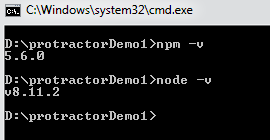
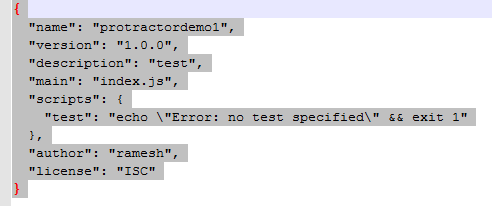
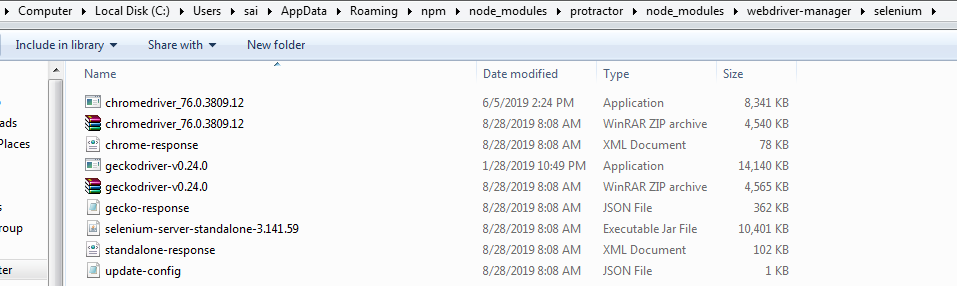
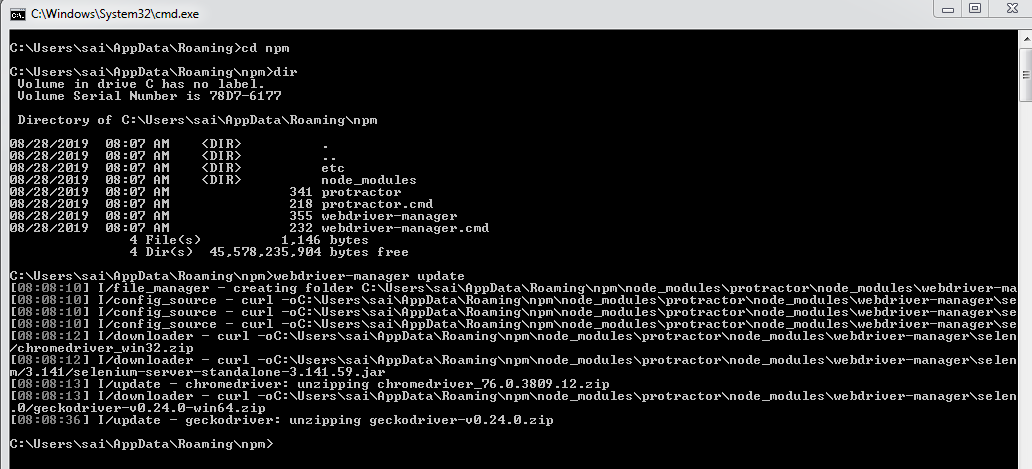
Install Node.js software then verify below versions availability



