

**Protractor TypeScript TESTING framework TOOL Documentation**

[TCoE- Automation CoP]

Version 0.1

Reference No.:

Version:

Initial Release Date:

**Date of Current Version:**

Version Control Notice

This document is a controlled document that superseded all previous versions. Please discard any previous copies of this document dated prior to the version and publication date noted above this page.

Standards for creating, updating, approving, controlling and disposing of documents are contained in the CTO Document Control Process located at: <https://www.deloitteresources.com/pgContent.aspx?sid=40795&cid=285608>

Anyone who obtains a printed version of this document is responsible for ensuring that they have the latest version which is listed on the Global Document Register (GDR) located at: https://er34.deloitteonline.com/eRoom/GlobalIM/GlobalDocumentRegister/0\_1809e

Proprietary Notice

This document contains proprietary information of Deloitte Touche Tohmatsu Services, Inc. Unauthorized copying, distribution, or dissemination of the information contained herein is a violation of the organizations policies and proprietary rights.

General Notice

Product names used herein are for identification purposes only, and may be trademarks of their respective companies.

Ownership and Master Copy

|  |  |
| --- | --- |
| Department that owns this document: | Testing COE – Automation COP |
| Incumbent manager of the department: |  |
| Manager’s title: |  |
| Manager’s email address: |  |
| Manager’s phone number: | +1- |
| Master copy storage location: | Insert URL to document record in Global Document Register |
| Document Template version | 8.0 |

Approval History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version: | Reviewed By: | Approved By: | Approver’s Position: | Date Approved: | Next Review Date: |
| X.X | Name | Name | Role | dd-mmm-yy | dd-mmm-yy |
| X.X | Name | Name | Role | dd-mmm-yy | dd-mmm-yy |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Updated By: | Date: | Description of changes: |
| 0.1 | Mubeen Ali | 10/10/2019 | draft |
|  |  |  |  |
|  |  |  |  |

NOTE: Documents which are in draft mode and have never been released, should always have a version <1.0. Increments to the drafts are denoted with a letter. Only when a document is approved and released can it have a version of 1.0 or higher. Also note that minor changes increment the version by 0.1 while major changes increment the version by 1.0. For complete details and examples on versioning, please refer to the Versioning Guideline appendix in the CTO Document Control Process found at: <https://www.deloitteresources.com/pgContent.aspx?sid=40795&cid=285608>

Table of Contents

[1. Purpose 5](#_Toc21959461)

[1.1 Objectives 5](#_Toc21959462)

[1.2 Team Members 5](#_Toc21959463)

[2. Basics - Protractor and TypeScript 6](#_Toc21959464)

[2.1 What is Protractor 6](#_Toc21959465)

[2.2 What is Angular Application 6](#_Toc21959466)

[2.3 Why Protractor? 6](#_Toc21959467)

[2.4 Can Protractor be used to automate Non-Angular Apps? 6](#_Toc21959468)

[2.5 What language does Protractor support? 6](#_Toc21959469)

[2.6 What is TypeScript? 7](#_Toc21959470)

[3. Installations Guide 7](#_Toc21959471)

[3.1 JDK Installation 7](#_Toc21959472)

[3.1.1 Un-Install Older Version(s) of JDK/JRE 7](#_Toc21959473)

[3.1.2 Download JDK 7](#_Toc21959474)

[3.1.3 Install JDK 7](#_Toc21959475)

[3.1.4 Setup Environment Variables 8](#_Toc21959476)

[3.2 Node.js Installation 11](#_Toc21959477)

[3.2.1 Installation Steps & Setup 11](#_Toc21959478)

[3.2.2 Setup Environment Variables 12](#_Toc21959479)

[3.3 Protractor Installation 14](#_Toc21959480)

[3.3.1 Installation Steps & Setup 14](#_Toc21959481)

[3.4 WebDriver Update and Start 16](#_Toc21959482)

[3.4.1 Installation Steps & Setup 16](#_Toc21959483)

[3.5 Visual Studio IDE Installation 17](#_Toc21959484)

[3.5.1 Installation Steps & Setup 17](#_Toc21959485)

[3.6 TypeScript Installation 18](#_Toc21959486)

[3.6.1 Installation Steps & Setup 18](#_Toc21959487)

[3.7 Database Setup 18](#_Toc21959488)

[3.7.1 Download MySql Workbench 18](#_Toc21959489)

[3.7.2 Install the MySql Workbench on the hub /your local machine. 18](#_Toc21959490)

[3.7.3 Once Installed open the MySql Workbench from your program files. 18](#_Toc21959491)

[3.8 GIT Installation 18](#_Toc21959492)

[3.8.1 Installation Steps and Setup 18](#_Toc21959493)

[4. User Guide 20](#_Toc21959494)

[4.1 Export the project to VS Code IDE 20](#_Toc21959495)

[4.2 Understanding of Framework Folder Structure 21](#_Toc21959496)

[4.3 Understanding the Framework Functionality 21](#_Toc21959497)

[4.3.1 Package.json 21](#_Toc21959498)

[4.3.2 Cucumberconf.ts 23](#_Toc21959499)

[4.3.3 Feature Files 24](#_Toc21959500)

[4.3.4 Stepdefinition.ts 25](#_Toc21959501)

[4.3.5 Hooks.ts 25](#_Toc21959502)

[4.3.6 Timeouts.ts 26](#_Toc21959503)

[4.3.7 PageObjects 26](#_Toc21959504)

[4.3.8 PageClasses 27](#_Toc21959505)

[4.3.9 Tsconfig.ts 28](#_Toc21959506)

[4.3.10 JSFiles Folder 28](#_Toc21959507)

[4.3.11 Utils 28](#_Toc21959508)

[4.3.12 Cucumber HTML Reports 29](#_Toc21959509)

[5. Run Book 31](#_Toc21959510)

[5.1 Framework Flow 31](#_Toc21959511)

[5.2 Pre-requisite Database Queries to Run 32](#_Toc21959512)

[5.2.1 Creates following database with below command: 32](#_Toc21959513)

[5.2.2 Creates following table with below command: 32](#_Toc21959514)

[5.2.3 Provide the username and password to the database 32](#_Toc21959515)

[5.2.4 Insert values to the table testdata 32](#_Toc21959516)

[5.3 Execution – Test Run 32](#_Toc21959517)

[5.4 Post Execution 33](#_Toc21959518)

[5.4.1 Run the query to fetch the end result 33](#_Toc21959519)

[5.4.2 Clears the data in testdata table 33](#_Toc21959520)

[6 Reference Documents 33](#_Toc21959521)

# Purpose

## Objectives

Objective of this document is to provide instructions on

1. Brief Description on Protractor and the major difference between Protractor and Selenium
2. Why Protractor is implemented using TypeScript and not JavaScript
3. How to setup and install Protractor and related components
4. How to setup the framework and run sample testcases

## Team Members

|  |  |
| --- | --- |
| **Resource Name** | **Role** |
|  |  |
|  |  |
|  |  |

# Basics - Protractor and TypeScript

## What is Protractor

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.

