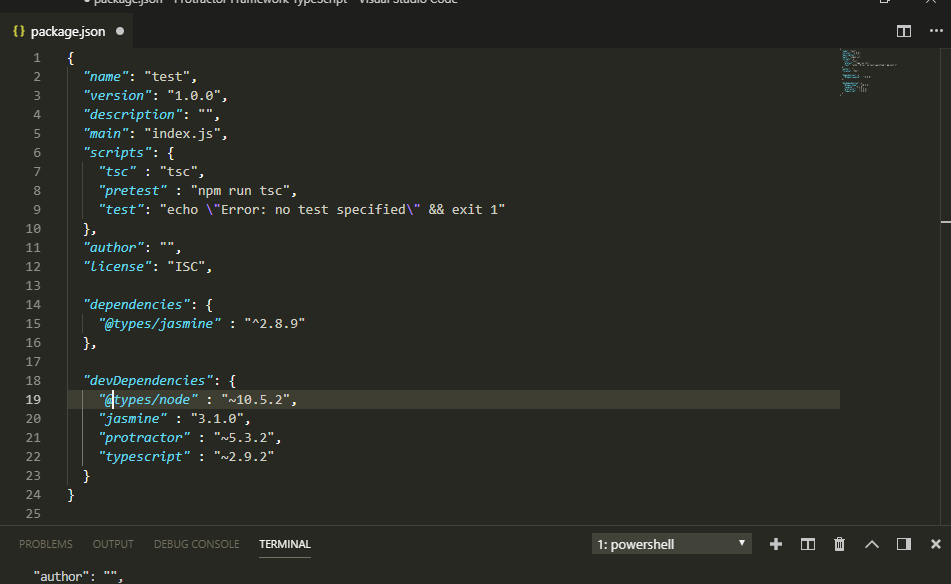
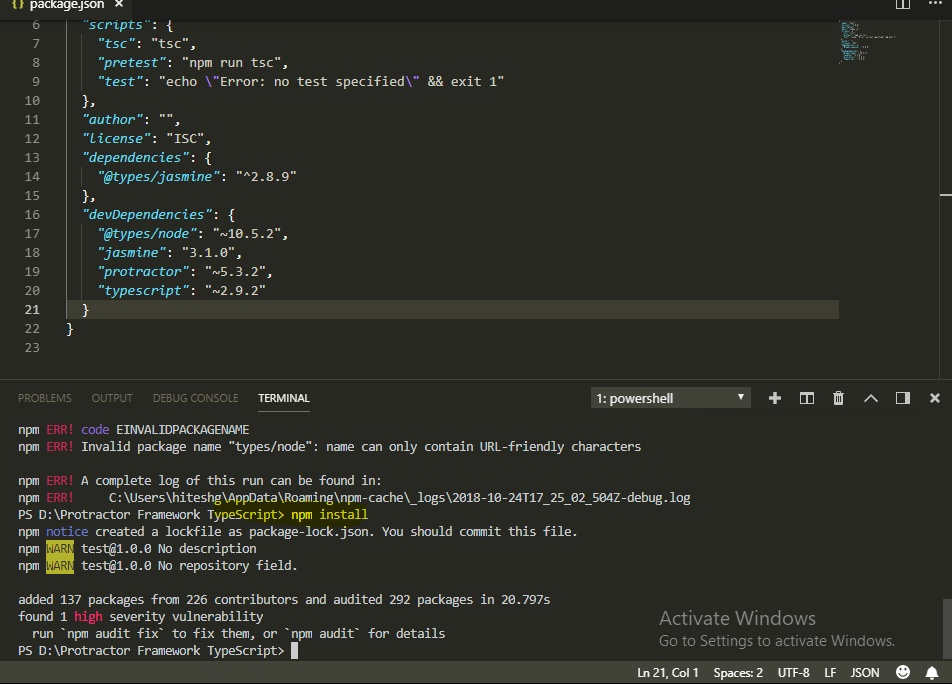
Step 2:

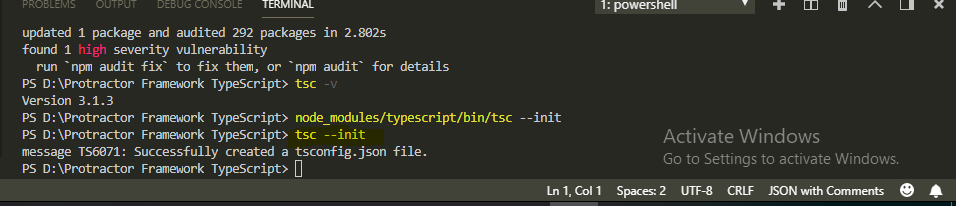


Step 3: Issue npm install

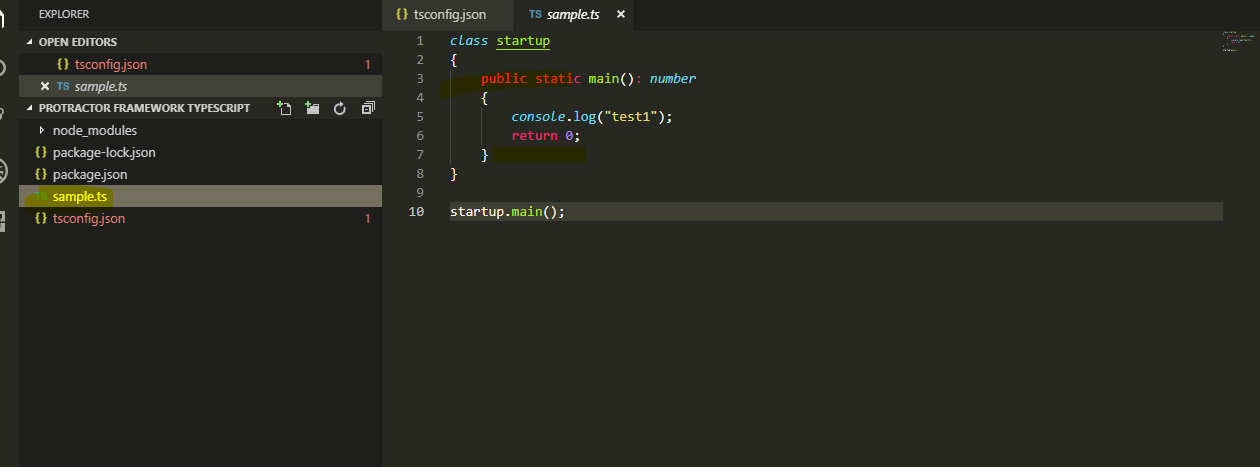


Step 3: Creation of tsconfig.json file by issuing the following command

Tsc -init



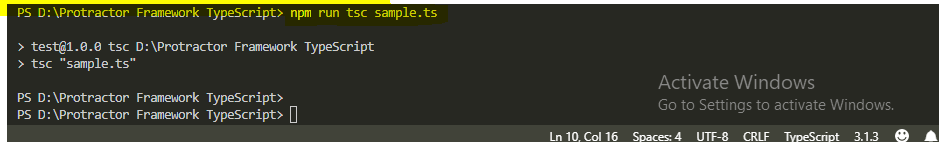
Step 4: Now create a sample .ts file as follow:



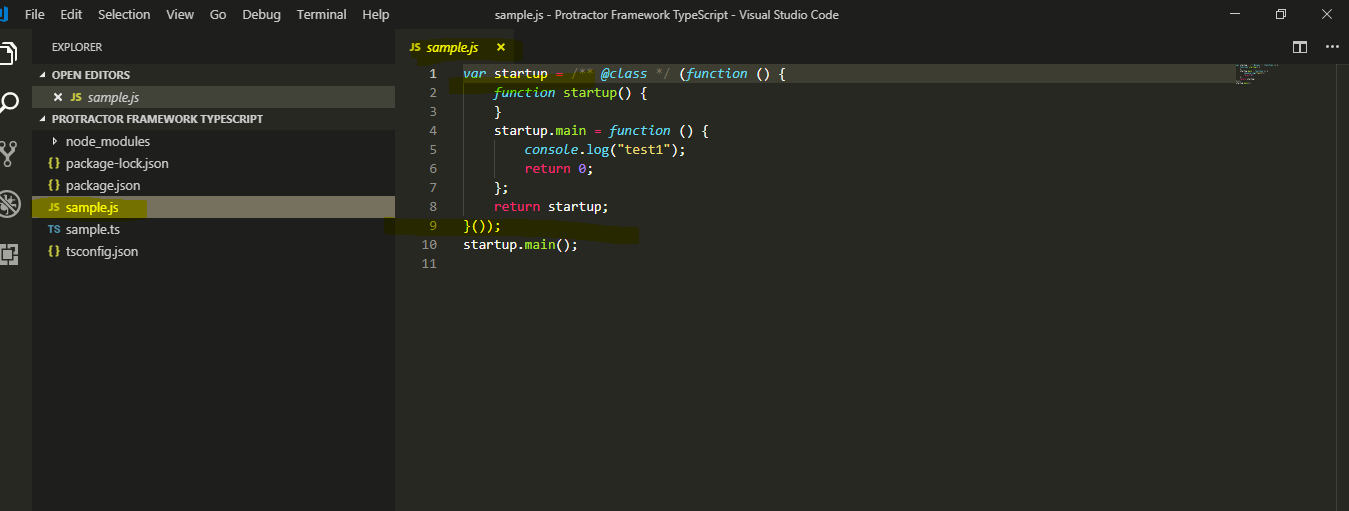
Step 5: Use the following command to convert .ts file into .js (java script) file

Npm run tsc sample.ts

We have to use above command if type script is installed at local level if type script is installed as global level then use ‘tsc sample.ts’ command



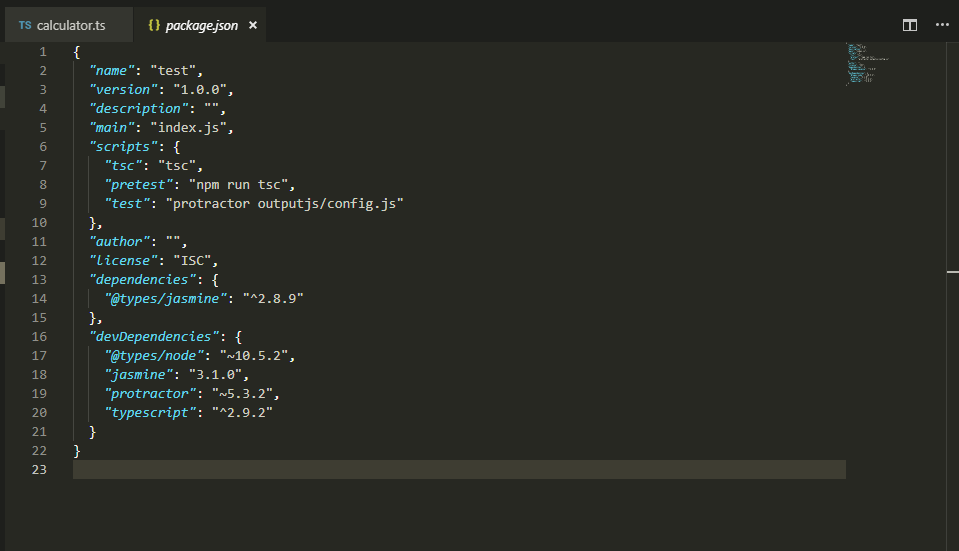
After running the above command, the sample.js (a javascript file) will be generated as follow:



Now create a protractor script

Project Structure:

1. Package.json:



{

*"name"*: "test",

*"version"*: "1.0.0",

*"description"*: "",

*"main"*: "index.js",

*"scripts"*: {

*"tsc"*: "tsc",

*"pretest"*: "npm run tsc",

*"test"*: "protractor outputjs/config.js"

},

*"author"*: "",

*"license"*: "ISC",

*"dependencies"*: {

*"@types/jasmine"*: "^2.8.9"