Npm init

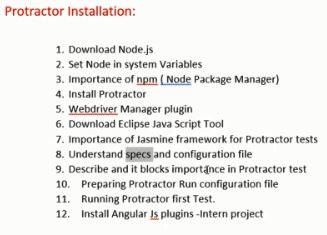


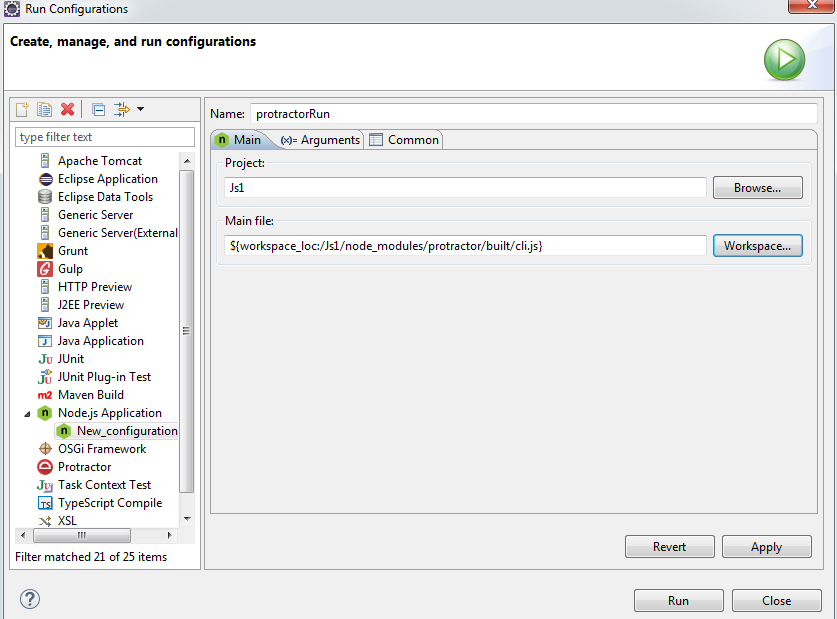
Npm install -g protractor

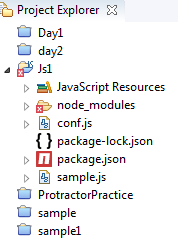


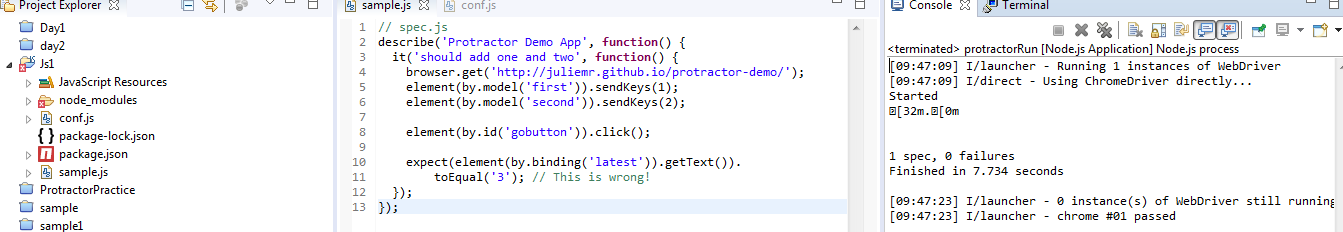
Npm uninstall –g protractor

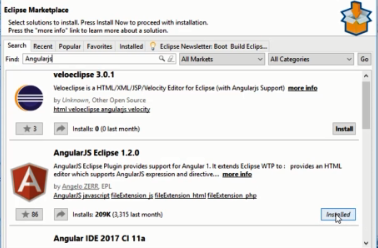
Webstorm , ecipse, visual studio

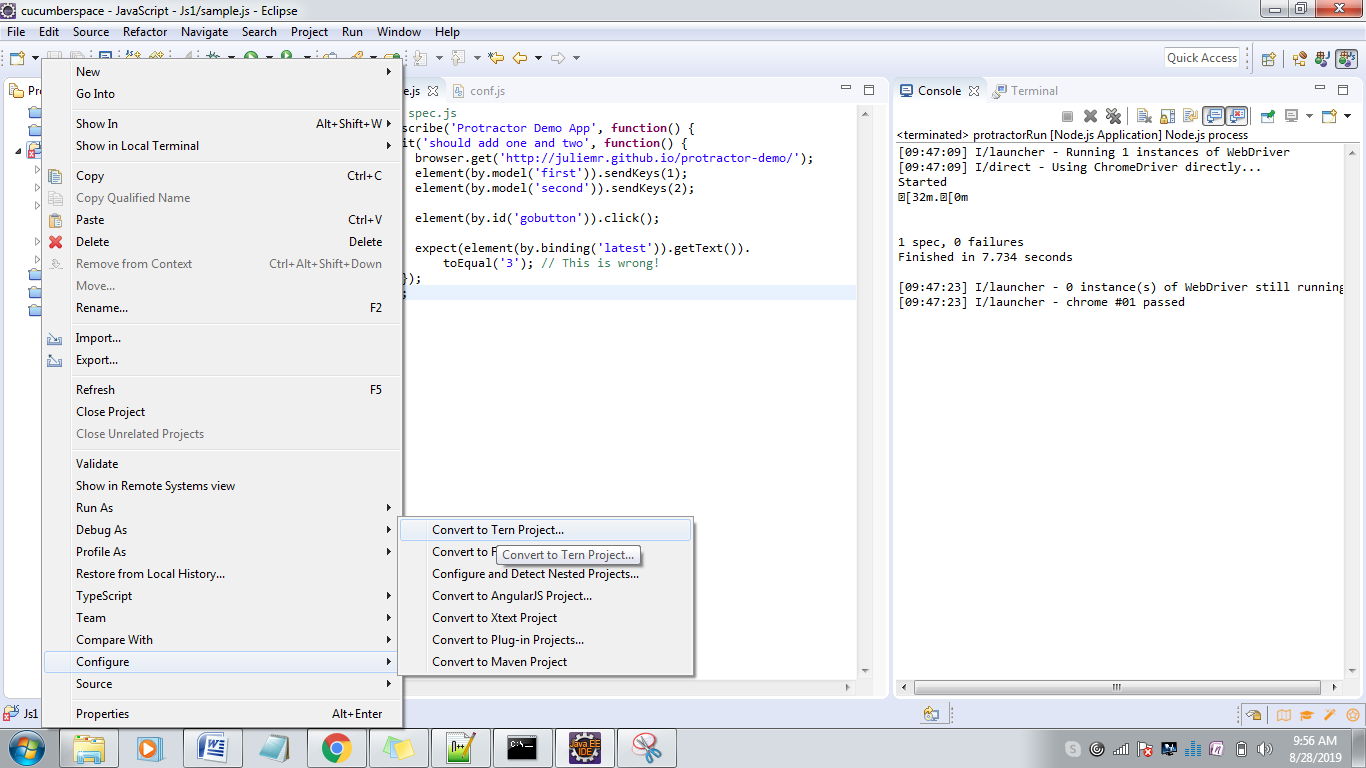


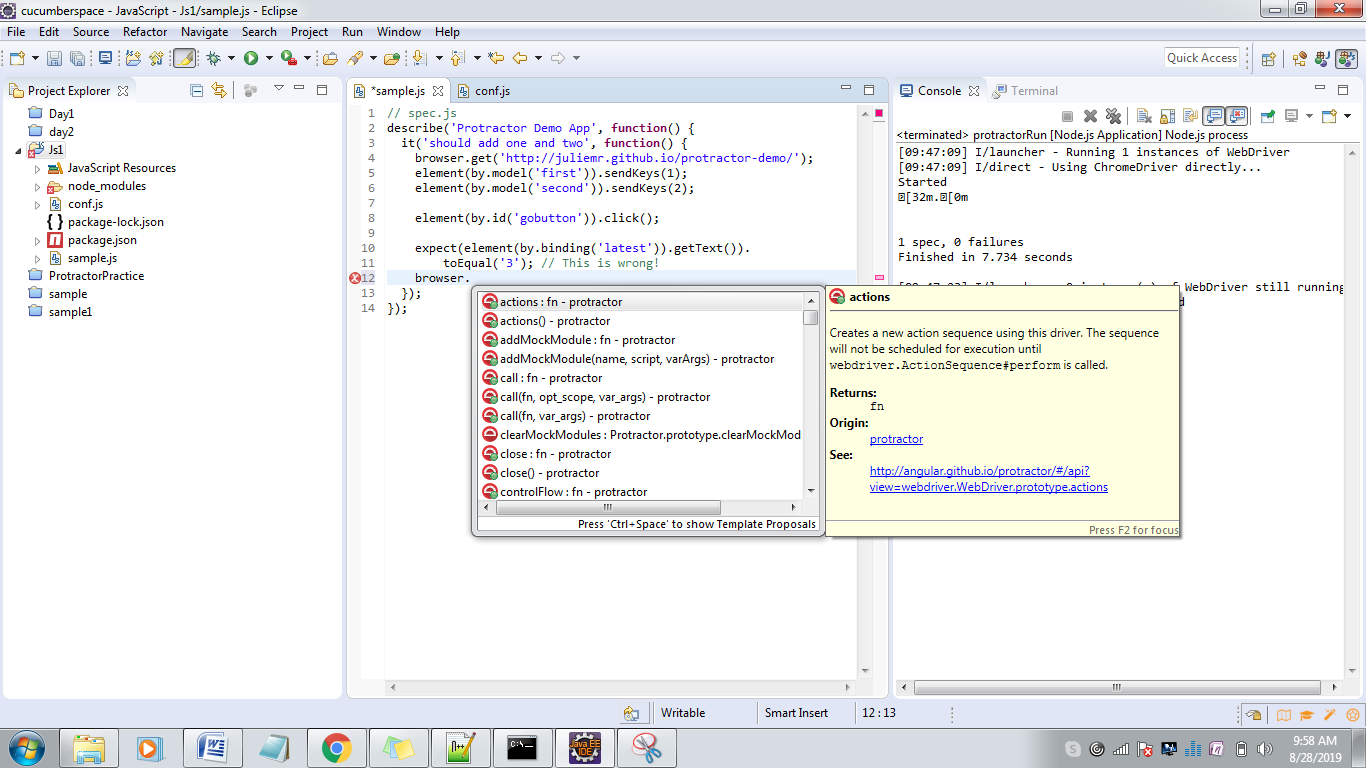




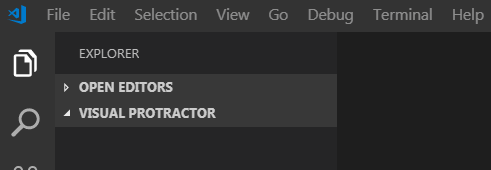




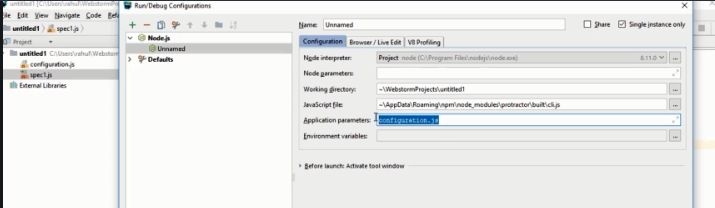




Install Microsoft visual studio code then create a new folder( File - > new folder)



Webstrom



https://spin.atomicobject.com/2014/12/17/asynchronous-testing-protractor-angular/

https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/

https://bridge360blog.com/2015/05/05/improving-protractor-tests-using-shared-functions-and-promises/

javascript is Asynchronous

Every Javascript step returns promise

Promise resembles state of your step

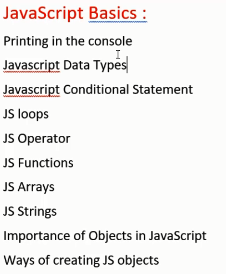
1. pending 2 resolved 3 Rejected

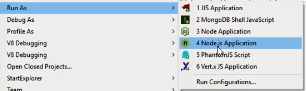
synchronous: you will move to next step only after promise for current steps is either resolved or rejected

Asynchronous : javascript movies to next step even if promise is pending

90% all the action performed in the browser out protractor API will not move to next step until promise resolve

if you want to retrieve anything from browser then that particular functionality related methods have not support of promise resolving



To Run javascript files

Console.log(“Hello world !”);

Global variables

Browser , Element , By ,Protractor

Locators

Id , name , css , model , binding

