

JS Unit Testing

Team: El Pimpi
Mihail Gaberov



Lottoland

dreamIT 

Motivation

- To have all team members acknowledged in the field and be able to write unit test for our Javascript based projects.
- This could be continued as series of internal workshops in order to keep improving ourselves in all front end skills that we need or might need.

What is a JS Unit test?

- A pure function that you can deal with – a function that always gives you the same result for a given input. This makes unit testing pretty easy, but most of the time you need to deal with side effects (DOM manipulations, API calls, etc). It's still useful to figure out which units we can structure our code into and to build unit tests accordingly.
- 5 Questions Every Unit Test Must Answer
- Example:

```
it('Description of the test', () => {  
  // Implementation of the test  
  expect(true).to.be(true)  
})
```

Why do we need to write unit tests?

- Make us to refactor the code, in order to make it testable. This means to make it scalable and modularised and easy to reason about it. Helps automation and avoiding manual testing.
- Slow us in the beginning, speed up after.
- A must for almost every project - no tests only for very, really very small ones.
- Help us reasoning about the quality of our code.
- Change the way of thinking when programming.

How do we write JS unit tests?

- Testing libraries - [mocha](#), [jasmine](#)
- Assertion libraries - [chai](#), [expect.js](#), [should.js](#)
- Async testing - [sinonjs](#) (next time more on this)

Examples

<https://github.com/mihailgaberoov/javascript-testing>

References:

- <http://howtodoinjava.com/scripting/javascript/jasmine-javascript-unit-testing-tutorial/>
- <https://classroom.udacity.com/courses/ud549/lessons/3769099021/concepts/38687287730923>
- <https://www.smashingmagazine.com/2012/06/introduction-to-javascript-unit-testing/>
- <https://www.sitepoint.com/javascript-testing-tool-showdown-sinon-js-vs-testdouble-js>