**Calculator https://dzone.com/articles/unit-testing-in-angular-4-using-jasmine-amp-karma**

**KARMA**

Karma used for unit testing and runs tests automatically when changes are made to production.

**CLASSES AND PIPES**

**1 Testing Classes**

\*we are testing classes mostly

To test a specif componenent it needs to sit next to the component and be imported.e.g

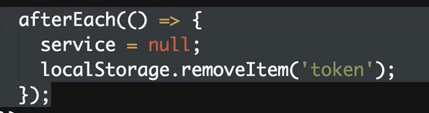
import {AuthService} from './auth.service';// you literally import the file to be tested hear

you always want to test a fresh instances and clean instances of a component so you create before and after each

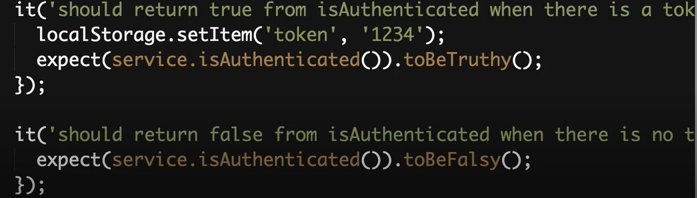
after the describe function. U create a variable and then new instance.e.g



After the test you want to null out and clear data e.g.



Don’t forget to create the test data as well e.g

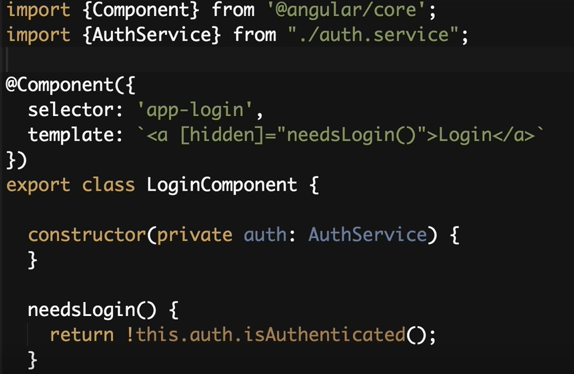


**2 Testing pipes.**

Pipes have a function called transform so basically all you do is put inputs and expect outputs.

**MOCKS AND SPIES**

Used to test classes that have dependencies on other classes or so. E.g



Remember you are testing classes. Look up at export class logincomponent.

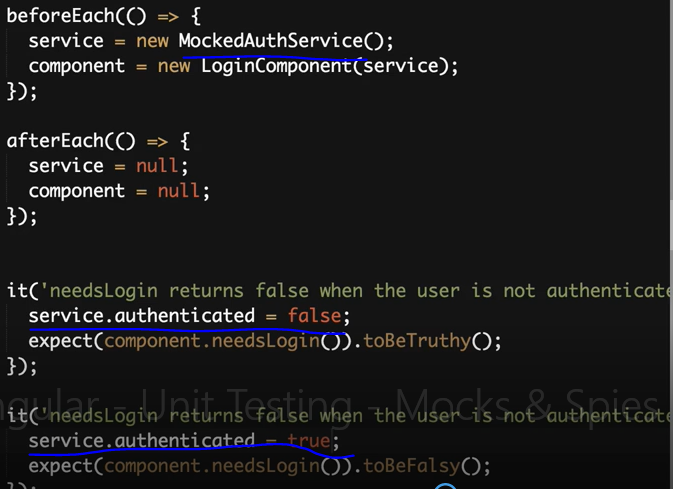
Inject auth service by creating private constructor.

What you import is dependent on what the class is depending on.

Mocking is an acting dependency. Its used to act like a dependency you import but this time you allow it to return what you want it to return. That way you can truly test something isolated but what if things hang …? Research.

With mocks your tests will always pass.





The spy function is a better mock it literally spys on the function but you do not have to worry about the actual functionality you just tell it what to return.



The to have been called function is used to make sure that the function was actually called.

**ANGULAR TESTBED**

Higher level of only testing angular frameworks.

The TestBed.configureTestingModule() method takes a metadata object that can have most of the properties of an [@NgModule](https://angular.io/guide/ngmodules).

To test a service, you set the providers metadata property with an array of the services that you'll test or mock.

**NgModules** configure the injector and the compiler and help organize related things together.

Angular modularity

Modules are a great way to organize an application and extend it with capabilities from external libraries.

Angular libraries are NgModules, such as [FormsModule](https://angular.io/api/forms/FormsModule), [HttpClientModule](https://angular.io/api/common/http/HttpClientModule), and [RouterModule](https://angular.io/api/router/RouterModule). Many third-party libraries are available as NgModules such as [Material Design](https://material.angular.io/), [Ionic](http://ionicframework.com/), and [AngularFire2](https://github.com/angular/angularfire2).

NgModules consolidate components, directives, and pipes into cohesive blocks of functionality, each focused on a feature area, application business domain, workflow, or common collection of utilities.

Modules can also add services to the application. Such services might be internally developed, like something you'd develop yourself or come from outside sources, such as the Angular router and HTTP client.

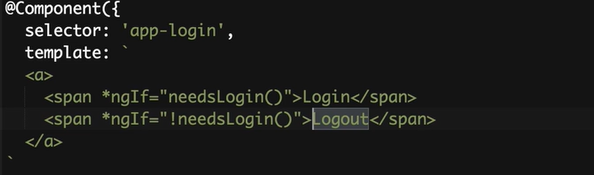
Modules can be loaded eagerly when the application starts or lazy loaded asynchronously by the router.