***WebDriver + Angular Support = Protractor***

## What is Angular Application

Angular is a platform and framework for building client applications in HTML and TypeScript. The organization who built Angular Applications developed Protractor with TypeScript to test their Applications

Currently 80% of the Applications are built using cutting edge technologies like Angular Framework, Vue and React

## Why Protractor?

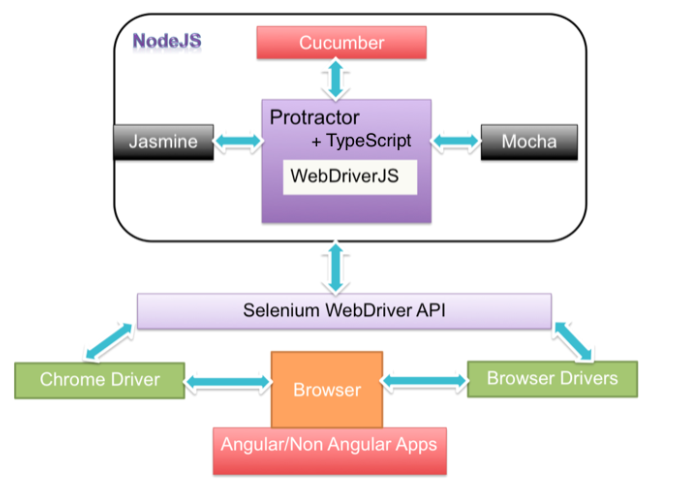
Selenium with Java/Python or C# can be used to automate Angular Applications like how normal applications are automated. However, Protractor has additional features which makes it more stable framework and takes an edge over Selenium when used for Angular Applications.

## Can Protractor be used to automate Non-Angular Apps?

Yes, it supports Regular Applications as well

## What language does Protractor support?

Protractor is a Node.js program built on top of WebDriverJS, so it uses JavaScript and TypeScript as core language to develop test.



## What is TypeScript?

TypeScript is a typed superset of JavaScript. A valid JavaScript code will work in TypeScript framework

***JavaScript + Additional Features (like OOPS) = TypeScript***

# Installations Guide

## JDK Installation

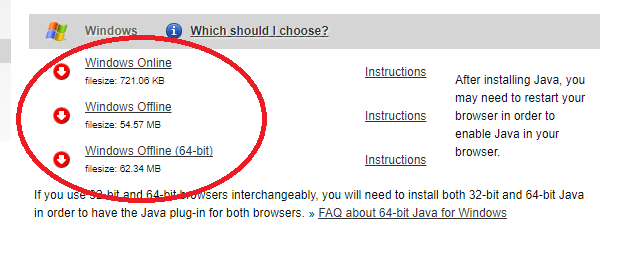
### Un-Install Older Version(s) of JDK/JRE

It is recommended to install only the latest JDK. If you have previously installed older version(s) of JDK/JRE, un-install ALL of them.

* + - 1. Go to "Control Panel"🡪 "Programs" 🡪"Programs and Features" ⇒ Un-install ALL programs begin with "Java", such as "Java SE Development Kit ...", "Java SE Runtime ...", "Java X Update ...", and etc.
      2. If you don’t have older JDK versions installed in your system, skip this step.

### Download JDK

1. Go to Java SE download site @ https://java.com/en/download/manual.jsp
2. Click on Windows Online



1. The File Download dialog box appears prompting you to run or save the download file
2. To run the installer, click Run. Or to save the file for later installation, click Save.
3. Choose the folder location and save the file to your local system.
4. The File Download dialog box appears prompting you to run or save the download file

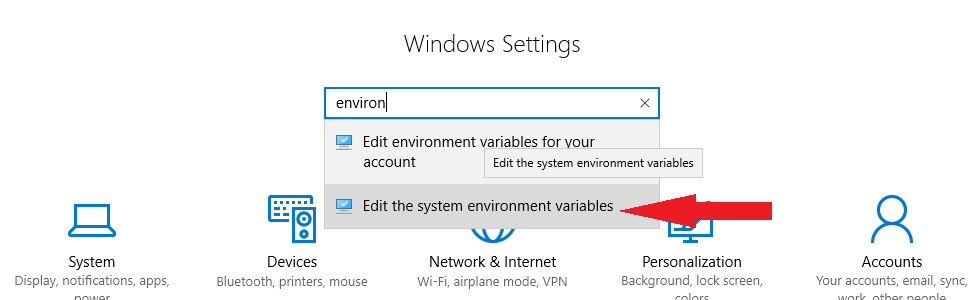
### Install JDK

1. Double-click on the saved file to start the installation process.
2. The installation process starts. Click the Install button to accept the license terms and to continue with the installation. 
3. Oracle has partnered with companies that offer various products. The installer may present you with option to install these programs when you install Java. After ensuring that the desired programs are selected, click the Next button to continue the installation.
4. A few brief dialogs confirm the last steps of the installation process; click Close on the last dialog. This will complete Java installation process.

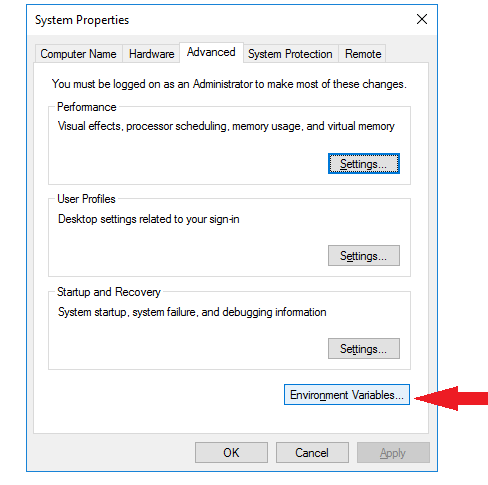


### Setup Environment Variables

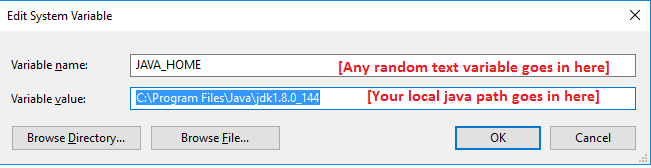
1. In Windows, Open settings and search for environment as shown below and click on the Edit the System environment variables option.



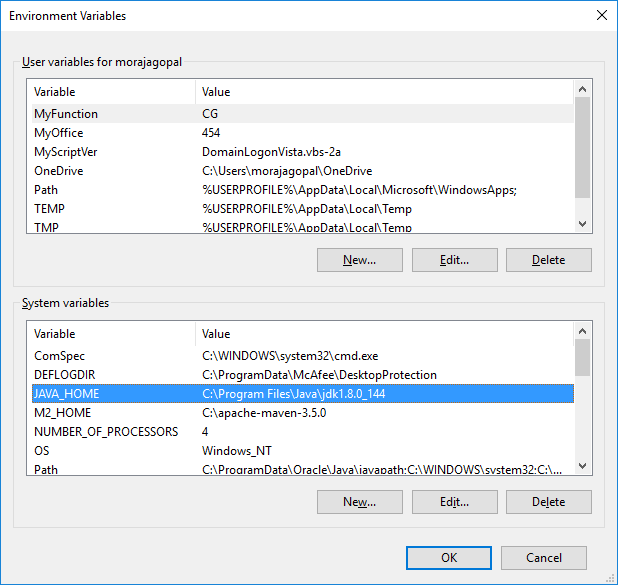
1. Click on the Environment variables button in System properties window.



1. Click on the New button under the system variables and enter a random variable name for your Java system variable and your exact local jdk path as shown below and click ok.

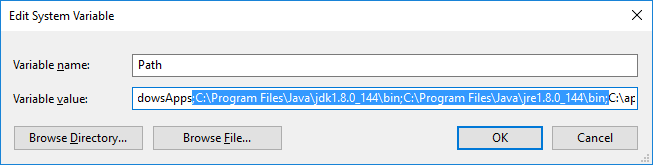


1. Your Java system variable will be added as shown below:



1. Select and double click on the Path system variable
2. Click on the Edit text button
3. **Append** your local jdk & jre directory path until bin folder starting with semi colon ‘;’ to the path variable value field as shown below and click ok to finish the installation setup.