Css

Tagename[attribute=value]

Console.log(element(by.css(“h2[class=’nig-binding’]”)).getText(); // real output is not coming here

Console.log(element(by.css(“h2[class=’nig-binding’]”)).getText().then(function(text){

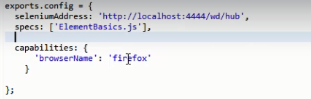
Console.log(“”+ text)

});

// jasmine will take care of promises

Expect(element(by.css().getText().toBe(“”));

Execute cases in Firefox



1. // An example configuration file
2. exports.config = {
3. // The address of a running selenium server.
4. seleniumAddress: 'http://localhost:4444/wd/hub',
6. // Capabilities to be passed to the webdriver instance.
7. capabilities: {
8. browserName: 'chrome'
9. },
11. // Spec patterns are relative to the configuration file location passed
12. // to protractor (in this example conf.js).
13. // They may include glob patterns.
14. specs: ['example-spec.js'],

17. };

Webdriver-manager update –ie

.verify page is angular or not

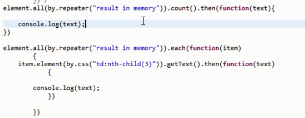
Browser.waitforangularenabled(false)

Chain locators

browser.get('http://juliemr.github.io/protractor-demo/');  
//repeater ,  chain locators, And css for identical tags  
element(by.model("first")).sendKeys("3");  
element(by.model("second")).sendKeys("5");  
  
element(by.id("gobutton")).click();

element(by.repeater("result in memory")).element(by.css("td:nth-child(3)")).getText().then(function(text)  
{  
  
console.log(text);  
})

using element.all



element.all(by.repeater("result in memory")).each(**function**(item) {

item.element(by.css("td:nth-child(3)")).getText().then(

**function**(text) {

console.log(text);

