

Unit test JS

March 2020

1 Introduction

This unit test document applies to the Web sharing site sub team of Team Gamma. For the purposes of performing unit tests for the web development of the website we shall be making use of the Jasmine unit test framework which can test Javascript code as well a few others but for the purposes of this document only js code will be shown. The steps below describe an example of how you can create, write and run them.

NOTE: Due to the formatting of this document, directly copy pasting the code may result in some characters(for example: ',') not being correctly interpreted by your IDE or text/code editor please retype them or copy them from the references and jasmine files included in the link below if need be.

2 Creating a unit test file

Jasmine runs unit tests in the browser so in order to create a basic unit test as an example you will require the following files:

- 1.) An empty HTML file
- 2.) A empty test.js file
- 3.) The .js file which contains functions you would like to test.
- 4.) The jasmine package: jasmine.css, jasmine.js, jasmine-html.js and boot.js which can be found in the /lib directory when downloaded from here:
<https://github.com/jasmine/jasmine/releases/download/v3.5.0/jasmine-standalone-3.5.0.zip>

In the newly created HTML file add the following code (edit with source relative to where you have stored the above jasmine package) in the head tag, note that the ordering of the script tags and files is important :

```
<link rel="stylesheet" href="jasmine.css">
<script src="jasmine.js"></script>
<script src="jasmine-html.js"></script>
<script src="boot.js"></script>
```

After that add the script source of the js file that has the functions to be tested and then the name of the test file last. for example if main.js which contains the function code and test.js which will be where the test functions will be entered is in the same directory as the HTML file then add:

```
<script src="main.js"></script>
<script src="test.js"></script>
```

3 Writing the test file

To write to the test file open the test.js file. Below is an example of a test file function which tests if the helloWorld() function located in the main.js is working correctly as expected by evaluating its output.

```
describe('Hello world', () => { (1)
  it('says hello', () => { (2)
    expect(helloWorld()) (3)
      .toEqual('Hello world!'); (4)
  });
});
```

In line (1) above the describe(string, function) function defines what we call a Test Suite, a collection of individual Test Specs where by the first parameter gives the name or description of the function being tested and the second argument defines a function that will run.

In line (2) The it(string, function) function defines an individual Test Spec, this contains one or more Test Expectations.

In line (3) The expect(actual) expression is what we call an Expectation. In conjunction with a Matcher(described below) it describes an expected piece of behaviour in the application.

In line (4) The matcher(expected) expression is what we call a Matcher. It does a boolean comparison with the expected value passed in vs. the actual value passed to the expect function, if they are false the spec fails.

With regards to the matcher jasmine comes pre-built in with a few matchers for example:

```
expect(array).toContain(member);
expect(fn).toThrow(string);
expect(fn).toThrowError(string);
expect(instance).toBe(instance);
expect(mixed).toBeDefined();
expect(mixed).toBeFalsy();
expect(mixed).toBeNull();
expect(mixed).toBeTruthy();
expect(mixed).toBeUndefined();
expect(mixed).toEqual(mixed);
expect(mixed).toMatch(pattern);
```

```
expect(number).toBeCloseTo(number, decimalPlaces);  
expect(number).toBeGreaterThan(number);  
expect(number).toBeLessThan(number);  
expect(number).toBeNaN();  
expect(spy).toHaveBeenCalled();  
expect(spy).toHaveBeenCalledTimes(number);  
expect(spy).toHaveBeenCalledWith(...arguments);
```

To run the tests simply open the HTML file we created in the beginning and once it has loaded it will run the tests.

4 References

- 1. http://jasmine.github.io/edge/introduction.html#section-Included_Matchers
- 2. <https://codecraft.tv/courses/angular/unit-testing/jasmine-and-karma/>