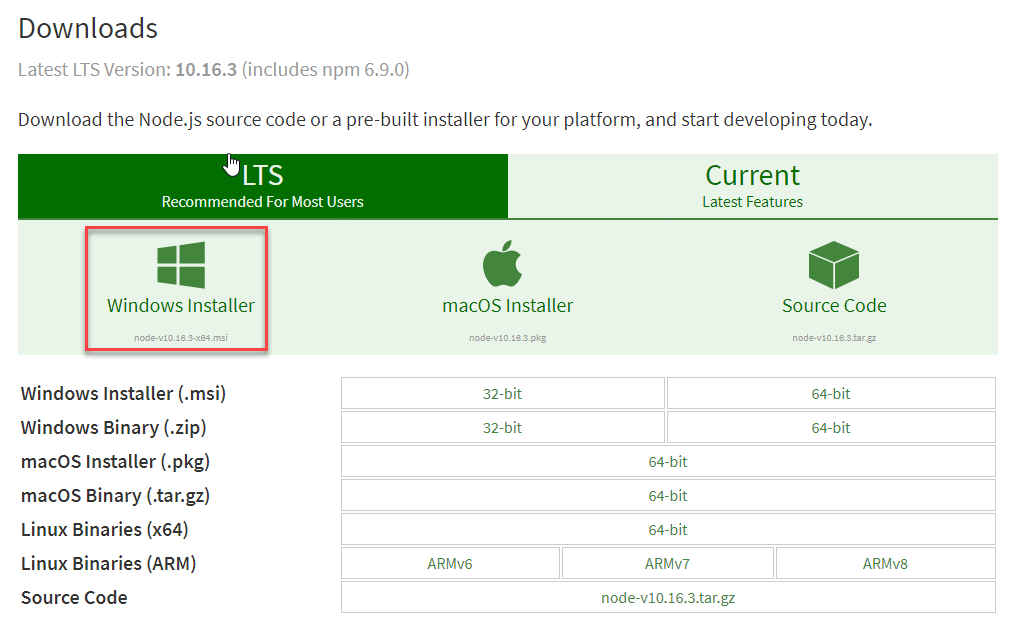
Note: Do not delete existing Variable value in your system



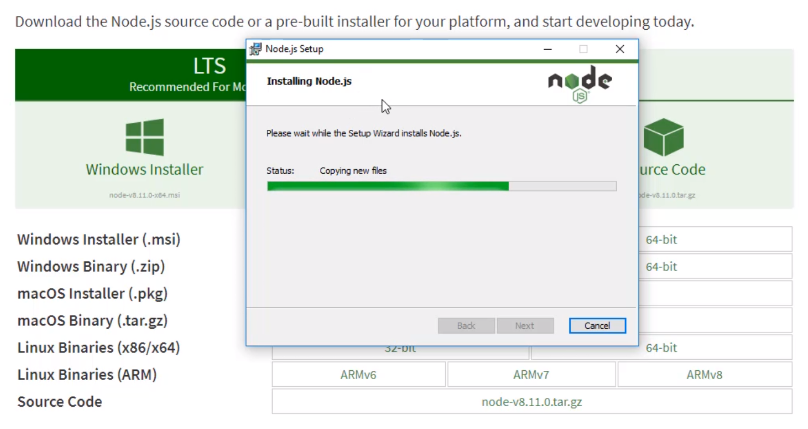
## Node.js Installation

### Installation Steps & Setup

1. Navigate to following URL in the browser. <https://nodejs.org/en/download/>
2. Click on Windows Installer to download the **node-v10.16.3-x64.msi** *(current version)* file



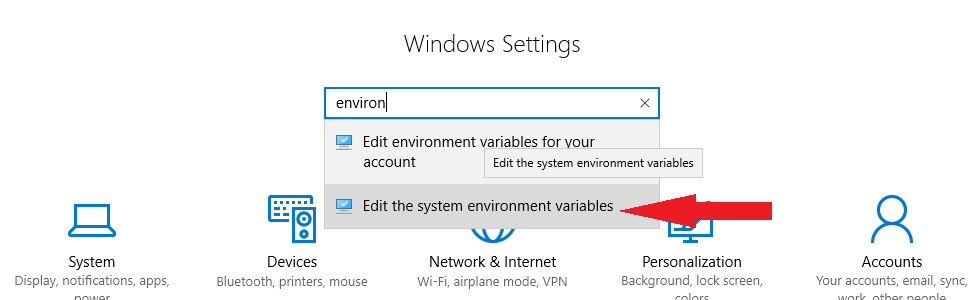
1. Unzip and run the executable file ‘node-v10.16.3-x64.msi’ *(current version)*
2. Click Next and Next to install the file (Do Not change the file path while installing)



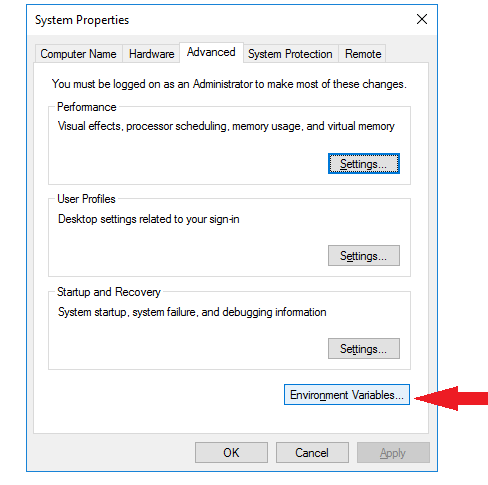
1. Click on Finish

### Setup Environment Variables

1. In Windows, Open settings and search for environment as shown below, click on the Edit the System environment variables option.