})

})

})

element.all(by.tagName("option")).each(**function**(item) {

item.getAttribute("value").then(**function**(values)

{

**if** (values == c) {

item.click();

}

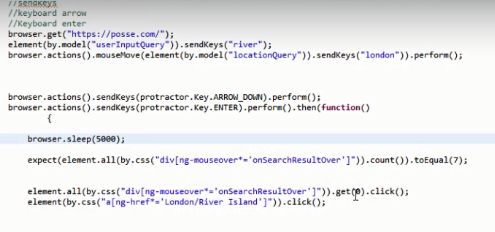
})

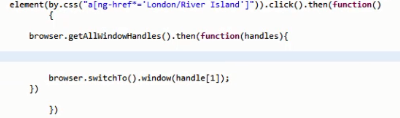
})

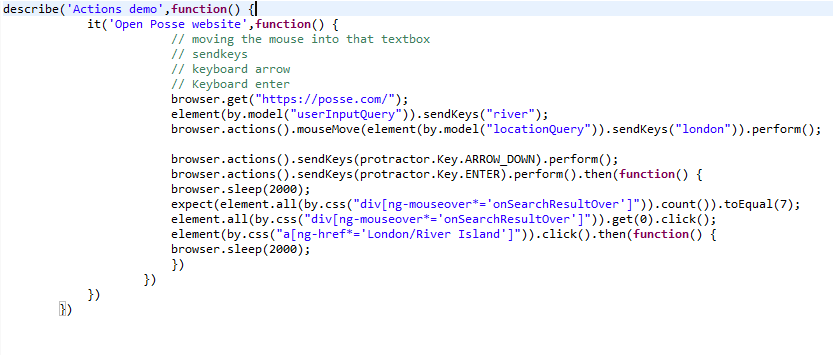
Upwork

Actions

// moving the mouse into the textbox







<http://www.qaclickacademy.com/>

work with Alerts – accept & dismiss

browser.waitForAngularEnabled(false);

browser.switchTo().alert().dismiss().then(function(){  
browser.sleep(5000); })

working with Frames

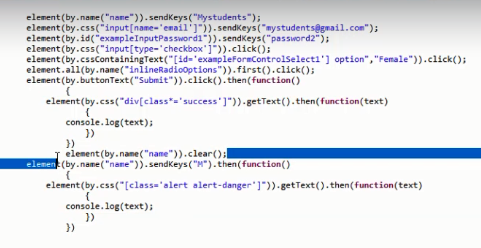


Page synchronization application



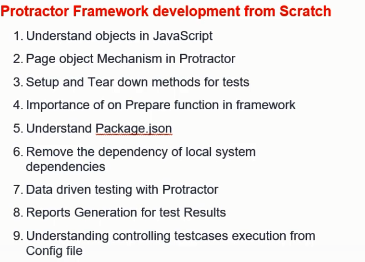


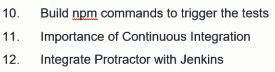
<https://qaclickacademy.github.io/protocommerce/>

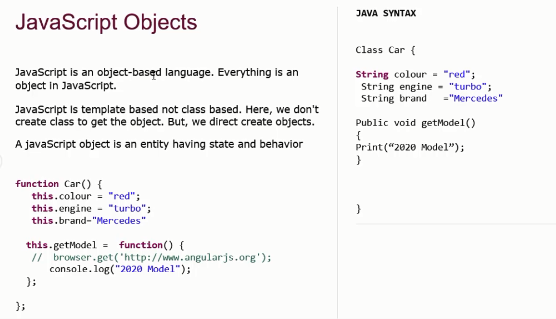


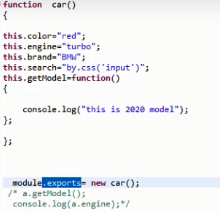








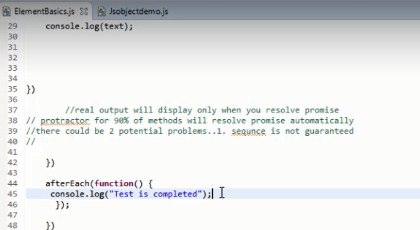


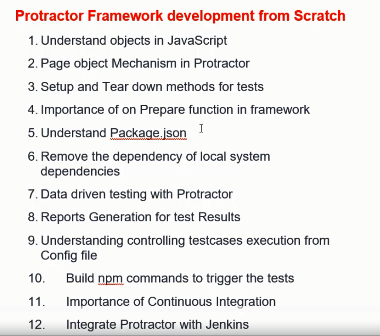




Console.log(obj.title);

beforeEach , afterEach



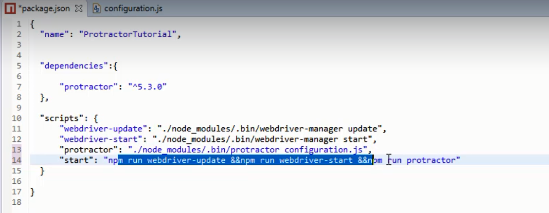


* Npm init –f

Command will create package.json with default values

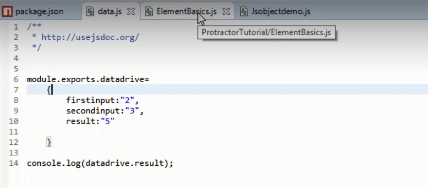
* Npm install

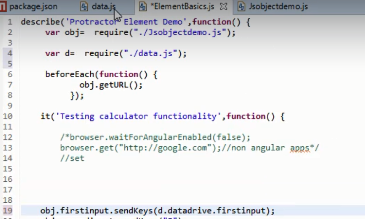
Command will execute and install required software’s which are mentioned in package.json

