bitpy

Release 0.1.0

Sep 27, 2017

Contents

Python bindings fror Bitbucket’s rest API.

CHAPTER 1

Installing

Installation is simple with pip:

```
pip install bitpy
```

Note: Project not yet available on PyPi.

Building from source

You can also build manually from the source for more control. First obtain a copy of the source by either downloading the zipball or cloning the public repository:

```
git clone git@bitbucket.org:caalle/bitpy.git
```

Then you can build and install the package into your current Python site-packages folder:

```
python setup.py install
```

Alternatively, just build locally and manage yourself:

```
python setup.py build
```

Building documentation from source

To build the documentation from source:

```
python setup.py build_sphinx
```

Then view in your browser:

```
file:///path/to/bitpy/build/doc/html/index.html
```

Running tests against the source

With a copy of the source it is also possible to run the unit tests:

```
python setup.py test
```

Dependencies

- Python >= 2.6, < 3
- requests == 2.4.3

Additional For building

- Sphinx >= 1.2.2, < 2
- sphinx_rtd_theme >= 0.1.6, < 1

Additional For testing

- Pytest >= 2.3.5

CHAPTER 2

Using

These examples assumes that you have already downloaded and installed bitpy.

Getting a repository

First you need to create a new bitpy client. To do this bitpy requires your Bitbucket username and password.

```python
import bitpy

client = bitpy.Client(<username>, <password>)
```

Once you've got a client you can use that to retrive repositories. The example below will get the bitpy repository.

```python
import bitpy

client = bitpy.Client(<username>, <password>)

bitpy_repository = client.get_repository(name='bitpy', account='caalle')
```

Manage pull requests

When you've retrived a repository you can both list and create pull requests. To list all open pull requests for the repository do something like:

```python
import bitpy

client = bitpy.Client(<username>, <password>)

bitpy_repository = client.get_repository(name='bitpy', account='caalle')
```

```
for pull_request in bitpy_repository.list_pull_requests():
    print(pull_request._title)
```

To approve and merge a specific pull request pass in the id as argument.

```
import bitpy

client = bitpy.Client(<username>, <password>)

bitpy_repository = client.get_repository(name='bitpy', account='caalle')

pull_request = bitpy_repository.get_pull_request(pull_request_id=123)

pull_request.approve()
pull_request.merge()
```

You can also create new pull requests by using the *create_pull_request* method.

```
import bitpy

client = bitpy.Client(<username>, <password>)

bitpy_repository = client.get_repository(name='bitpy', account='caalle')

pull_request = bitpy_repository.create_pull_request(
    title='A title of the pull request',
    description='''Longer description that might include *markdown*''',
    source_branch='feature/branch', target_branch='master'
)

print(pull_request.get_link()) # Output http link to pull request.
```

Teams and Members

When creating pull requests you might want to add people as reviewers. Let’s create a pull request and add the user ‘caalle’ as reviewer.

```
import bitpy

client = bitpy.Client(<username>, <password>)

member = client.get_member('caalle')

bitpy_repository = client.get_repository(name='bitpy', account='caalle')

pull_request = bitpy_repository.create_pull_request(
    title='Pull request that needs to be reviewed.',
    description='''Longer description that might include *markdown*''',
    source_branch='feature/branch', target_branch='master',
    reviewers=[member]
)
```

Note: In a real world scenario the user who creates the pull request cannot be added as a reviewer.

If you have a team on Bitbucket and want to use it when assigning to pull requests you can do something like this:

```python
import random
import bitpy

client = bitpy.Client(<username>, <password>)

team = client.get_team('my_bitbucket_team')
team_members = team.get_members()

reviewer = [random.choice(team_members)]

bitpy_repository = client.get_repository(name='bitpy', account='caalle')

pull_request = bitpy_repository.create_pull_request(
    title='Pull request with random reviewer.',
    description='''Longer description that might include *markdown*''',
    source_branch='feature/branch', target_branch='master',
    reviewers=[reviewer]
)
```

This will assign a random user from the team as reviewer for my newly created pull request.

CHAPTER 3

API reference

`bitpy`

CHAPTER 4

Glossary

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

CHAPTER 6

Copyright and license