1. Click on the Environment variables button in System properties window.

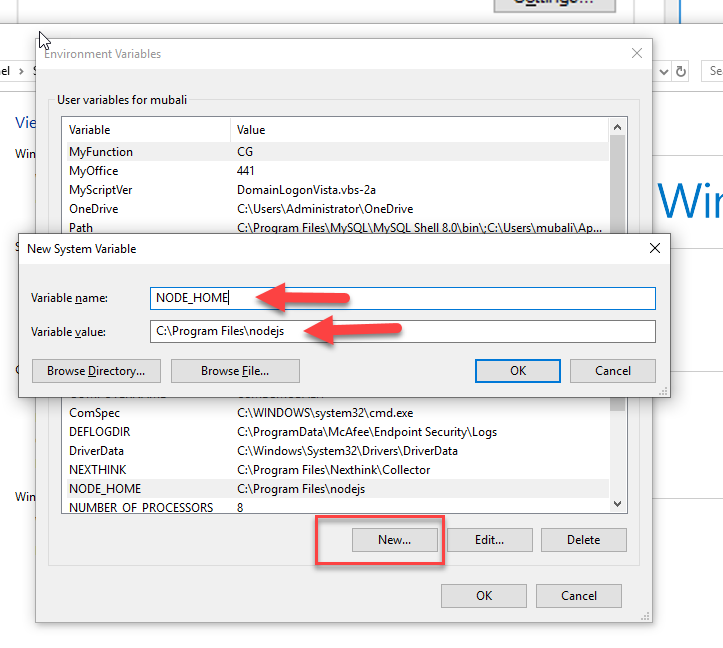


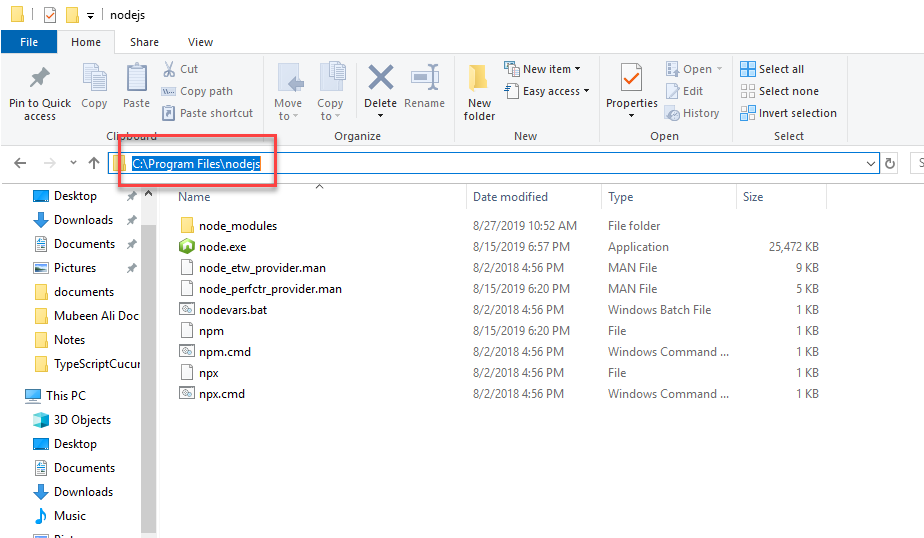
1. Click on the New button under the **system variables** and enter the following values

**Variable name** as **NODE\_HOME**

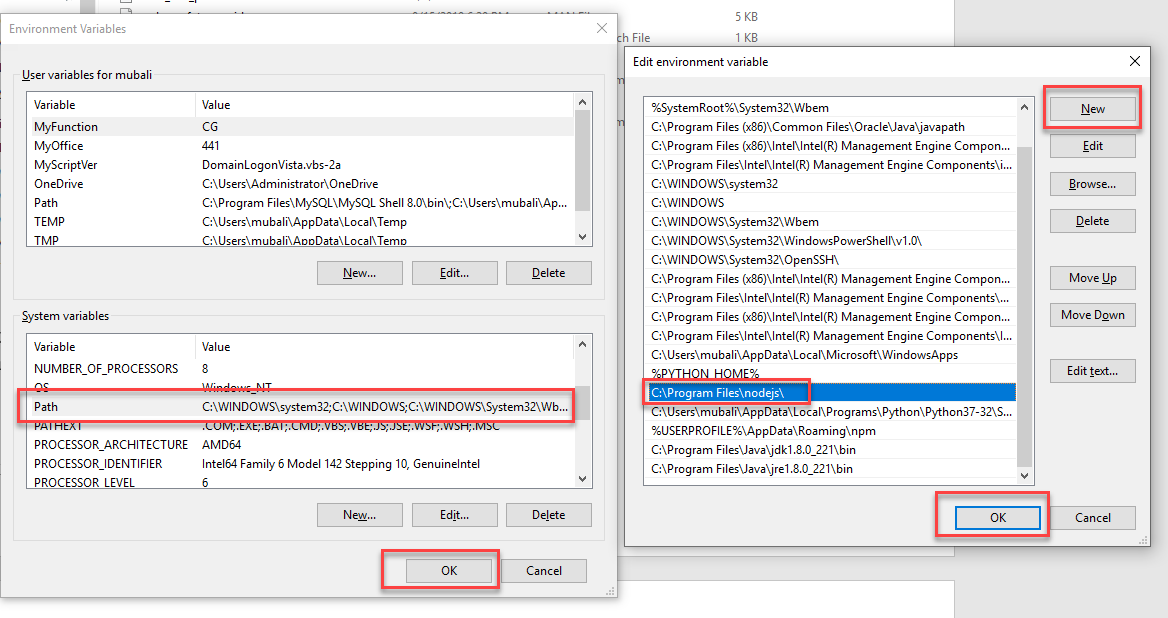
**Variable value** as **C:\Program Files\nodejs**

Note: Value is the path where nodejs is installed.





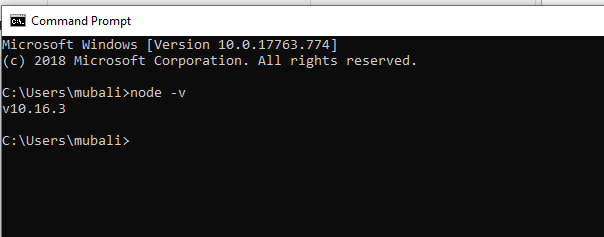
1. Select and double click on the **Path** system variable
2. Click on the Edit button
3. Append your local nodejs path variable value: “**C:\Program Files\nodejs\”** and Click Ok



**Verify if Node.js installation is successful**

1. Open Command Prompt by searching ‘cmd’ in Windows search field
2. Type **“node -v”** in the terminal and hit Enter
3. It should display the version of node.js installed in the system

Note: If Command Prompt is already open, close and open again to follow the above 3 steps



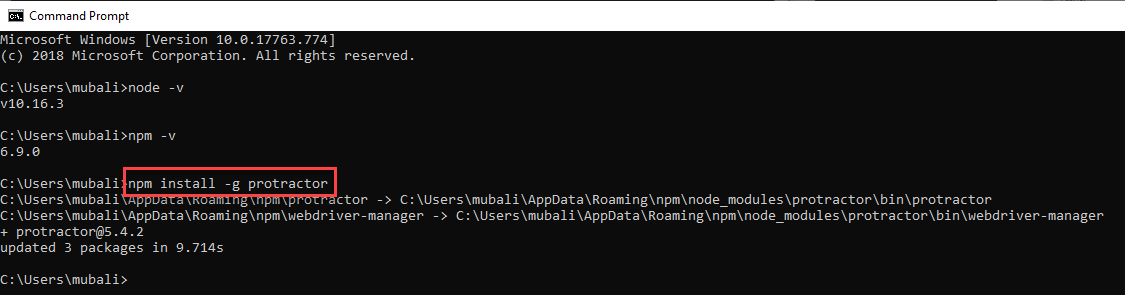
## Protractor Installation

### Installation Steps & Setup

1. Open Command Prompt by searching ‘cmd’ in Windows search field
2. Type “**npm install -g protractor”** and hit Enter

Note: npm is node package manager which helps in installing the protractor package within Node.js

-g stands for global installation of a package. Global installation means across the system and local installation is installing within a specific path

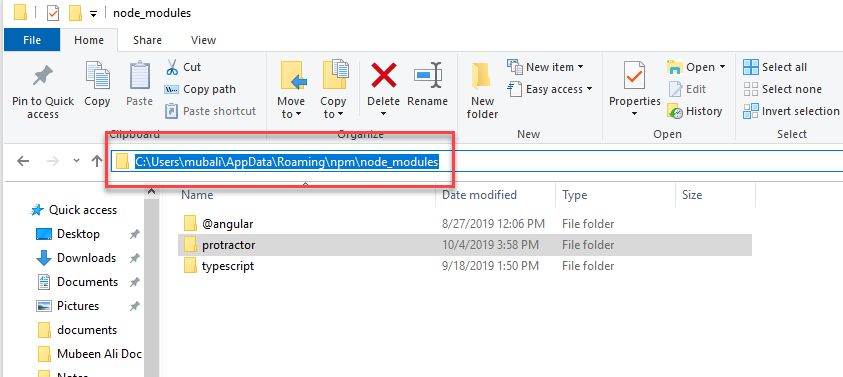


**Verify if Node.js installation is successfully**

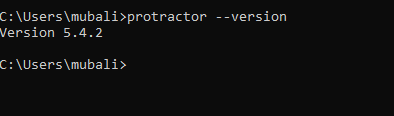
1. Go to following path and verify if Protractor folder is available