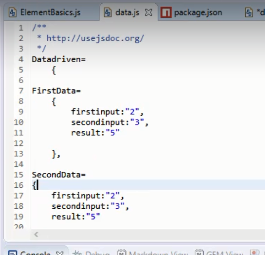


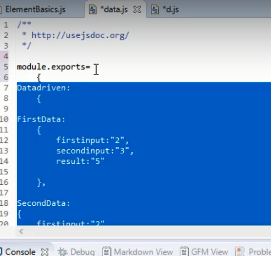
Datadriven testing using jasmine

* Jasmine –data provider

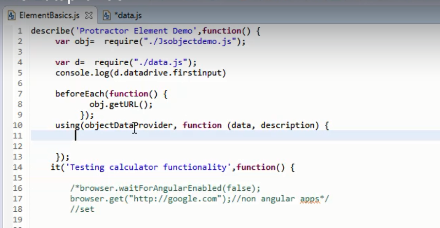


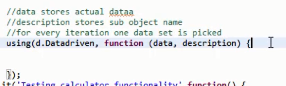


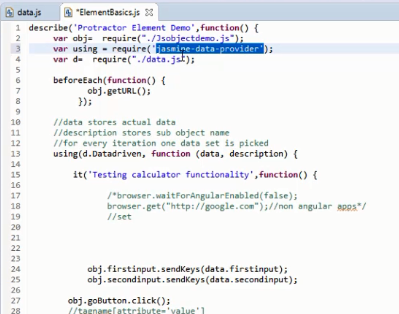




Jasmine dataprovider

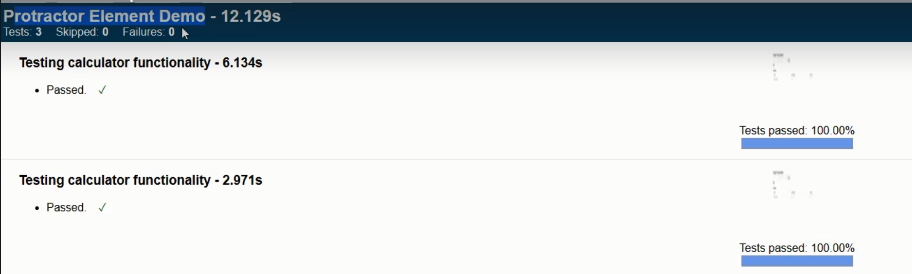


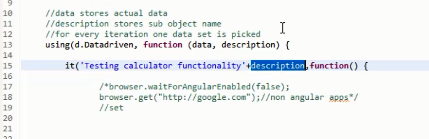


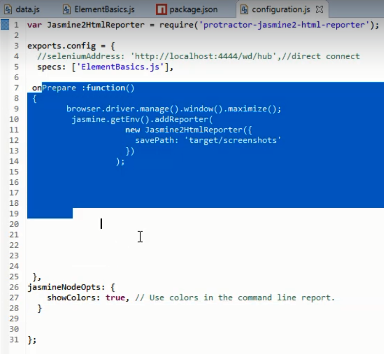


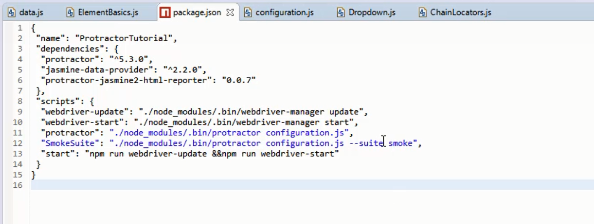


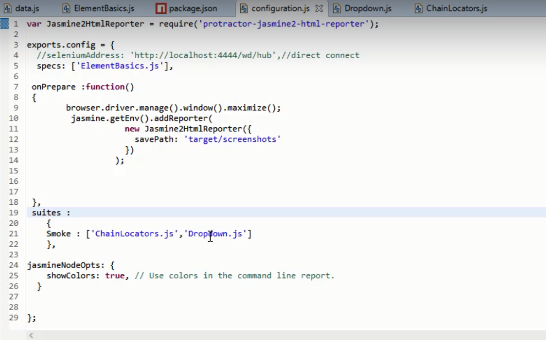


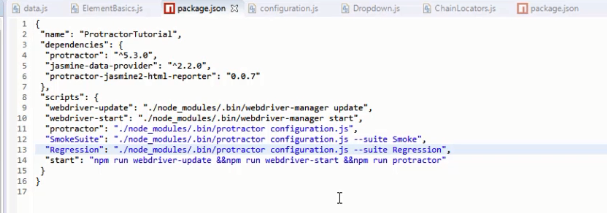




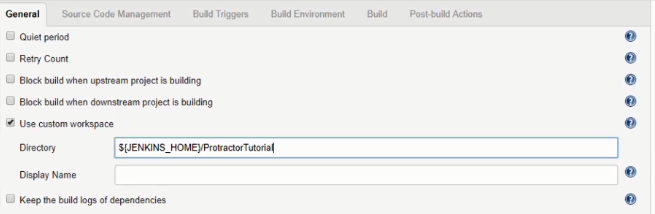


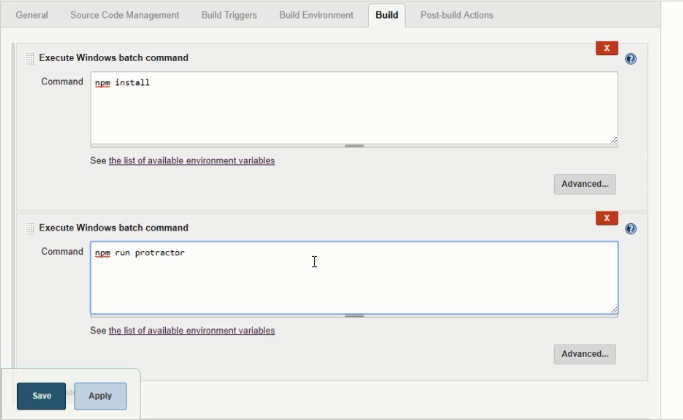


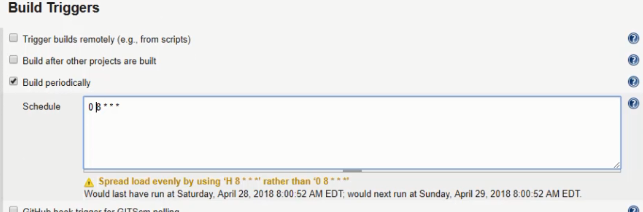


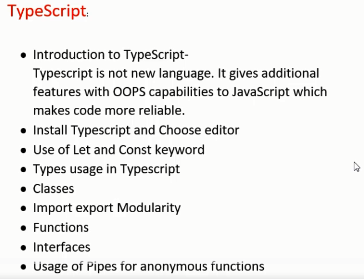










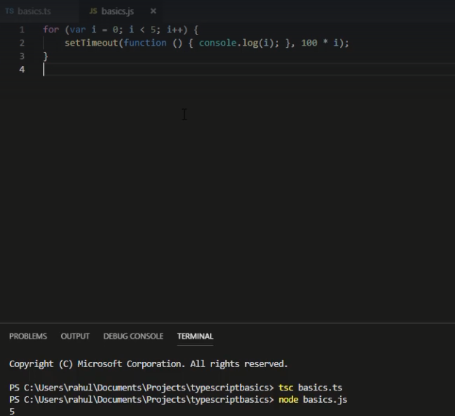


Npm install –g typescript

Keywords

Let/const

* Tsc helloworld.ts



Tsc abc.ts then after node abc.js

Usage of Let keyword, While using let script will run synchronously

Constant declaration

Let a:string = “Hello”

Let b:number = 400 // typescript

