
Testing Policy Document

Testing Policy Document

Ctrl Alt Defeat

Contents

1	Quality Requirements Testing	2
1.1	Usability	2
1.2	Security	2
1.3	Performance	2
1.4	Scalability	2
1.5	Modifiability	2
2	Code Coverage	3
3	Choice of testing tools/frameworks	4
3.1	Frontend Testing	4

1 Quality Requirements Testing

1.1 Usability

1.2 Security

1.3 Performance

1.4 Scalability

1.5 Modifiability

2 Code Coverage

Any commit made to a branch causes automated tests to be run on the codebase of that branch, thereafter the code coverage of said branch is calculated. Any branch being merged via pull request into the development branch (dev) needs to have the coverage of changes to the codebase to match or better the coverage of the development branch. Furthermore, the coverage of the newly committed code (ie: patch) must match or exceed the coverage percentage of the project. This ensures that the coverage of the codebase is never decreasing, and that sufficient testing is being done on the codebase. Furthermore any time the development branch is merged into the master branch, the coverage of the development branch must match or be higher than that of the master branch, ensuring an increasing coverage and sufficient testing.

3 Choice of testing tools/frameworks

3.1 Frontend Testing

For our frontend testing frameworks and tools we decided to use the following:

- **Karma and Jasmine** - Jasmine is the testing framework that is used to write actual tests and are typed in Javascript, Karma is the test runner that executes the tests. Karma is run from a CLI(Command Line Interface) and it will open up a browser window and run the tests in that browser. Karma will then report the results of the tests back to the CLI and can be used to generate a coverage report. Karma and Jasmine are recommended by Angular which is what our frontend is primarily built upon and they are the most popular testing frameworks for Angular applications. The advantages of using Karma and Jasmine over other testing frameworks is that they are easy to set up and use, they are well documented and they are popular amongst Angular developers which means there is a extensive amount of resources available online for help if need be.
- **Cypress** - Cypress is a testing framework that is used to write end to end tests which are tests that try and simulate a user using the application. End to end tests are needed to ensure that the application is working exactly as expected from the user interface level all the way through the application to the database level and checks all the integration between these componets work as expected. Cypress runs in a browser which makes it easy to setup and follow the tests as they excute in the browser. Cypress is documented well with a thriving community which allows for easy access to information if any problems arise.

For our backend testing framework and tools we decided to use the following:

- **Django built in testing module** - Django has a built in testing module which allows for the testing of Django applications, this is yet another reason why we decided to use django as it has amazing functionality out of the box. The django testing module allows for extensive testing of the application. The advantages of using django testing framework over an external framework is that one is already used to the syntax of django since our backend is primarily built on django and hence saves valuable time trying to learn syntax of another backend testing framework. Django allows for fast pace development which is much needed in certain situations such as in ours when following an agile development strategy. Django testing framework is also well documented and has a large community which allows for easy access to information if any problems arise.