C:\Users\**mubali**\AppData\Roaming\npm\node\_modules

Note: remember to replace username in the path with your username



1. Open Command Prompt by searching ‘cmd’ in Windows search field
2. Type “**protractor –version”** and hit Enter

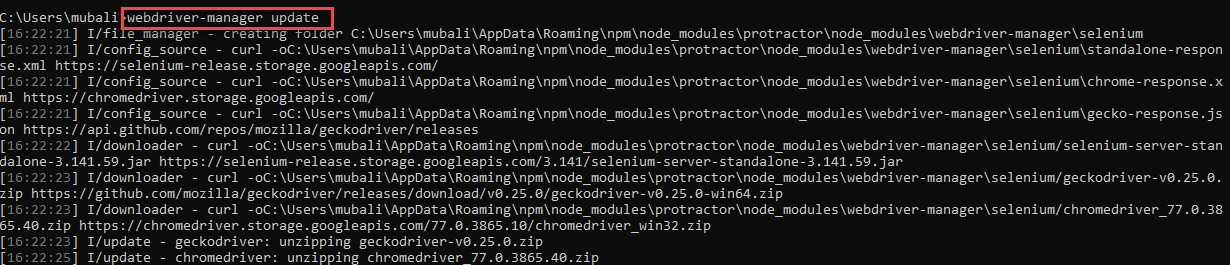


1. If version number is displayed as a result in command prompt shows the successful installation of Protractor and WebDriver-manager

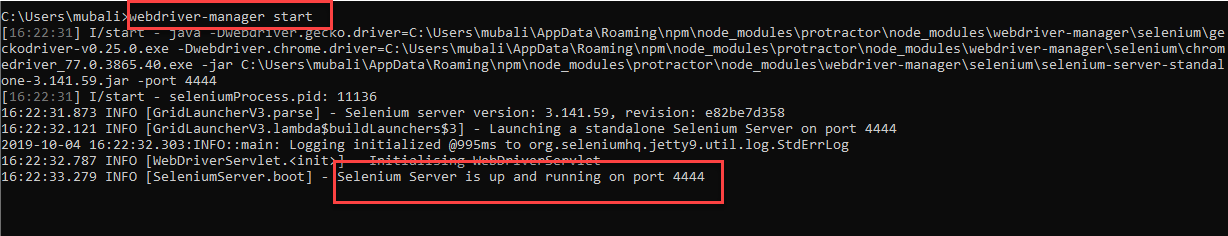
## WebDriver Update and Start

### Installation Steps & Setup

1. WebDriver-Manager will be installed along with Protractor.
2. To Update and Start follow below commands in Command Prompt and Hit Enter Key
3. **webdriver-manager update**



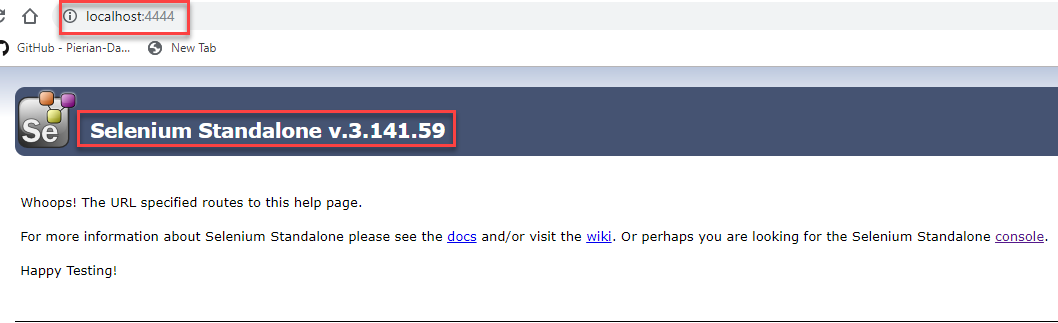
1. **webdriver-manager start**



Note: Selenium WebDriver Server is up and running on port 4444

**Verify if Selenium Server is up and running**

1. Verify if Selenium Server is up and running by copying the following command in browser and verify if Selenium Standalone version is displayed

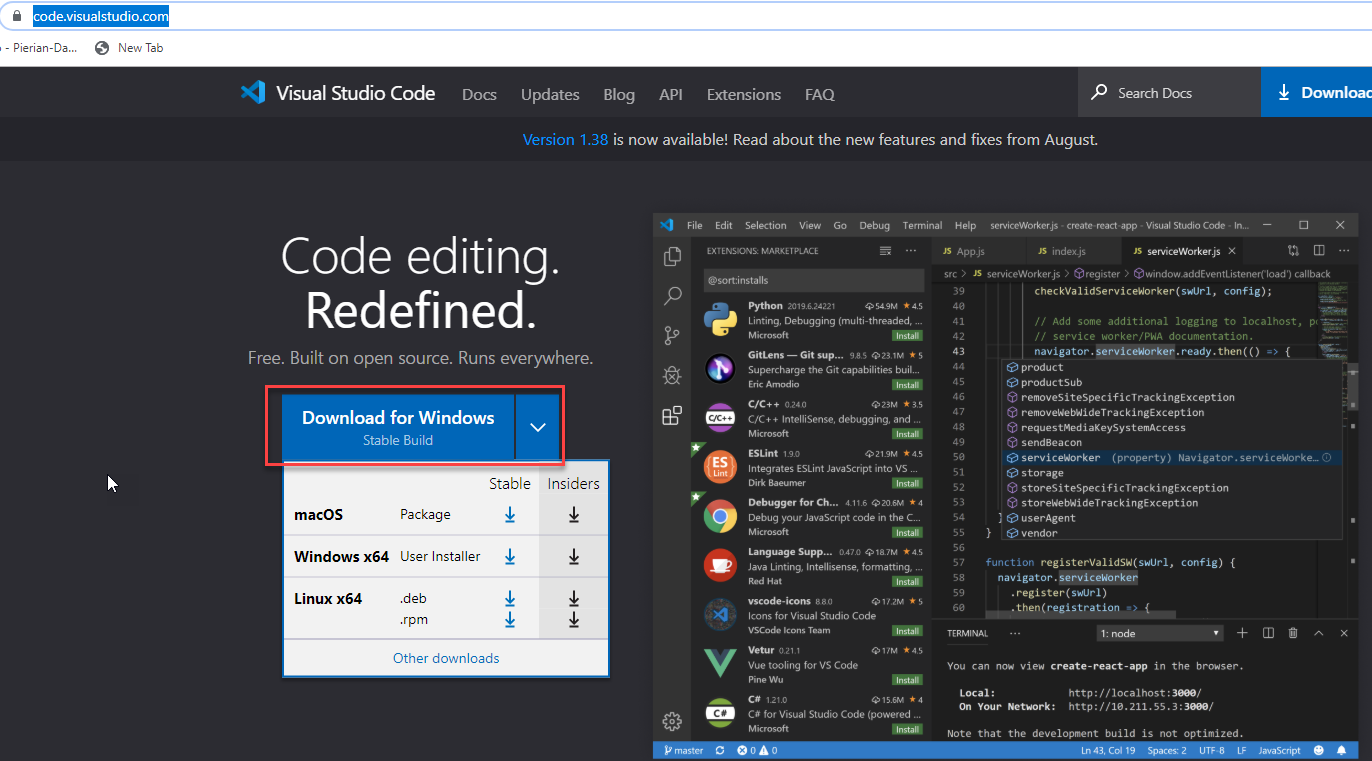


Note: 4444 is the port number, which is subject to change. Take your system port number from the Command prompt after entering webdriver-manager start on the Terminal

