## Unit Testing - Overview

Just like it’s predecessor Angular 1, Angular has been designed with testability as a primary goal.

When we talking about testing in Angular we are usually talking about two different types of testing:

**Unit Testing**

This is sometimes also called *Isolated* testing. It’s the practice of testing small isolated pieces of code. If your test uses some external resource, like the network or a database, it’s *not* a unit test.

**Functional Testing**

This is defined as the testing of the complete functionality of an application. In practice with web apps, this means interacting with your application as it’s running in a browser just like a user would interact with it in real life, i.e. via clicks on a page.

This is also called ***End To End*** or ***E2E*** testing.

In this section we are only going to cover how to perform *unit testing* with Angular.

You will learn:

* How to write and run tests in Angular using Jasmine & Karma.
* How to write tests for classes, components, directives and pipes.
* How to use the Angular Test Bed to help test behaviours that depend on the Angular framework.
* How to write tests using mocks and spys.
* How to write tests that depend on change detection.
* How to write tests that involve an asynchronous action.
* How to write tests that require the dependency injection framework.
* How to write tests that involve the Http service.
* How to write tests that involve model driven forms (template driven forms require E2E testing).
* How to write tests to test the Router.