angular testing

unit testing your angular app

The importance of testing

- Angular is designed with testability in mind
- the all architecture of creating components is really testable
- with unit testing we achieve the following
    - supply a more reliable product
    - improve the build and delivery process of our app
    - communicate better how to use our components, directives, pipes and services
    - test edge scenarios
    - run previous tests and make sure we didn’t break anything
- unit test should not be affected from previous tests

Testing Technologies

When we start a new project using **@angular/cli** we already have unit testing technologies installed

- Jasmine
- Karma
- Protractor

- to create a unit test we need to create a file with **.spec.ts** extension
- we will put this test file in component, directive, pipe, service folder

Our first test file

- let’s write our first test which will verify that true is equal true
- in **Jasmine** we place a group of tests in **describe** block
- every test we place in an **it** function
- we can run the tests by running: **npm test**

Debugging our tests

- debugging the tests is similar to how we debug our application
- we use the developer tools to debug our test
- source maps will be created so we can place breakpoints in our source code

testing component

- the previous lesson we practiced routing by creating todo app with search and task details screen
- we will practice testing on the app we created
- we will write a simple test for the about page
- the about page contains a header we will run a test that we have an header with certain text

Testing Module

- to unit test a component we need to create a testing module for that component
- the testing module contain the minimal imports, declarations and providers needed to test our component
- we create the testing module with **TestBed.configureTestingModule**
- **TestBed** is in charge of creating the testing module
- we configure the testing module every time from scratch before each test
- we can run code before each test in **Jasmine** by adding **beforeEach**

createComponent

- we also use **TestBed** to create a **ComponentFixture** of the component that we want to test
- **ComponentFixture** is a wrapper around our component which adds additional testing tools on our component
- **beforeEach** test we need to create a component fixture
- let's use **TestBed.configureTestingModule** and **createComponent** to create our first test
- we also need to run **detectChanges** on the component otherwise it will get unrendered

Testing user input

- a test can also fill the fields in a form and create events on elements of the component
- from the previous lesson we created a search box and changing the search box should call our input change event in the component class
- let’s create a test that verify that the function is called
-

test ngmodel

- let’s add NgModel to our search component and check that it updates a property
- it’s recommended to wrap functions in **async** this will make sure that the test will complete when all async tasks are done

mock server response

- it’s recommended not to query the server from unit testing
- we can mock server response
- let’s test the detail screen and mock the server response and verify that we are getting the right text in the element
- it’s recommended to mock angular services and not your services