## Visual Studio IDE Installation

### Installation Steps & Setup

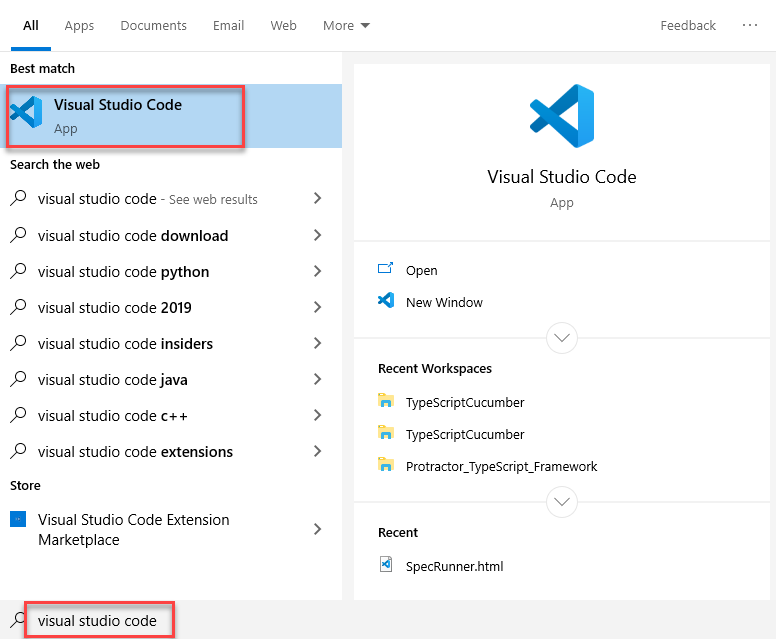
1. Download and Install Visual Studio 2019 from the following link <https://code.visualstudio.com/>
2. Click on Download for Windows



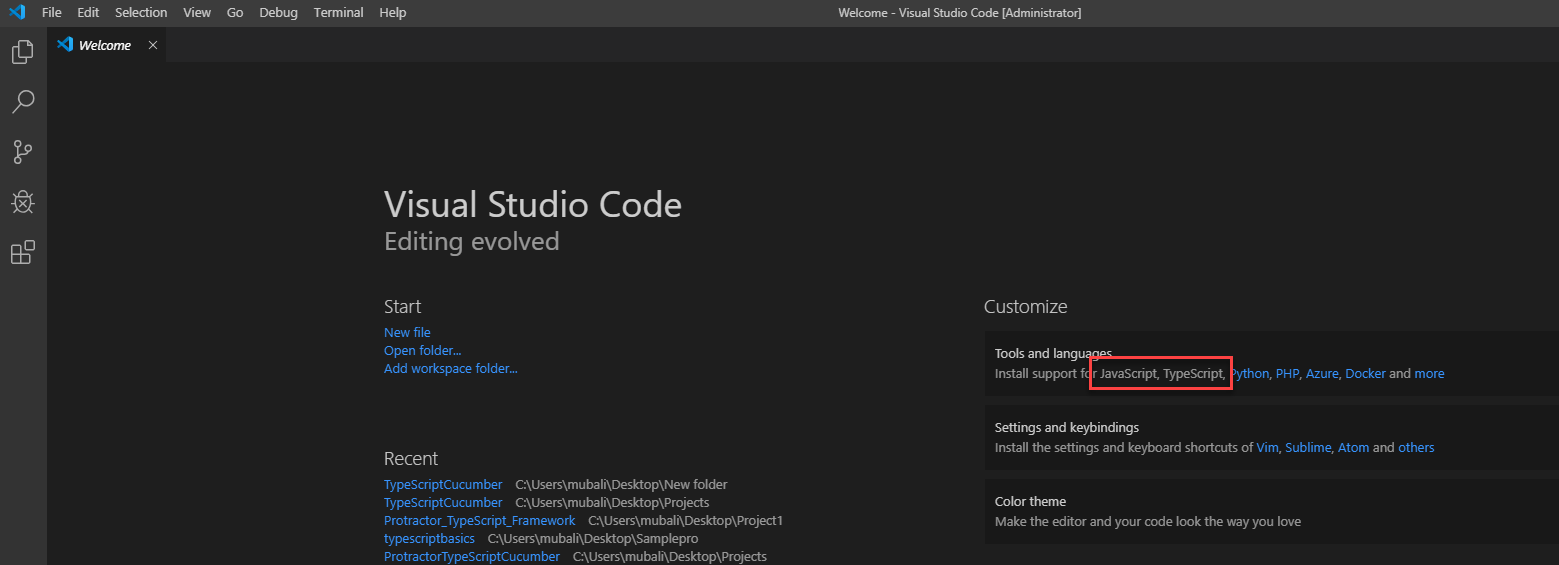
1. Install the downloaded exe file to install Visual Studio Code **“VSCodeUserSetup-x64-1.38.1.exe”** *(current version)*

**Verify if Visual Studio installation is successful**

1. In Windows search bar type Visual Studio Code



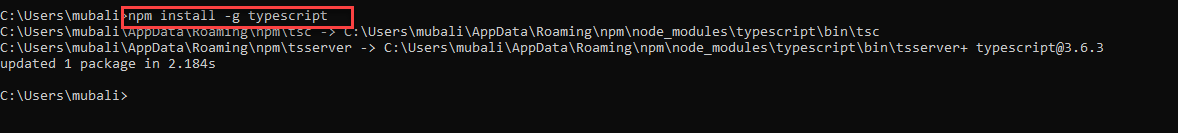
1. Click on the App
2. Click to install support for Javascript and Typescript



## TypeScript Installation

### Installation Steps & Setup

1. Open Command Prompt by searching ‘cmd’ in Windows search field
2. Type **npm install -g typescript** and hit Enter



## Database Setup

### Download MySql Workbench

<https://dev.mysql.com/downloads/workbench/>

### Install the MySql Workbench on the hub /your local machine.

### Once Installed open the MySql Workbench from your program files.

## GIT Installation

### Installation Steps and Setup

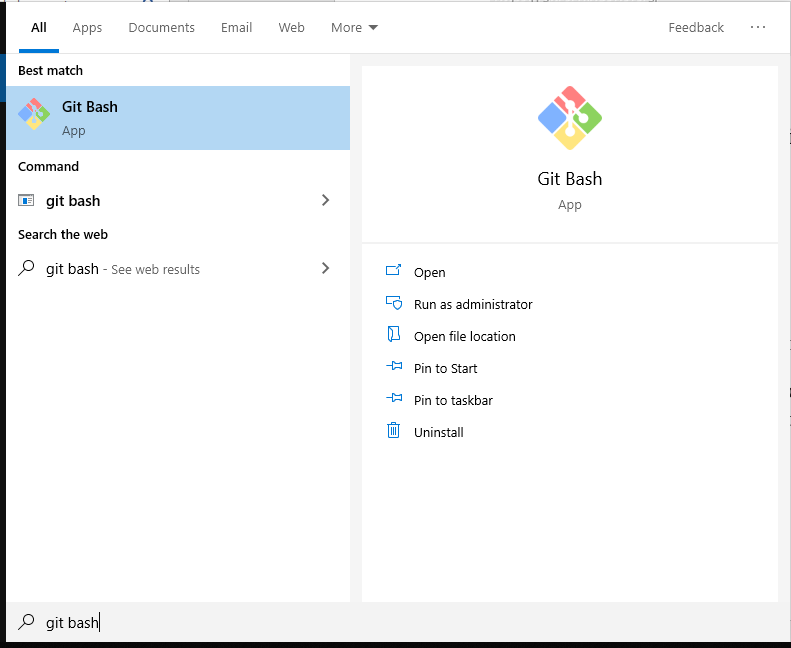
1. Navigate to following link <https://git-scm.com/download/win>
2. Download GIT for Windows
3. Execute the downloaded .exe file **Git-2.23.0-64-bit** *(current version)*
4. Click Next with all default values

