TESTING POLICY

Follow Me Drones

Members

Len Bekker
Devon Petrie
Gilad Tabul
Brendon van Biljoen
Francois Venter

5 Guys 1 Branch
5guys1branch@gmail.com

TESTING

Mocha & Chai (Server Unit Testing)

- Mocha is used as a testing framework for our NodeJS server along with Chai.
- Chai is our assertion library of choice since it incorporates an HTTP module, which is useful for testing our API endpoints.
- Chai was chosen over the default NodeJS assertion library due to the vast number of added capabilities.
- Mocha & Chai are used to ensure our routes are set up correctly on the server by testing for expected response codes and expected response bodies.
- These two technologies make feature isolation much more intuitive and we are able to better test individual elements of our code.
- We also test the ability to read and write data through our socket connection to ensure we do not lose connectivity to our server.

Jasmine & Karma (Application Unit Testing)

- Jasmine is used as a testing framework for the creation of tests.
- Karma is used as a task runner for our created tests.
- There exists a configuration file for Karma which is used to set up the reporters, the browser, the testing framework and the startup file.
- The current browser used for testing is Mozilla Firefox.
- For most of the tests, we are checking two things:
  - Each component contains the expected properties
  - Each component is rendered properly by the DOM

Travis CI (Automated Integration Testing)

- Travis CI runs our unit testing files (as described above) every time a file is committed to our development or master branch, and when a pull request is initiated from either of the two branches.
- Every time a team member begins implementing new feature, they branch off development and merge back into development once the feature is completed.
- Our intention with this logic is to not allow faulty code to be committed to our development or master branches.
- This creates a situation where our master branch always has a stable and bug-free version of our code and the development branch is used for integration testing with the rest of the system's features.
- Merges are not finalised until every single unit test completes successfully.
- Test reports are not generated locally but are rather saved by Travis CI on the website. Each build report is accessible via the Travis CI "build history" section through our repository's link on GitHub.