},

*"devDependencies"*: {

*"@types/node"*: "~10.5.2",

*"jasmine"*: "3.1.0",

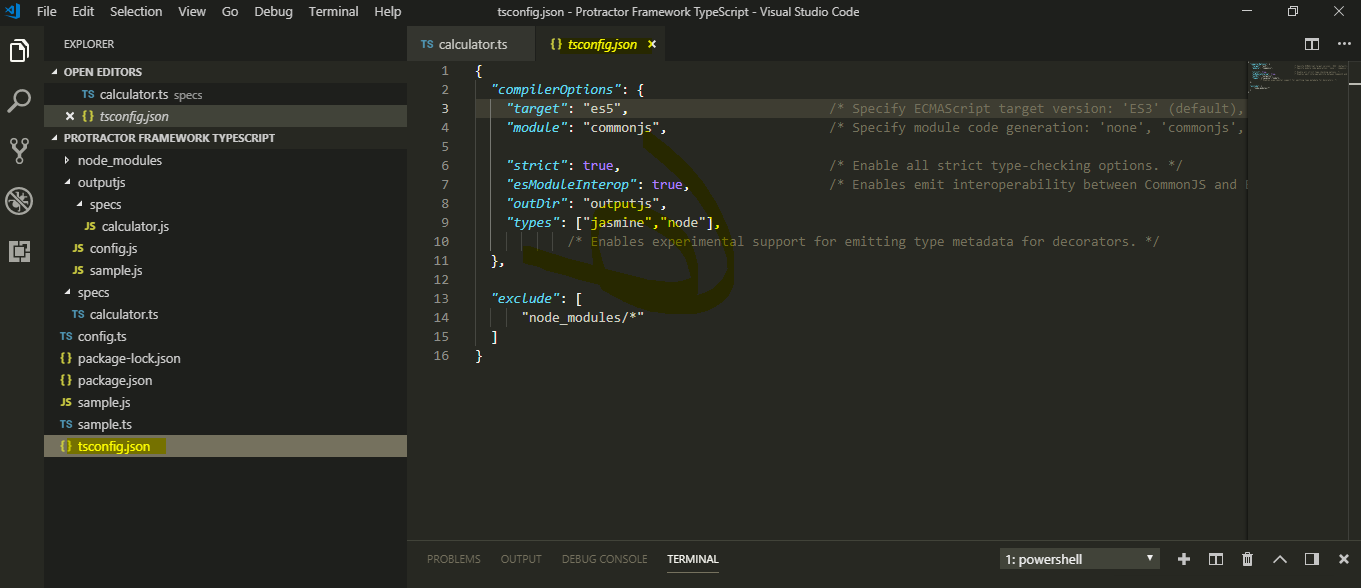
*"protractor"*: "~5.3.2",

*"typescript"*: "^2.9.2"

}

}

1. Tsconfig.json:



{

*"compilerOptions"*: {

*"target"*: "es5", /\* Specify ECMAScript target version: 'ES3' (default), 'ES5', 'ES2015', 'ES2016', 'ES2017','ES2018' or 'ESNEXT'. \*/

*"module"*: "commonjs", /\* Specify module code generation: 'none', 'commonjs', 'amd', 'system', 'umd', 'es2015', or 'ESNext'. \*/

*"strict"*: true, /\* Enable all strict type-checking options. \*/

*"esModuleInterop"*: true, /\* Enables emit interoperability between CommonJS and ES Modules via creation of namespace objects for all imports. Implies 'allowSyntheticDefaultImports'. \*/

