**Protractor Testing Tutorial: Automation Tool Framework**

<https://www.guru99.com/protractor-testing.html>

Protractor plays an important role in the[Testing](https://www.guru99.com/software-testing.html)of AngularJS applications and works as a Solution integrator combining powerful technologies like Selenium, Jasmine, Web driver, etc. It is intended not only to test AngularJS application but also for writing automated regression tests for normal Web Applications as well.

In this beginner's tutorial, you will learn-

* [Why Do We Need Protractor Framework?](https://www.guru99.com/protractor-testing.html#1)
* [Protractor Installation](https://www.guru99.com/protractor-testing.html#2)
* [Sample AngularJS application testing using Protractor](https://www.guru99.com/protractor-testing.html#3)
* [Execution of the Code](https://www.guru99.com/protractor-testing.html#4)
* [Generate Reports using Jasmine Reporters](https://www.guru99.com/protractor-testing.html#5)

## ****Why can't we find Angular JS web elements using Normal Selenium Web driver?****

Angular JS applications have some extra HTML attributes like ng-repeater, ng-controller, ng-model.., etc. which are not included in Selenium locators. Selenium is not able to identify those web elements using Selenium code. So, Protractor on the top of Selenium can handle and controls those attributes in Web Applications.

The protractor is an end to end testing framework for Angular JS based applications. While most frameworks focus on conducting unit tests for Angular JS applications, Protractor focuses on testing the actual functionality of an application.

Before we start Protractor, we need to install the following:

1. Selenium

You can find the Selenium Installation steps in the following links, (<https://www.guru99.com/installing-selenium-webdriver.html> )

1. NPM (Node.js)

NodeJS Installation, we need to install NodeJS to install Protractor. You can find this installation steps in the following link. ( <https://www.guru99.com/download-install-node-js.html> )

## Running your first Hello world application in Node.js

Once you have downloaded and installed Node.js on your computer, lets try to display "Hello World" in a web browser.

Create file Node.js with file name firstprogram.js

var http = require('http');

http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/html'});

res.end('Hello World!');

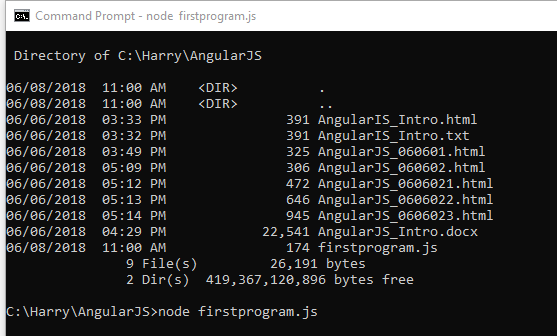
}).listen(8080);

**Code Explanation:**

1. The basic functionality of the "require" function is that it reads a[JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)file, executes the file, and then proceeds to return an object. Using this object, one can then use the various functionalities available in the module called by the require function. So in our case, since we want to use the functionality of http and we are using the require(http) command.
2. In this 2nd line of code, we are creating a server application which is based on a simple function. This function is called, whenever a request is made to our server application.
3. When a request is received, we are asking our function to return a "Hello World" response to the client. The writeHead function is used to send header data to the client and while the end function will close the connection to the client.
4. We are then using the server.listen function to make our server application listen to client requests on port no 7000. You can specify any available port over here.

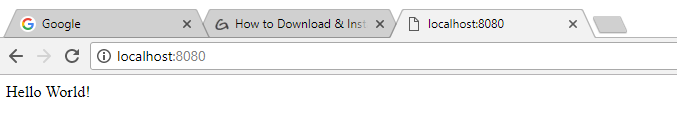
**Executing the code**

1. Save the file on your computer: C:\Harry\AngularJS\ firstprogram.js
2. In the command prompt, navigate to the folder where the file is stored. Enter the command Node firstprogram.js



1. Now, your computer works as a server! If anyone tries to access your computer on port 8080, they will get a "Hello World!" message in return!
2. Start your internet browser, and type in the address: <http://localhost:8080>

Output:



**Summary**

* We have seen the installation of Node.js via the msi installation module which is available on the Node.js website. This installation installs the necessary modules which are required to run a Node.js application on the client.
* Node.js can also be installed via a package manager. The package manager for windows is known as Chocolatey. By running some simple commands in the command prompt, the Chocolatey package manager automatically downloads the necessary files and then installs them on the client machine.
* A simple Node.js application consists of creating a server which listens on a particular port. When a request comes to the server, the client automatically sends a 'Hello World' response to the client.

## Protractor

<https://www.protractortest.org/#/>

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.

Setup

Use npm to install Protractor globally with:

npm install -g protractor

This will install two command line tools, protractor and webdriver-manager. Try running protractor --version to make sure it's working.

The webdriver-manager is a helper tool to easily 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 and will output a bunch of info logs. 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>.

Write a test

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

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

Let's start with a simple test that navigates to the todo list example in the AngularJS website and adds a new todo item to the list.

Copy the following into todo-spec.js:

describe('angularjs homepage todo list', function() {

it('should add a todo', function() {

browser.get('https://angularjs.org');

element(by.model('todoList.todoText')).sendKeys('write first protractor test');

element(by.css('[value="add"]')).click();

var todoList = element.all(by.repeater('todo in todoList.todos'));

expect(todoList.count()).toEqual(3);

expect(todoList.get(2).getText()).toEqual('write first protractor test');

// You wrote your first test, cross it off the list

todoList.get(2).element(by.css('input')).click();

var completedAmount = element.all(by.css('.done-true'));

expect(completedAmount.count()).toEqual(2);

});

});

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.

Configuration

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

exports.config = {

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

specs: ['todo-spec.js']

};

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

Run the test

Now run the test with:

protractor conf.js

You should see a Chrome browser window open up and navigate to the todo list in the AngularJS page, then close itself (this should be very fast!). The test output should be 1 test, 3 assertions, 0 failures. Congratulations, you've run your first Protractor test!