Var a =10 // javascript

Array declaration

Let list:Array<number>= [2,3,5]

Let dynamic:any;

Let value:Boolean = true

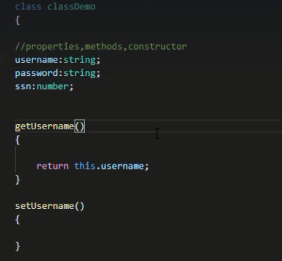
Compile program using tsc filename.ts

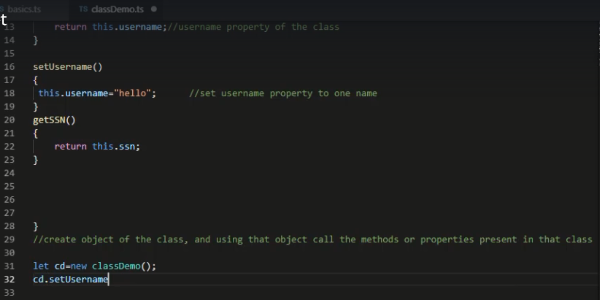
Run program node filename.js

Class Helloworld {

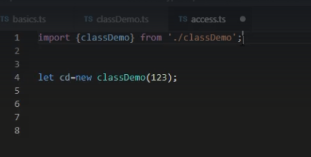
// properties , methods, constructor

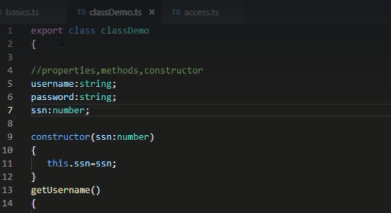
}





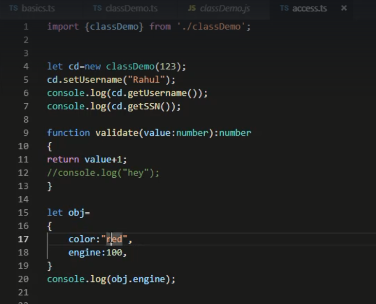
Import class



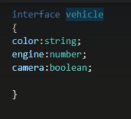


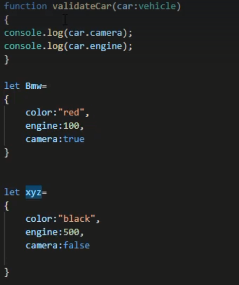


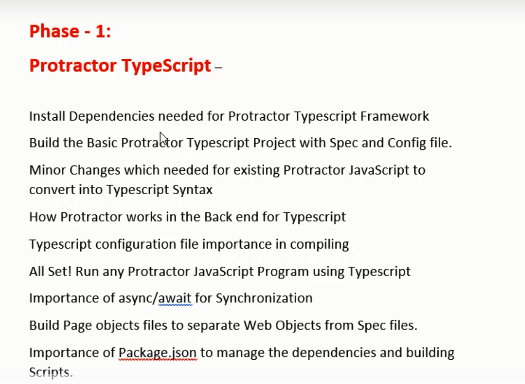




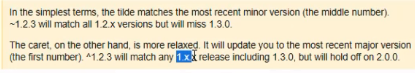
Interface



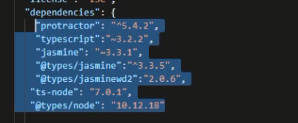


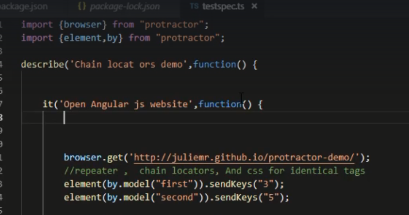


Jasmine cant under by default typescript



Required dependencies







Tsconfig.json file is mandatory to compile typescript file

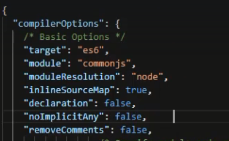
Install typescript globally and then run below command to create tsconfig.json file

