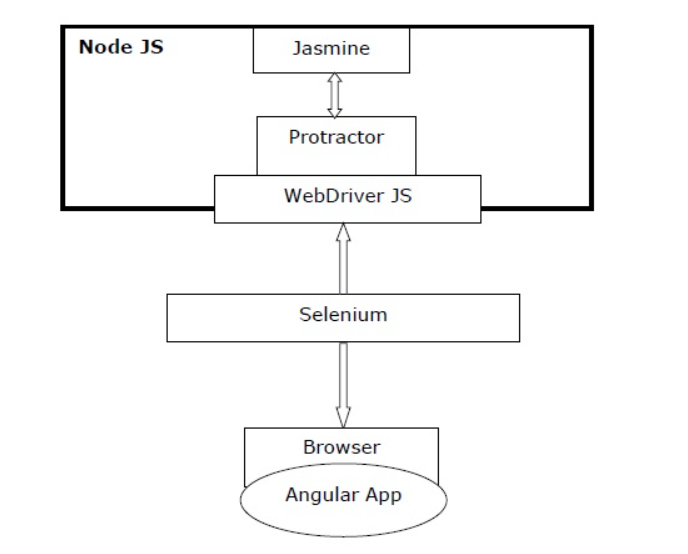
Protractor:

**Protractor** is an automation and end-to-end behavior-driven testing tool that 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. The purpose of Protractor Testing is not only to test AngularJS applications but also for writing automated regression tests for normal Web Applications as well.

Architecture:



* Protractor − it is a wrapper over WebDriver JS especially designed for angular apps.
* Jasmine − It is basically a behavior-driven development framework for testing the JavaScript code. We can write the tests easily with Jasmine.
* WebDriver JS − It is a Node JS bindings implementation for selenium 2.0/WebDriver.
* Selenium − It simply automates the browser.

In order to work with Protractor the following has to be installed:

1. Selenium
2. Node js
3. Npm install -g protractor
4. Protractor –version
5. Webdriver-manager update
6. Webdriver-manager start

Start the web driver manager. This step will run the web driver manager in the background and will listen to any tests which run via protractor.

There should be 2 files created in the Visual studio code for creating protractor testcases.

1. Spec file (spec.js)
2. Conf.js

Protractor examples:

**Example 1:**

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);

});

});

Conf.js:

exports.config = {

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

specs: ['spec.js']

};

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.

To run it:

On the terminal type: protractor conf.js

Example 2:

describe('demo calculator tests', function(){

it('addition test', function(){

browser.get('http://juliemr.github.io/protractor-demo/');

element(by.model('first')).sendKeys('2');

element(by.model('second')).sendKeys('4');

element(by.css('[ng-click="doAddition()"]')).click();

let res = element(by.cssContainingText('.ng-binding', '6'));

expect(res.getText()).toEqual('6');

browser.sleep(2000)

});

});

Add this spec to the conf.js

[Protractor API Cheatsheet (github.com)](https://gist.github.com/javierarques/0c4c817d6c77b0877fda)

[Jasmine cheatsheet (devhints.io)](https://devhints.io/jasmine)

**To take the screen shot of the test case using the Protractor and the Jasmine:**

conf.js

var HtmlScreenshotReporter = require('protractor-jasmine2-screenshot-reporter');

var reporter = new HtmlScreenshotReporter({

dest: 'target/screenshots',

filename: 'my-report.html'

});

exports.config = {

directConnect: true,

capabilities : {

'browserName': 'chrome'

},

framework: 'jasmine',

specs: ['../tests/calaculator.js],

jasmineNodeOpts: {

defaultTimeoutInterval: 30000

},

// Setup the report before any tests start

beforeLaunch: function() {

return new Promise(function(resolve){

reporter.beforeLaunch(resolve);

});

},

// Assign the test reporter to each running instance

onPrepare: function() {

jasmine.getEnv().addReporter(reporter);

},

// Close the report after all tests finish

afterLaunch: function(exitCode) {

return new Promise(function(resolve){

reporter.afterLaunch(resolve.bind(this, exitCode));

});

}

};