Note: Do not change any default values



**Verify if GIT is installation is successful**

Go to Windows search options and type ‘GIT Bash’ and ensure the following icon in below screenshot is available in your system



# User Guide

## Export the project to VS Code IDE

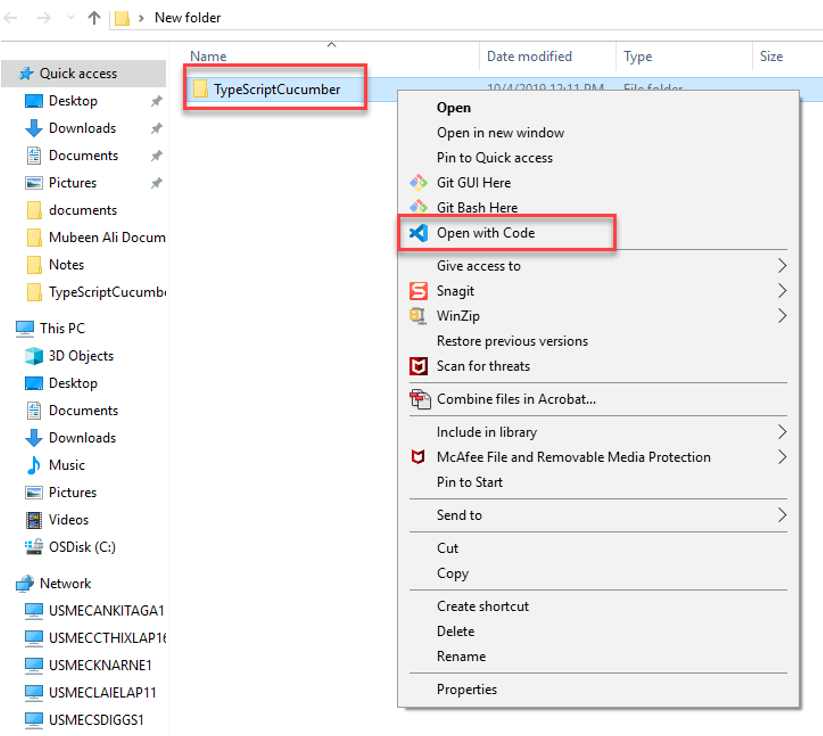
1. Download the project –

From GIT Bash

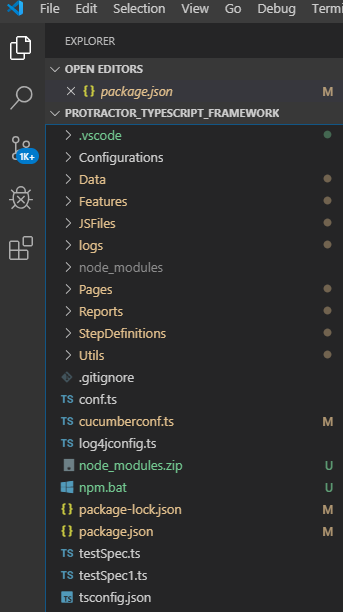
git clone ssh://tcoegit@10.11.21.27/git/tcoe-automationtest-fws.git

* User Name: **tcoegit**
* Password: **tcoegit#1234**

1. Right click to Open with Code as highlighted in below screenshot



## Understanding of Framework Folder Structure

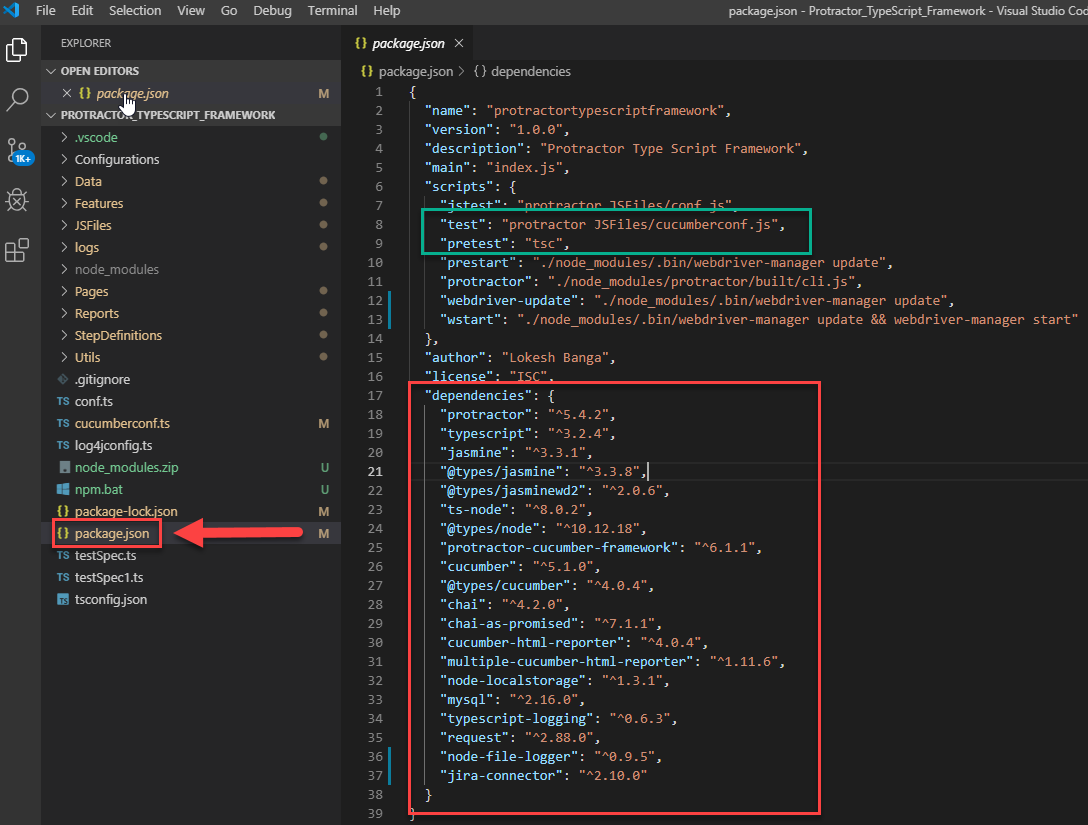


## Understanding the Framework Functionality

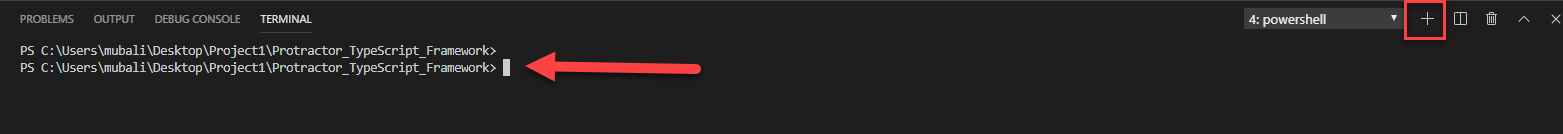
### Package.json

This file has all the dependencies/components with versions that should be installed before using the framework

Highlighted box in Green are the base files that needs to be run to execute the test case



Open the terminal in VS Code by clicking on + icon highlighted below and hit Enter



* + - 1. Type **"tsc”** and hit Enter.

