# Testing Policy - Docks

### **TripleParity**

### Contents

1	Testing Process	1
	1.1 Peer Review	1
	1.2 Automated Testing and Continuous Integration	
	1.2.1 Travis CI	
	1.2.2 Docker Cloud	

## 1 Testing Process

#### 1.1 Peer Review

Before code can be merged into the develop branch a Pull Request has to be created. Three reviews are required from other developers before the pull request can be merged into develop.

During peer reviews the following should be checked:

- Architectural problems will this cause problems in the future?
- Compliance with requirements and design is that what we need?
- Coding Standards proper code formatting and security standards?



Figure 1: At least 3 peer reviews are needed for a feature to be merged



Figure 2: An example of a positive peer review

#### 1.2 Automated Testing and Continuous Integration

When a commit is made to the 'docks-ui' repository on GitHub two processes are started:

- Travis CI Builds the repository
- Docker Hub builds the repository and creates a Docker Image that can be deployed in production

#### 1.2.1 Travis CI

The history of test reports for Travis CI can be viewed at https://travis-ci.org/TripleParity/docks-ui/branches. If the build was not successful it will be marked as 'failed'

Travis CI is used for running tests associated with each repository.

For the frontend Angular generates a set of tests for each component to verify that the component was successfully created. These tests run inside a headless (no screen required) Chrome browser.

The backend also has a suite of unit tests that are executed on Travis CI.

#### **Default Branch** -⊳- ff703e7 ⊘ master # 48 passed Evert Geldenhuys 1 build 27 days ago **Active Branches** oo networks2 # 263 started ->- 10affa1 ∅ 27 -CDuPlooy ⊕ 3 builds X loginPage # 262 failed -⊳- 95d81d1 Ø Paul Wood 🟗 2 builds 19 minutes ago -0- 8119ea9 ⊘ ✓ develop # 261 passed 🟗 19 builds GitHub 29 minutes ago

Figure 3: Screenshot of Travis CI branch build history

```
The command "ng build --prod --no-progress" exited with 0
2774 $ ng test --no-progress
2775 (node:5873) ExperimentalWarning: The fs.promises API is experimental
      10 05 2018 20:06:46.174:INFO [launcher]: Launching browser ChromeHeadless with unlimited c
      10\ 05\ 2018\ 20:06:46.179:INFO\ [launcher]:\ \textbf{Starting browser ChromeHeadless}
                                                                0.0.0)]: Connected on socket vos
2780 HeadlessChrome 0.0.0 (Linux 0.0.0): Executed 0 of 18 SUCCESS (0 secs / 0 secs)
2781 e 0.0.0 (Linux 0.0.0): Executed 1 of 18 SUCCESS (0 secs / 0.036 secs)
2782 e 0.0.0 (Linux 0.0.0): Executed 2 of 18 SUCCESS (0 secs / 0.046 secs)
  83 e 0.0.0 (Linux 0.0.0): Executed 3 of 18 SUCCESS (0 secs / 0.057 secs)
2784 e 0.0.0 (Linux 0.0.0): Executed 4 of 18 SUCCESS (0 secs / 0.066 secs)
2785 e 0.0.0 (Linux 0.0.0): Executed 5 of 18 SUCCESS (0 secs / 0.076 secs)
2786 e 0.0.0 (Linux 0.0.0): Executed 6 of 18 SUCCESS (0 secs / 0.086 secs)
2787 e 0.0.0 (Linux 0.0.0): Executed 7 of 18 SUCCESS (0 secs / 0.165 secs)
  88 e 0.0.0 (Linux 0.0.0): Executed 8 of 18 SUCCESS (0 secs / 0.225 secs)
2789 e 0.0.0 (Linux 0.0.0): Executed 9 of 18 SUCCESS (0 secs / 0.248 secs)
2790 e 0.0.0 (Linux 0.0.0): Executed 10 of 18 SUCCESS (0 secs / 0.296 secs)
2791 e 0.0.0 (Linux 0.0.0): Executed 11 of 18 SUCCESS (0 secs / 0.307 secs)
    e 0.0.0 (Linux 0.0.0): Executed 13 of 18 SUCCESS (0 secs / 0.353 secs)
2794 e 0.0.0 (Linux 0.0.0): Executed 14 of 18 SUCCESS (0 secs / 0.372 secs)
2795 e 0.0.0 (Linux 0.0.0): Executed 15 of 18 SUCCESS (0 secs / 0.38 secs)
2796 e 0.0.0 (Linux 0.0.0): Executed 16 of 18 SUCCESS (0 secs / 0.406 secs)
     e 0.0.0 (Linux 0.0.0): Executed 18 of 18
                                                SUCCESS (0 secs / 0.525 secs)
2799 e 0.0.0 (Linux 0.0.0): Executed 18 of 18 SUCCESS (0.565 secs / 0.525 secs)
     The command "ng test --no-progress" exited with \theta.
```

Figure 4: 18 UI tests running on Travis

#### 1.2.2 Docker Cloud

Docker images are build on Docker Cloud. These image can then be deployed in a production environment or for development and testing. Images can be viewed at https://hub.docker.com/u/tripleparity/

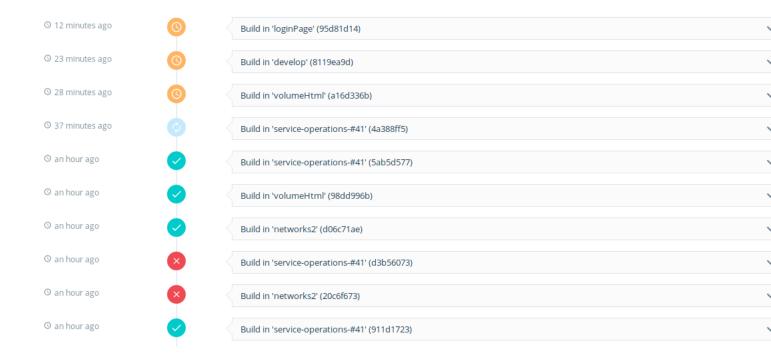


Figure 5: Screenshot of Docker Cloud Build History