.Collection` method), 7

`delete()` (`durga.element.Element` method), 7

`durga` (module), 8

`durga.collection` (module), 7

`durga.element` (module), 7

`durga.resource` (module), 8

E

`Element` (class in `durga.element`), 7

F

`filter()` (`durga.collection.Collection` method), 7

G

`get()` (`durga.collection.Collection` method), 7

`get_element()` (`durga.collection.Collection` method), 7

`get_raw()` (`durga.element.Element` method), 7

`get_resource()` (`durga.element.Element` method), 7

`get_url()` (`durga.element.Element` method), 7

`get_url()` (`durga.resource.Resource` method), 8

O

`order_by()` (`durga.collection.Collection` method), 7

R

`Resource` (class in `durga.resource`), 8

U

`update()` (`durga.collection.Collection` method), 7

`update()` (`durga.element.Element` method), 7

V

`validate()` (`durga.collection.Collection` method), 7