* Tsc –init

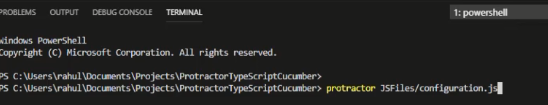
Tsconfig.json file will be useful to convert typescript into javascript file

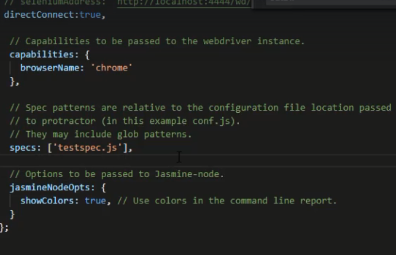
"target": "es5",

Specify javascript enginee



“outDir”: “JSfiles”





Async and await concept

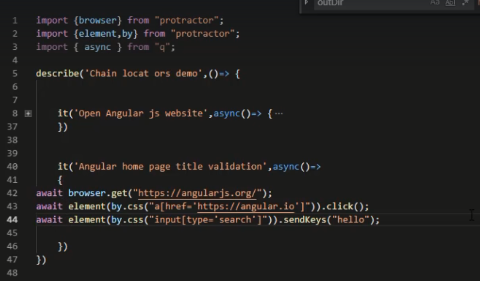
Webdrivercontrol deprecation

Promise chaining

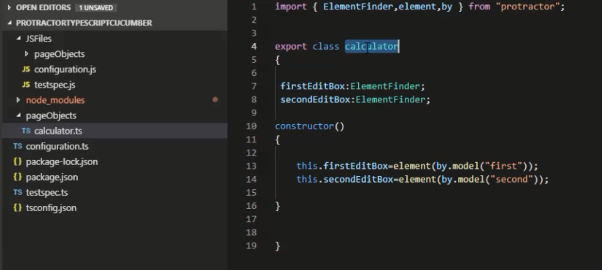
Fat arrow concept

ES7 javascript enginee

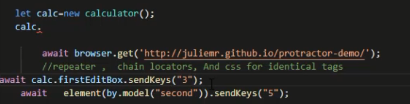
Including await and async concepts

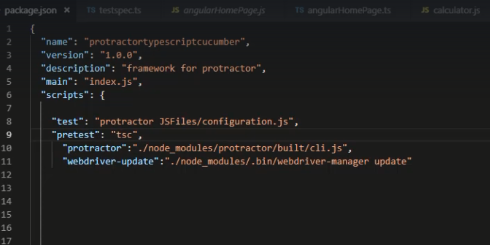


Define elements in different class

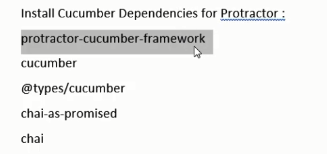




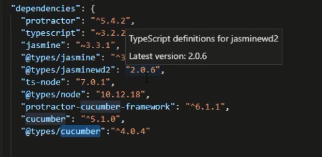




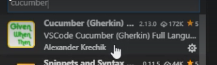
Cucumber – behavior driven development tools



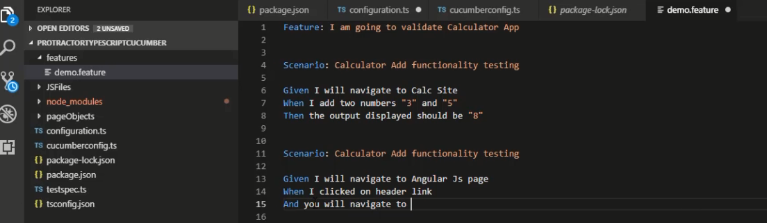
Package.json file dependencies

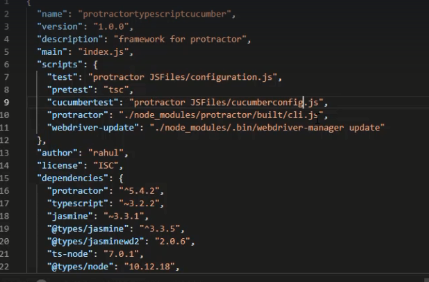


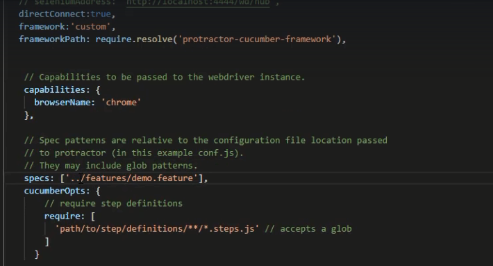
Visual studio code extension

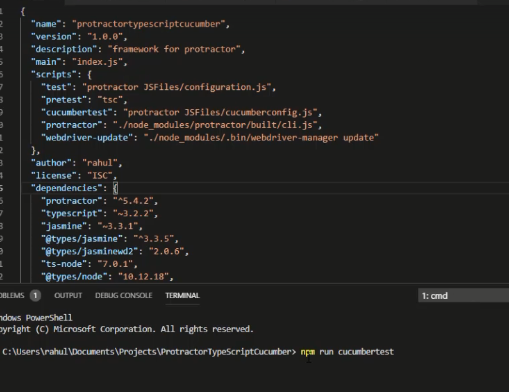


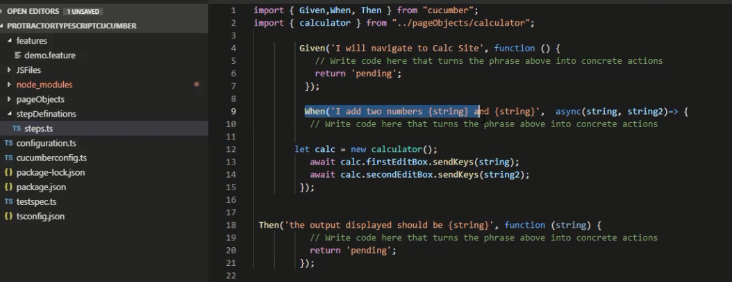
<https://github.com/protractor-cucumber-framework/protractor-cucumber-framework>

