Cloudify GraphQL Documentation

Release 0.0.0

Javier Collado

Jul 07, 2017

Contents

Contents:

CHAPTER 1

Cloudify GraphQL

Experimental Cloudify GraphQL API

- Free software: MIT license
- Documentation: https://cloudify-graphql.readthedocs.io.

Description

This project is a flask application that implements a GraphQL API that wraps the Cloudify REST API to provide a different way to access data with the benefits of GraphQL. As explained in GraphQL's main page:

- Ask for what you need, get exactly that
- Get many resources in a single request

Usage

To give a try to this API, launch a Cloudify manager with its REST API and then the GraphQL API as follows:

```
FLASK_APP=server.py flask run
```

Note that for this to work the manager IP address needs to be configured in the GraphQL API application. By default, as it can be seen in `config.py`, this is set to `172.20.0.2` which is the default IP address of the docker container when running Cloudify through docl.

Once the GraphQL API is running queries can be sent to it using any tool such as curl:

```
curl -X POST \
    -u admin:admin \
    -H 'Tenant: default_tenant' \
    -H 'Content-Type: application/graphql' \
    http://localhost:5000/graphql \
    -d '{ ping }'
```

where the `Authorization` and `Tenant` headers are passed directly to the Cloudify REST API and are required for authentication purposes.

The GraphiQL tool is also available by opening `http://localhost:5000/graphql` in a web browser. Note, however, that the browser also needs to send the `Authorization` and `Tenant` headers with proper values, so that the GraphQL API can interact successfully with the REST API. To do that, a browser extension such as Modify Header Value can be used. Keep in mind that the `Authorization` header value can be generated easily with something like:

```
echo "Basic $(echo -n 'admin:admin' | base64)"
```

Query Examples

- Get all tenant names

```
{
  tenants {
    name
  }
}
```

- Get all blueprints and their deployments

```
{
  blueprints {
    id
    deployments {
      id
    }
  }
}
```

- Get all deployments and the blueprints they belong to

```
{
  deployments {
    id
    blueprint {
      id
    }
  }
}
```

- Get all executions and their events and logs

```
{
  executions {
    logs {
      message
    }
    events {
      message
```

```
        }
    }
}
```

Mutation Examples

- Create new tenant

  mutation {

  createTenant(name: “newTenant”) { statusCode tenant {

  name

  }

  }

  }

- Delete tenant

  mutation {

  deleteTenant(name: “newTenant”) { statusCode tenant {

  name

  }

  }

  }

Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

Stable release

To install Cloudify GraphQL, run this command in your terminal:

```
$ pip install cloudify_graphql
```

This is the preferred method to install Cloudify GraphQL, 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 Cloudify GraphQL can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/jcollado/cloudify_graphql
```

Or download the tarball:

```
$ curl  -OL https://github.com/jcollado/cloudify_graphql/tarball/master
```

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

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

CHAPTER 3

Usage

To use Cloudify GraphQL in a project:

```python
import cloudify_graphql
```

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/jcollado/cloudify_graphql/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

Cloudify GraphQL could always use more documentation, whether as part of the official Cloudify GraphQL 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/jcollado/cloudify_graphql/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 *cloudify_graphql* for local development.

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

   ```
   $ git clone git@github.com:your_name_here/cloudify_graphql.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 cloudify_graphql
   $ cd cloudify_graphql/
   $ 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 cloudify_graphql 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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/jcollado/cloudify_graphql/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_cloudify_graphql
```

CHAPTER 5

Credits

Development Lead

- Javier Collado <javier@gigaspaces.com>

Contributors

None yet. Why not be the first?

CHAPTER 6

History

0.0.0 (2017-06-16)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search