*"outDir"*: "outputjs",

*"types"*: ["jasmine","node"],

/\* Enables experimental support for emitting type metadata for decorators. \*/

},

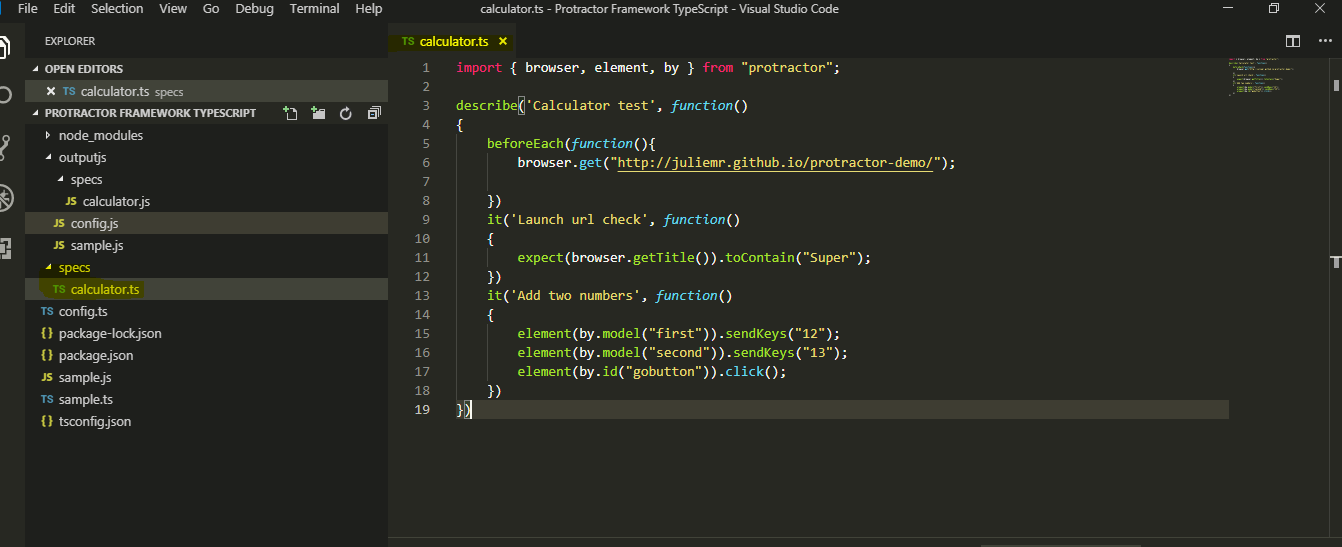
*"exclude"*: [

"node\_modules/\*"

]

}

Step 3: Now create a ‘specs’ folder at project level and create test script inside it say ‘calculator.ts’. Please note the extension of script, it is ‘.ts’ (type script) (Actually this type script will be converted into java script which will be run by the browser because browser does not recognize any type script file.



import { browser, element, by } from "protractor";

describe('Calculator test', *function*()

{

beforeEach(*function*(){

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

})

it('Launch url check', *function*()

{

expect(browser.getTitle()).toContain("Super");

})

it('Add two numbers', *function*()

{

element(by.model("first")).sendKeys("12");

element(by.model("second")).sendKeys("13");

element(by.id("gobutton")).click();

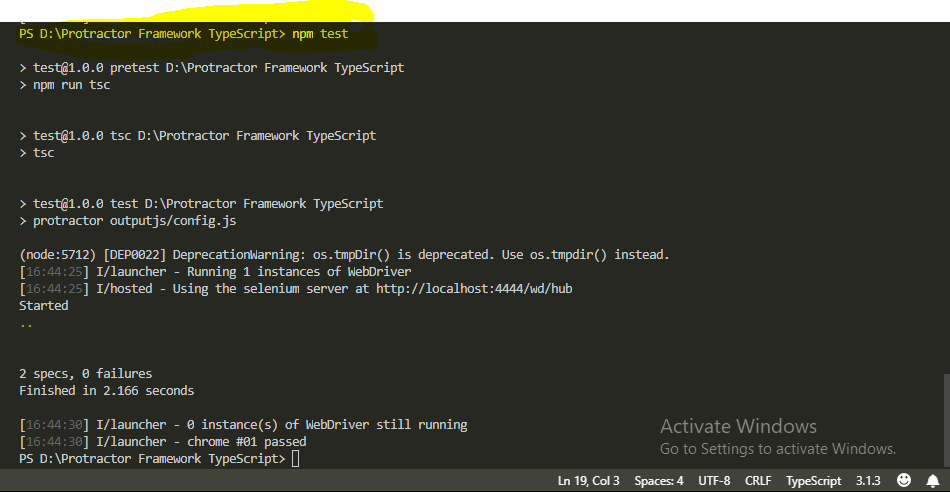
})

})

Note: Before running the script, please make sure server is running. Issue the following command to run the webdriver server

$ webdriver-manager start

Step 4: Now run ‘npm test’ command. This command will first convert the .ts files into .js files and then run the script.



Note: How to run the multiple scripts residing in specs folder.

