# Starting Protractor

Protractor is an end-to-end test framework for Angular and AngularJS applications.

**End**-to-**end testing** is a methodology used to **test** whether the flow of an application is performing as designed from start to **finish**.

## **Setup**

Protractor is a [Node.js](http://nodejs.org/) program. To run, you will need to have Node.js installed.

By default, Protractor uses the [Jasmine](http://jasmine.github.io/) test framework for its testing interface.

You will need to have the [Java Development Kit (JDK)](http://www.oracle.com/technetwork/java/javase/downloads/index.html)installed to run the standalone Selenium Server.

npm install -g protractor

This will install two command line tools, protractor and webdriver-manager

The webdriver-manager is a helper tool to get an instance of a Selenium Server running. Use it to download the necessary binaries with:

webdriver-manager update

Now start up a server with:

webdriver-manager start

This will start up a Selenium Server. Your Protractor test will send requests to this server to control a local browser. You can see information about the status of the server at <http://localhost:4444/wd/hub>.

**Note: Selenium Server** is a test tool that allows you to write automated web application UI tests in any programming language against any HTTP website using any mainstream JavaScript-enabled browser.

Selenium Server comes in two parts.

1. A server which automatically launches and kills browsers, and acts as a HTTP proxy for web requests from them.
2. Client libraries for your favorite computer language.

# Three Step process

1. Write a Test

Open a new command line or terminal window and create a clean folder for testing as test.

Protractor needs two files to run, a **spec file** and a **configuration file**.

1. **Configuration file**: This File helps protractor to where the test files are placed (specs.js) and to talk with Selenium server. Chrome is the default browser for Protractor.
2. **Spec file:**This File containsthe logic andlocatorsto interact with the application**.**

Copy the following into spec.js:

// spec.js

describe('Protractor Demo App', function() {

it('should have a title', function() {

browser.get(http://localhost:3456');

expect(browser.getTitle()).toEqual('Super Calculator');

});

});

The describe and it syntax is from the Jasmine framework. browser is a global created by Protractor, which is used for browser-level commands such as navigation with browser.get.

Now create the configuration file. Copy the following into conf.js:

// conf.js

exports.config = {

framework: 'jasmine',

seleniumAddress: 'http://localhost:4444/wd/hub',

specs: ['spec.js']

}

This configuration tells Protractor where your test files (specs) are, and where to talk to your Selenium Server (seleniumAddress). It specifies that we will be using Jasmine for the test framework. It will use the defaults for all other configuration. Chrome is the default browser.

Now run the test with

protractor conf.js

You should see a Chrome browser window open up and navigate to the Calculator, then close itself (this should be very fast!). The test output should be 1 tests, 1 assertion, 0 failures.