



Unit testing



End-to-end testing

- **Jasmine:** Jasmine is a behavior-driven development framework in order to test JavaScript code, you can find more information on Jasmine at: <https://jasmine.github.io>
- **Karma:** Karma is a test runner we use for running our unit test while developing; you can find more information on Karma at: <http://karma-runner.github.io>
- **Protractor:** Protractor is an end-to-end testing framework for Angular applications. You can find more information on Protractor at: <http://protractortest.org>



[intro](#) [config](#) [plus](#) [dev](#) [about](#) [v1.0](#)

**//** *On the [AngularJS](#) team, we rely on testing and we always seek better tools to make our life easier. That's why we created Karma - a test runner that fits all our needs.*

 [View project on GitHub](#)

 `npm install karma`



Fork me on GitHub



[GET STARTED](#)

[DOCS](#)

[SUPPORT](#)

[RELEASES](#)

[GITHUB](#)

#### FAST

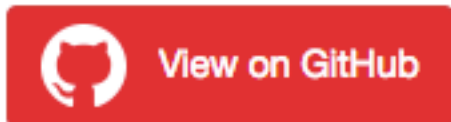
Low overhead, no external dependencies.

#### BATTERIES INCLUDED

Comes out of the box with everything you need to test your code.

#### NODE AND BROWSER

Run your browser tests and Node.js tests with the same framework.



Protractor is an end-to-end test framework for Angular and AngularJS applications. Protractor runs tests against your application running in a real browser, interacting with it as a user would.

### Test Like a User

Protractor is built on top of WebDriverJS, which uses native events and browser-specific drivers to interact with your application as a user would.

### For Angular Apps

Protractor supports Angular-specific locator strategies, which allows you to test Angular-specific elements without any setup effort on your part.

### Automatic Waiting

You no longer need to add waits and sleeps to your test. Protractor can automatically execute the next step in your test the moment the webpage finishes pending tasks, so you don't have to worry about waiting for your test and webpage to sync.

### Setup



`describe()` : The describe function is a global Jasmine function. It is used for grouping similar kinds of tests/specs together in a suite. The describe functions can be nested. The syntax is as follows:



`it()` : It is a Jasmine function used for writing the actual unit tests. The syntax is as follows:



**Matchers:** Matchers are the built-in Jasmine functions used along with the **expect()** function to compare the actual value with the expected value. Here are the matcher functions provided by Jasmine:



`toBe()`



`toEqual()`



`toMatch()`



`toBeDefined()`



`toBeUndefined()`



`toBeNull()`



`toBeNaN()`



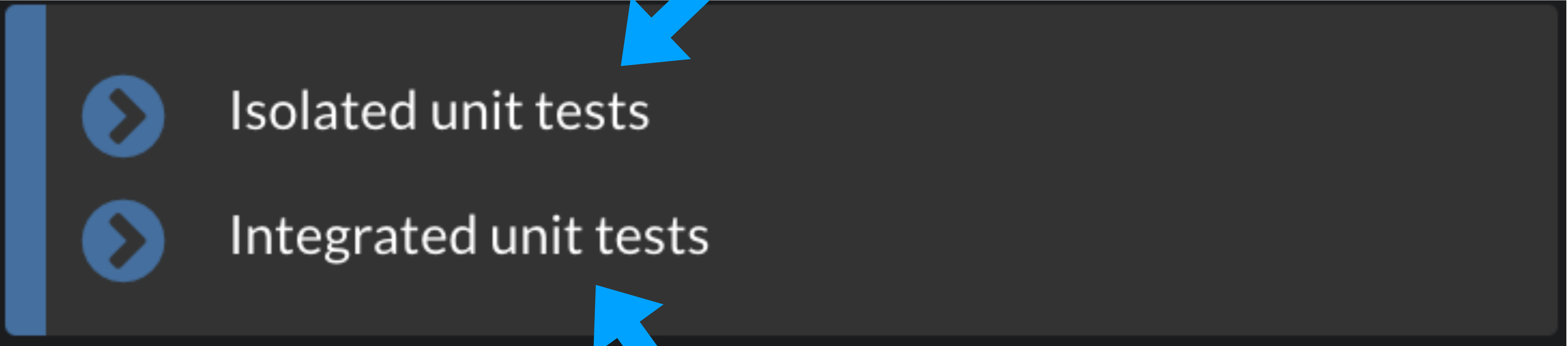
`toBeTruthy()`



- `expect()` : It is an another Jasmine function that takes a value named actual value, and it is used along with matcher functions to assert the expected value.
- `beforeEach()` : `beforeEach()` is a Jasmine built-in function that runs the code inside it before every test in the `describe()` function.
- `afterEach()` : `afterEach()` is a Jasmine built-in function that runs the code inside it after every test in the `describe()` function.
- `beforeAll()` : `beforeAll()` is a Jasmine built-in function that runs the code inside it only once before all the tests in the `describe()` function.
- `afterAll()` : `afterAll()` is a Jasmine built-in function that runs the code inside it only once after

# Test **Unitario**

# SÓLO LÓGICA

- 
- Isolated unit tests
  - Integrated unit tests

# MODULOS, TEMPLATES...

```
npm install jasmine-core jasmine --save-dev  
  npm install karma karma-cli --save-dev  
npm install karma-jasmine karma-chrome-launcher
```

# package.json

```
"karma": "karma start karma.conf.js",  
  "pretest:once": "npm run build",  
  "pretest": "npm run build",  
  "test:once": "npm run karma -- --single-run",  
  "test": "concurrently \"npm run build:watch\" \"npm run  
    karma\""
```

# karma.conf



```
module.exports = function (config) {  
  var appSrcBase = 'src/';  
  var appAssets = '/base/app/';  
  config.set({  
    basePath: '',  
    frameworks: ['jasmine'],  
    plugins: [  
      require('karma-jasmine'),  
      require('karma-chrome-launcher')  
    ],  
    client: {  
      builtPaths: [appSrcBase]  
    },  
    files: [],  
    proxies: {},  
    exclude: [],  
    preprocessors: {},  
    reporters: ['progress'],  
    port: 9876,  
    colors: true,  
    logLevel: config.LOG_INFO,  
    autoWatch: true,  
    browsers: ['Chrome'],  
    singleRun: false,  
    concurrency: Infinity  
  });  
}
```

# **Implementación** del test

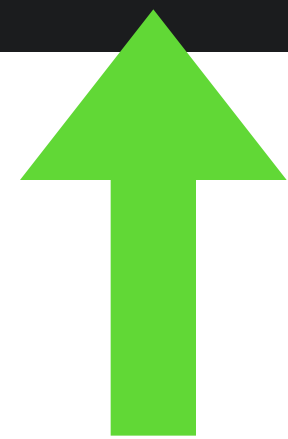
# product-list.component.spec.ts

```
describe('my first unit test', () => {  
  it('true is true', () => expect(true).toBe(true));  
});
```



# npm **run** test:once

```
04 2017 23:58:03.824:INFO [karma]: Karma v1.5.0 server started at http://0.0.0.0:9876/
04 2017 23:58:03.825:INFO [launcher]: Launching browser Chrome with unlimited concurrency
04 2017 23:58:03.836:INFO [launcher]: Starting browser Chrome
04 2017 23:58:04.582:INFO [Chrome 57.0.2987 (Mac OS X 10.12.4)]: Connected on socket kyD5x8aIlwE9XW82AAAA with id 96036093
04 2017 23:58:04.797:WARN [web-server]: 404: /base/src/systemjs.config.extras.js
Chrome 57.0.2987 (Mac OS X 10.12.4): Executed 1 of 1 SUCCESS (0.005 secs / 0.002 secs)
```



**npm run test**

# test

Filipe Silva edited this page 8 hours ago · 10 revisions

## ng test

[▶ Pages](#) 58

### Overview

`ng test` compiles the application into an output directory

- [Angular CLI](#)
- [Generate](#)
- [Stories](#)


### Running unit tests

```
ng test
```

Clone this wiki locally

<https://github.com/angular/angular-cli/wiki/test>



 Clone in Desktop

Tests will execute after a build is executed via [Karma](#), and it will automatically watch your files for changes. You can run tests a single time via `--watch=false` or `--single-run`.

You can run tests with coverage via `--code-coverage`. The coverage report will be in the `coverage/` directory.

### Options