

1.

Hello!

Today i'm going to tell you about how we are going to test our solution [semantify.io](https://semantify.io). Our mentor is Irlan, our team is Aleksandr, Shinho, Omar and me, Alexey.

2.

There are three main areas of testing that we have.

Unit tests, end-to-end tests and ensure that our code could be maintain painlessly. Let's talk about all this areas.

3.

When people talk about testing software than they usually mean unit tests. This is the good way to be sure that some code is not broken during development of other components. It is also the key principle of test-driven-development.

In our project we use mocha.js wrapped by test runner karma. In next slides I will also say some words about karma. For asserts we use chai.js. It's very flexible assets framework. It has an impressive number of plugins for all main cases. It supports, angular, react (that we use), ecmascript 2015, it has spies or integration with sinon.

4.

Now, let's move to end-to-end tests. End-to-end testing is a methodology used to test whether the flow of an application is performing as designed from start to finish.

The purpose of carrying out end-to-end tests is to identify system dependencies and to ensure that the right information is passed between various system components.

In other words, we are going to test some typical scenarios. For example, login, upload data, edit data. Or just checking the list of existing elements.

We will use nightwatch.js for end-to-end tests. It is a testing framework with clear syntax and great possibilities.

Nightwatch is based on selenium web driver.

5.

Important thing to mention is code quality.

One of criteria is easy-to-maintain code. Thus, we use in our javascript part ES LINT. It keeps code consistent, explicitly forcing programmers to follow a code style defined in a project. For example, use triple equality to compare values, keeping indents, etc.

Another part of code quality is well-commented code. Comments are necessary for custom methods, for complex or non-default behaviour and to describe concept of each single component.

So, ESLint and comments must be present in code in order to make the project maintainable.

6.

So, now some more information about karma.

We assume that in the future one could add continuous integration in the project.

And we give all necessary tools: karma and end to end tests.

In the nearest 1 or 3 months headless Chrome will be released. It comes as better substitution to Phantomjs. So, if one will decide to add continuous integration it will be even easier than to make it today.

7.

We are going to test the most important parts in our project.

For backend it is endpoints that are used to pass data.

For frontend it's redux reducers, small functions that transform data according to coming actions.

8.

Thank you for your attention!