

Angular 6 Training Course

Exercise M-testing

- The process of testing **isolated** pieces of code. If the code refers to some external dependency/resource, it is not a unit test.
- Jasmine is a Javascript testing framework.
- It uses a variant of **Test Driven Development (TDD)** called **Behaviour Driven Development (BDD)**.
- It attempts to make tests human readable.
- Taking a function like

```
function double( n ) { return n*2; }
```

- The **Jasmine** test looks like this, in ES5:

```
describe("Double function",
  function() {
    it( "doubles numbers" , function() {
      expect( double(2)).toEqual(4)
    })
  }
)
```

- **describe** is the Test Suite.
- **it** defines a Test Spec
- **expect** defines one assertion.
- toEqual is a **matcher**, which compares its value 4 to the result returned from double(2)
- If the two results do not match, the test fails.
- This is a list of Javascript-specific Jasmine matchers:

```
https://jasmine.github.io/edge/introduction.html#section-Included\_Matchers
```

Karma

- **Karma** is a test runner, which manages the process of running Jasmine, in a browser or on the command line, and refreshing results as we update our code.

Testing a Pipe

- If I create a PIPE to double numbers like this:

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'double'
})
export class DoublePipe implements PipeTransform {
  transform(value: any, args?: any): any {
    return value * 2;
  }
}
```

- I can create a variable in my component class:

```
sales = 1024;
```

- I can use that variable in a template, with the pipe:

```
<p>{{ sales | double }}</p>
```

- I can write Jasmine tests to test it creates an instance, and test it returns a doubled number:

```
describe('DoublePipe', () => {

  it('create an instance', () => {
    const pipe = new DoublePipe();
    expect(pipe).toBeTruthy();
  });

  it('doubles numbers', () => {
    const pipe = new DoublePipe();
    expect(pipe.transform(2)).toEqual(4);
  });

});
```

- I can refactor this code to reduce repetition, using setup code:

```
describe('DoublePipe', () => {

  let pipe:DoublePipe;

  beforeEach(() => {
```

```

        pipe = new DoublePipe();
    })

    it('create an instance', () => {
        expect(pipe).toBeTruthy();
    });

    it('doubles numbers', () => {
        expect(pipe.transform(2)).toEqual(4);
    });

})

```

Headless testing

- By editing the **Karma configuration file karma.conf.js**, we can set the Jasmine tests to display in the same Node window that we run **ng test** from.
- Add a **headless** configuration object to karma.conf.js

```

customLaunchers: {
  ChromeHeadless: {
    base: 'Chrome',
    flags: [
      '--headless',
      '--disable-gpu',
      // Without a remote debugging port, Google Chrome
      exits immediately.
      '--remote-debugging-port=9222',
    ],
  }
}

```

- Change this line to use this new object

```

browsers: ['ChromeHeadless'],

```

- Run tests using **ng test**, and the results should appear in the same Terminal window.