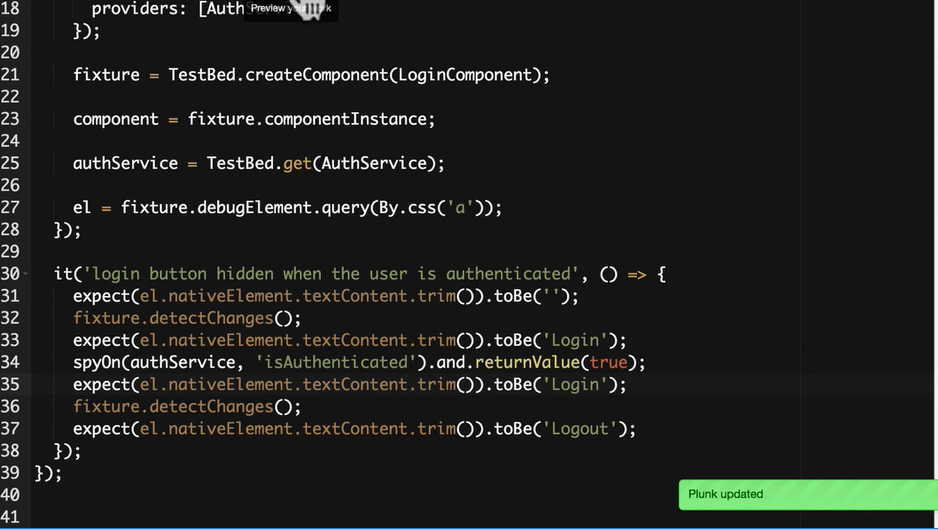
NgModule metadata does the following:

* Declares which components, directives, and pipes belong to the module.
* Makes some of those components, directives, and pipes public so that other module's component templates can use them.
* Imports other modules with the components, directives, and pipes that components in the current module need.
* Provides services that the other application components can use.

**DETECT CHANGES**

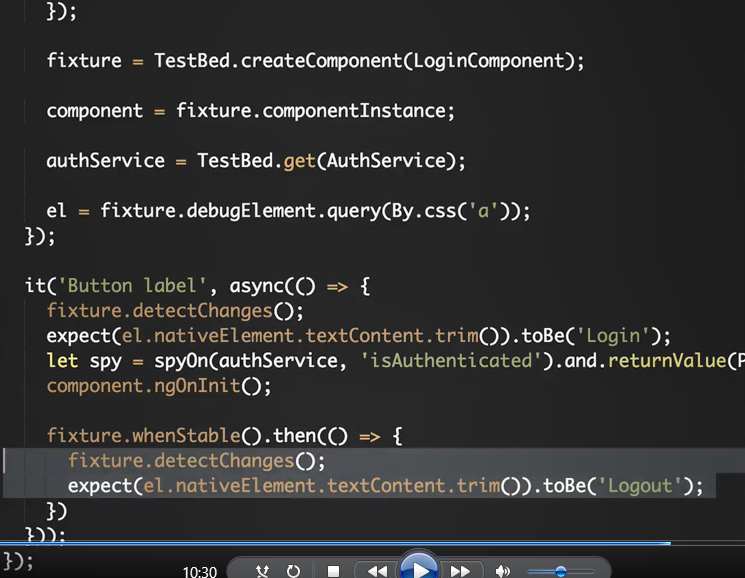
Fixture detect changes used to make changes and trigger something else.





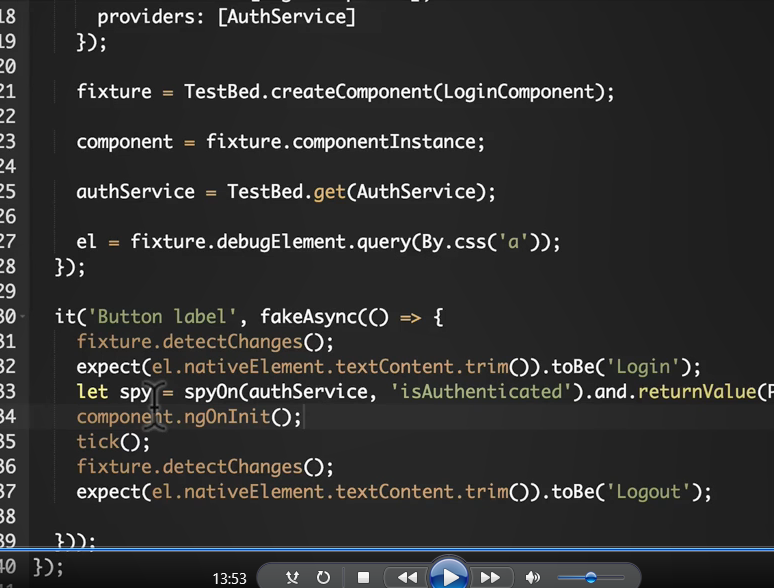
**TESTING ASYNCHRONUS FUNCTIONS**.

Adding async for each spec.



Fake async

Doesn’t make http requests etc this is used for pure isolated testing



**Testing components**

