



UBC
LAUNCH
PAD



Rocket 2

The Bad News

It's not working (yet)

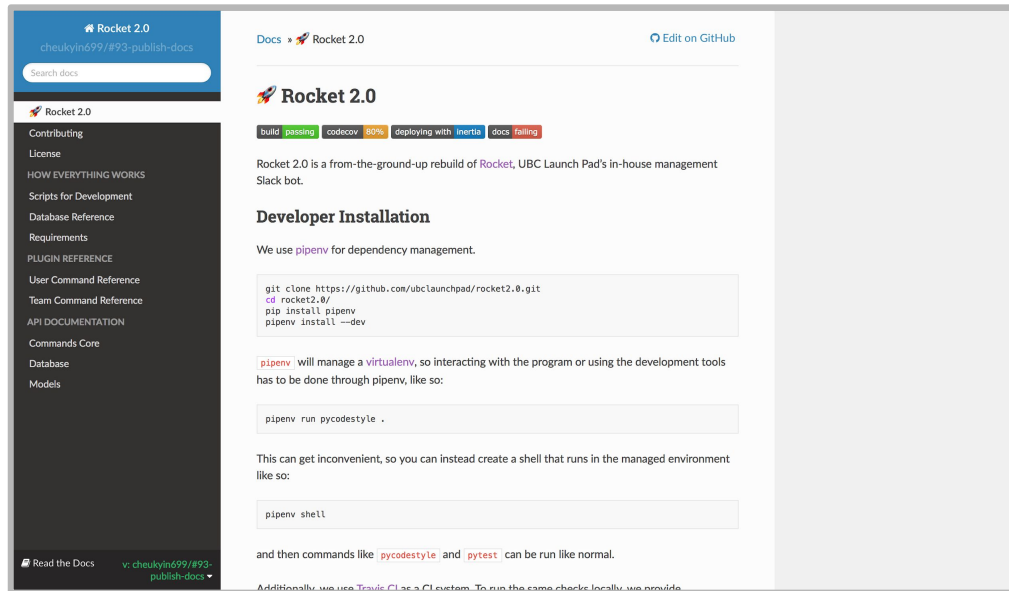
But at least we have this cool
status page?



The Good News

We do have docs!

They're very pretty.



The screenshot displays the Rocket 2.0 documentation website. The header includes the title 'Rocket 2.0' and a link to 'cheukyin699/#93-publish-docs'. A search bar is present. The left sidebar lists various sections: Contributing, License, HOW EVERYTHING WORKS, Scripts for Development, Database Reference, Requirements, PLUGIN REFERENCE, User Command Reference, Team Command Reference, API DOCUMENTATION, Commands Core, Database, and Models. The main content area shows the 'Rocket 2.0' title with a status bar indicating 'build passing', 'coverage 80%', 'deploying with inertia', and 'docs failing'. The text describes Rocket 2.0 as a rebuild of Rocket, UBC Launch Pad's in-house management Slack bot. Under the 'Developer Installation' section, it states that pipenv is used for dependency management. A code block shows the commands to clone the repository, enter the rocket2.0 directory, and install pipenv. Another code block shows the command to run pycodestyle. The text explains that pipenv will manage a virtualenv and that commands like pycodestyle and pytest can be run normally. At the bottom, it mentions using Travis CI for CI system.

Rocket 2.0
cheukyin699/#93-publish-docs

Search docs

Rocket 2.0

Contributing
License
HOW EVERYTHING WORKS
Scripts for Development
Database Reference
Requirements
PLUGIN REFERENCE
User Command Reference
Team Command Reference
API DOCUMENTATION
Commands Core
Database
Models

Read the Docs v: cheukyin699/#93-publish-docs

Docs » Rocket 2.0 Edit on GitHub

Rocket 2.0

build passing coverage 80% deploying with inertia docs failing

Rocket 2.0 is a from-the-ground-up rebuild of **Rocket**, UBC Launch Pad's in-house management Slack bot.

Developer Installation

We use **pipenv** for dependency management.

```
git clone https://github.com/ubclaunchpad/rocket2.0.git
cd rocket2.0/
pip install pipenv
pipenv install --dev
```

pipenv will manage a **virtualenv**, so interacting with the program or using the development tools has to be done through pipenv, like so:

```
pipenv run pycodestyle .
```

This can get inconvenient, so you can instead create a shell that runs in the managed environment like so:

```
pipenv shell
```

and then commands like **pycodestyle** and **pytest** can be run like normal.

Additionally, we use **Travis CI** as a CI system. To run the same checks locally, we provide



Architecture