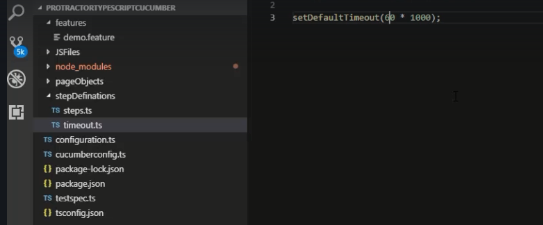


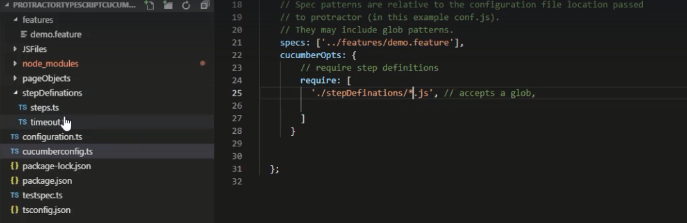




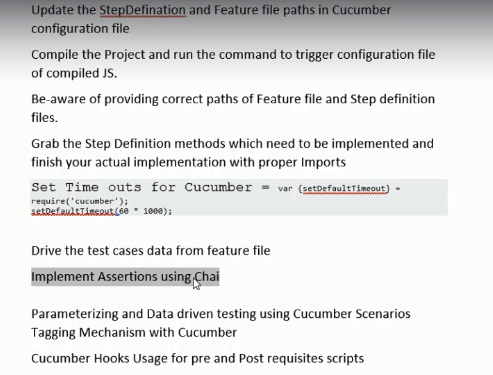




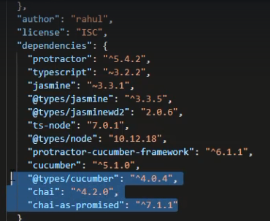




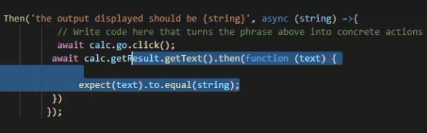
CTRL+ ? Comment and uncomment the things



Chai – assertion framework. While using cucumber , jasmine will not assist here. So jump into chai framework

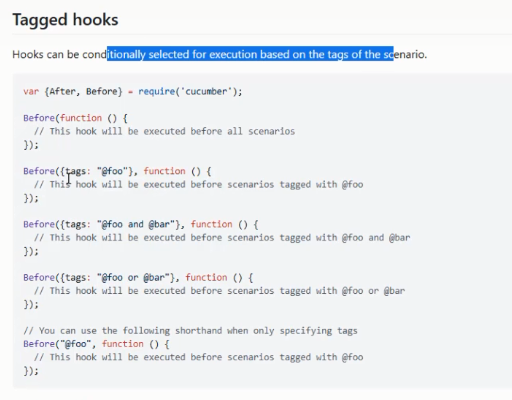


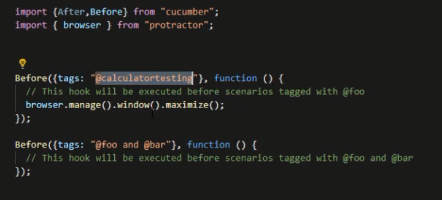


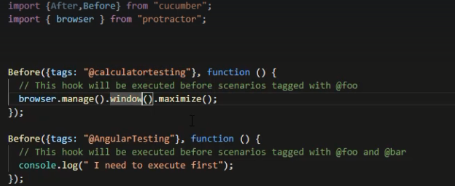


Hooks

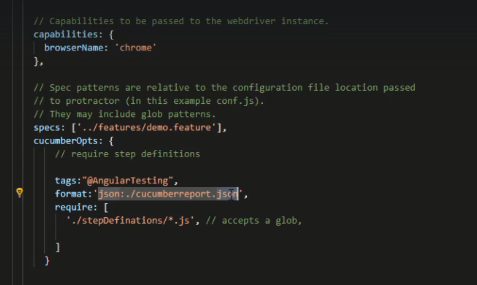
Navigate to cucumberjs





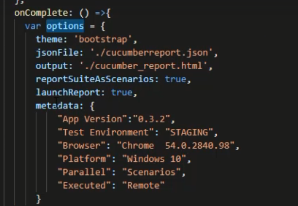


Cucumber-html-report



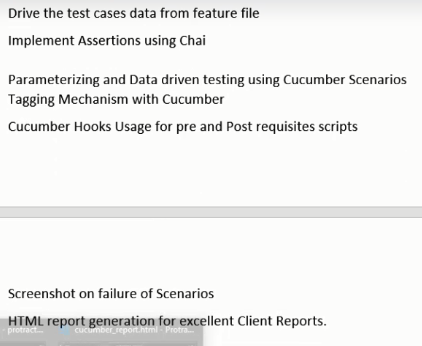


Following code we have to write in config file along with browser options

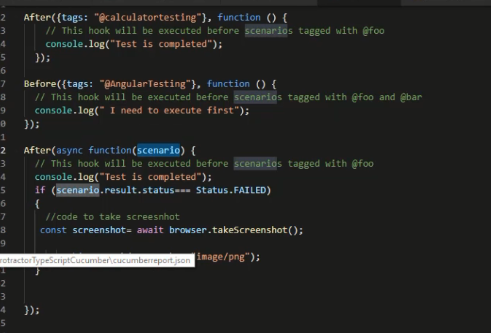


Import every thing from package





Inside hooks class



[raghulonlinetutor@gmail.com](mailto:raghulonlinetutor@gmail.com)