

---

# **Bitdock**

***Release 0.1.0***

August 07, 2014



|          |                                |           |
|----------|--------------------------------|-----------|
| <b>1</b> | <b>Installation</b>            | <b>3</b>  |
| 1.1      | Building from source . . . . . | 3         |
| 1.2      | Dependencies . . . . .         | 3         |
| <b>2</b> | <b>Usage</b>                   | <b>5</b>  |
| 2.1      | Mapping Users . . . . .        | 6         |
| <b>3</b> | <b>API reference</b>           | <b>9</b>  |
| 3.1      | bitdock . . . . .              | 9         |
| <b>4</b> | <b>Glossary</b>                | <b>11</b> |
| <b>5</b> | <b>Indices and tables</b>      | <b>13</b> |
| <b>6</b> | <b>Copyright and license</b>   | <b>15</b> |
|          | <b>Python Module Index</b>     | <b>17</b> |



Intermediate service to handle [Bitbucket](#) 'pull-request' notifications and push them into [Flowdock](#)



---

## Installation

---

Installation is simple with `pip`:

```
$ pip install bitdock
```

### 1.1 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:ftrack/bitdock.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
```

#### 1.1.1 Building documentation from source

To build the documentation from source:

```
python setup.py build_sphinx
```

Then view in your browser:

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

### 1.2 Dependencies

- Python `>= 2.6, < 3`
- CherryPy `>= 3.2, < 4`
- requests `>= 2.2, < 3'`

### 1.2.1 Additional For testing

- `Pytest >= 2.3.5, < 3`

---

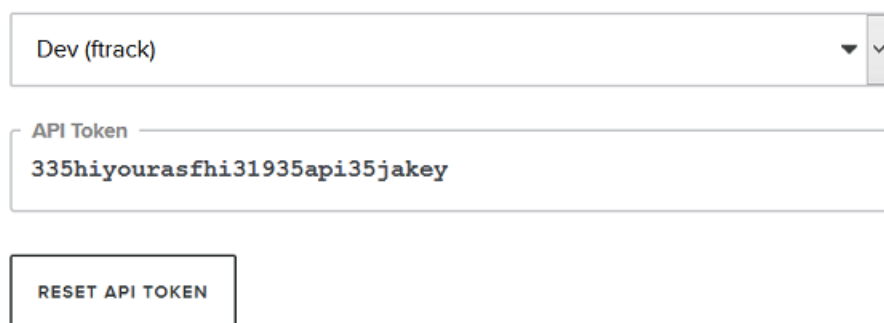
## Usage

---

Once *installed* you need to grab a relevant API token from Flowdock. Go to <https://www.flowdock.com/account/tokens> and copy an appropriate token (for example, a token for a flow called 'Dev'):

### Flow API tokens

With a flow's API token you can post team inbox or chat messages into the specific flow.



Dev (ftrack) ▼

API Token

335hiyourasfhi31935api35jakey

RESET API TOKEN

Then start the Bitdock service from the command line passing your copied API token as the sole parameter:

```
python -m bitdock [YOUR_FLOWDOCK_API_KEY]
```

---

**Note:** To see additional runtime options (such as setting host and port interface) use:

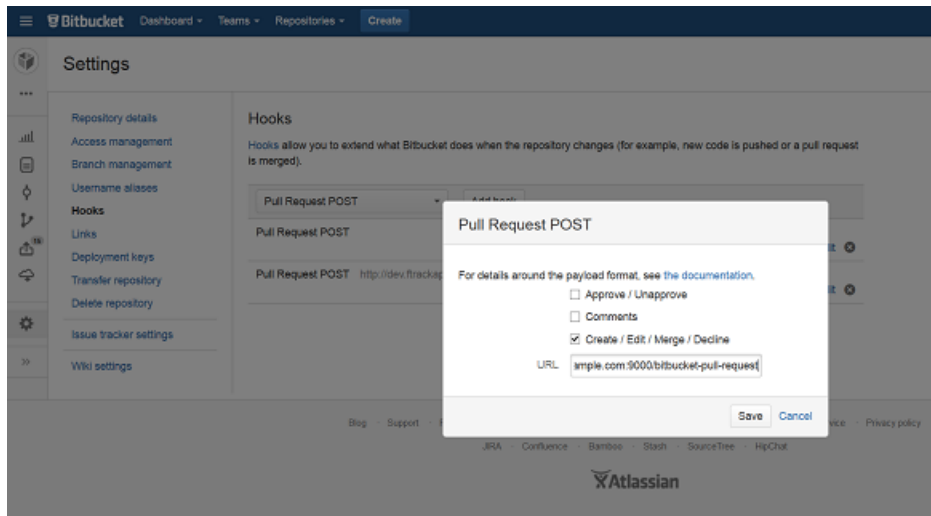
```
python -m bitdock --help
```

---

Now the server is up and running it is time to head over to the Bitbucket project you want to receive notifications for. Once there, navigate to the *Settings* → *Hooks* page and add a new *Pull Request POST* hook.

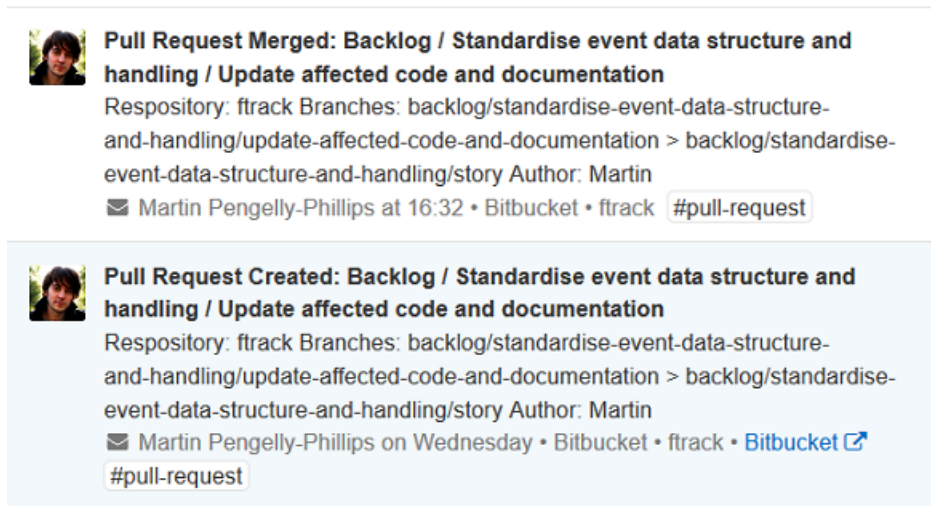
Check only the *Create / Edit / Merge / Decline* option and then enter the full public URL to your running server adding '/bitbucket-pull-request' at the end. For example:

```
http://example.com:9000/bitbucket-pull-request
```



Click *Save*.

Now whenever a Bitbucket pull request is created or updated you should see a new entry in your Flowdock inbox for the configured flow.



**Note:** Assigned reviewers will show up as people tags on the inbox entry.

## 2.1 Mapping Users

As folks may have different user details on Bitbucket and Flowdock there is a basic way to map from one to the other. Start the server passing in a path to a JSON file containing mappings of Bitbucket usernames to Flowdock user details. For example:

my\_mappings.json

```
{
  "martin": {
    "username": "mphilips",
    "display_name": "Martin Pengelly-Phillips",
    "email": "martin@example.com"
  }
}
```

```
    },  
    ...  
}
```

Run the server using:

```
python -m bitdock --user-mapping my_mappings.json
```



---

## API reference

---

### 3.1 bitdock

**class** `bitdock.__main__.Bitdock` (*flowdock\_api\_key*, *user\_mapping=None*)

Bases: `object`

Bitdock main handler.

**\_\_init\_\_** (*flowdock\_api\_key*, *user\_mapping=None*)

Initialise handler with *flowdock\_api\_key*.

In addition, optionally specify a *user\_mapping* to use when mapping users. It should be a dictionary of Bitbucket user names against Flowdock user dictionaries. The Flowdock user dictionary should contain keys for *username*, *display\_name* and *email*.

**bitbucket\_pull\_request** ()

Handle bitbucket pull request notification.

**flowdock\_api\_url**

Return flowdock api url.

`bitdock.__main__.main` (*arguments=None*)

Bitdock: Connect Bitbucket and Flowdock.



---

**Glossary**

---



---

## Indices and tables

---

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



---

## Copyright and license

---

Copyright (c) 2014 ftrack

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this work except in compliance with the License. You may obtain a copy of the License in the LICENSE.txt file, or at:

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.



—  
bitdock.\_\_main\_\_, 9

**b**  
bitdock, 9