Basic Testing in Angular

Index

Introduction to testing

- Testing can help reveal bugs before they appear, instll confidence in yout web application, and makes it easy to onboard new developer into the application
- With testable code we mean code that is broken into several method and functionality being part of a bigger picture

End-to-end vs Unit Testing

- There are two major ways to test your applications: end-to-end testing or unit testing

Testing tools

- Jasmine: is a behavior-driven development framework for testing JavaScript code. Using Jasmine you can set expectations about what your code should do when invoked
- Karma: with jasmine we can describe our tests and their expectations. Karma allows us to run JavaScript code within a browser like Chrome or Firefox, or on a headless browser like PhantomJS.

Writing Unit Tests in Angular

- Main testing framework can be found on the @angular/core/testing package
- Before start testing our code, it's necessary to setup Karma and the project to be tested
- As almost every project in Angular, we can configure our file using Angular CLI, in this case creating the karma.conf.js file

Testing Services

- Before start testing services mock (or stub) all of your dependency, e.g. not hit the real Spotify server
- Stubs are objects we create on the fly, with a subset of the behavior our dependency has
- Mocks are a more complete representation of objects, that overrides parts or all of the behavior of the dependency. Mocks can, and most of the time will be reused by more than one test across our suite.
- the biggest difference between a mock and a stub is that:
    - a stub provides a subset of functionality with "manual" behavior overrides whereas
    - a mock generally sets expectations and verifies that certain methods were

Testing Routing to Components

- When testing components, we can either:
  - write tests that will interact with the component from the outside, passing attributes in ad checking how the markup is affected (e.g. Black box testing)
  - test individual component methods and their output (e.g. White box testing)

Mocking dependencies

- To mock a dependency create a file named: nameOfTheService.service.mock.ts
- A spy is a specific type of mock object that gives us two benefits:
  - we can simulate return values and
  - count how many times the method was called and with which parameters

Testing HTTP requests

- We can test HTTP interaction mocking version of the HttpClient or HttpClient class, since it is an external dependency
- Angular testing library already provides a built in alternative: HttpTestingController

Testing a POST and other requests

- When writing our test for this method, our goal is to test two things:
  - the request method (POST) is correct and that
  - the URL we're hitting is also correct
- then you can also test DELETE method, GET method and the header of the request

Summary

Having test inside Angular simplifies all of the work to be done. It's easy to test: controllers, services, forms and HTTP requests.