This will convert all the codes written in Typescript to Javascript as Protractor/WebDriver can understand only Javascript code

 "test": "protractor JSFiles/cucumberconf.js",

"pretest": "tsc",

* + - 1. Type **“protractor JSFiles/cucumberconf.js”** and hit Enter

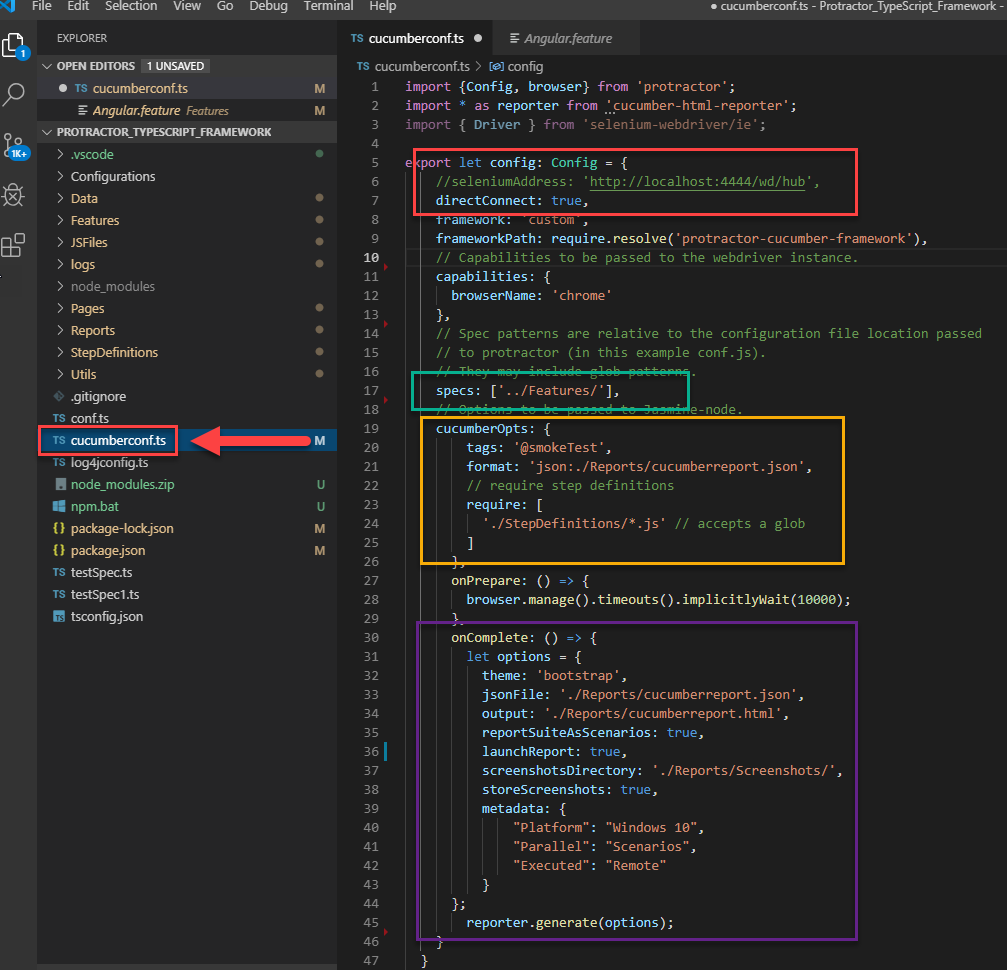
This will run all the test scripts path provided in cucumberconf.js file



### Cucumberconf.ts

1. After all the dependencies are added in Package.json file, the next main file that controls the complete test suite is cucumberconf.ts file.
2. All the file paths that should to be executed are provided in configuration file
3. You might have noticed that while trying to invoke the scripts with following commands

Protract JSFiles/ccucumberconfig.js. the .js file extension is provided and not the .ts extension as with the command ‘tsc’ all codes written in Typescript will get converted to complex Javascript code. However, the “file name” will not be renamed to .js by itself.



1. **DirectConnect** - Box highlighted in Red in above screenshot signifies that ‘webdriver-manager start’ command is not required to be executed separately. Instead the directConnect command will ensure that Selenium webdriver is up and running
2. **Specs: [‘../Features/’]** – Box highlighted in Green in above screenshot, all the cucumber feature file will be executed which is mentioned in this path.

Please be noted that Specs path has 2 dots. This signifies that cucumberconfig.js file which is inside JSFiles will not only look for Feature file within JSFile but also at the Parent level.

More details regarding JSFile will be shared in sub section [**1.3.10**](#_JSFiles)

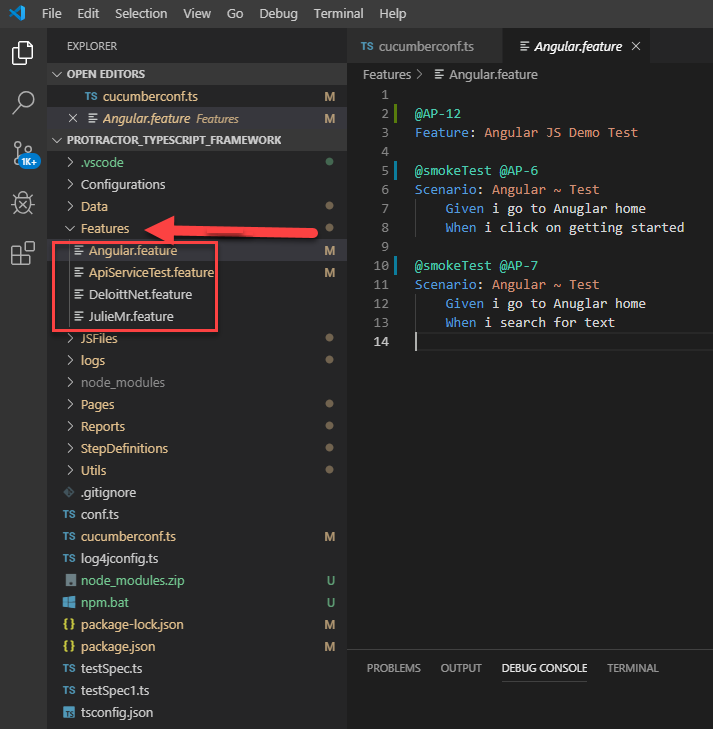
1. **cucumberOpts** – Yellow box highlighted in above screenshot,
   1. tags: ‘@SmokeTest’ – Denotes the features in Feature file with tag @SmokeTest will only be executed
   2. format – Format in cucumber options is the file format of the report should be generated in JSON, HTML etc.
   3. require: ['./StepDefinitions/\*.js'] – stepdefinition.ts file path associated with the feature file that should to be executed will be provided here along with hooks.ts and timeouts.ts

More details on hooks.ts and timeouts.ts will be provided in upcoming sub section [**1.3.5**](#_Hooks.ts)and[**1.3.6**](#_Timeouts.ts)

1. **onComplete** – Provides the path where the report needs to be generated after the test cases are executed

### Feature Files

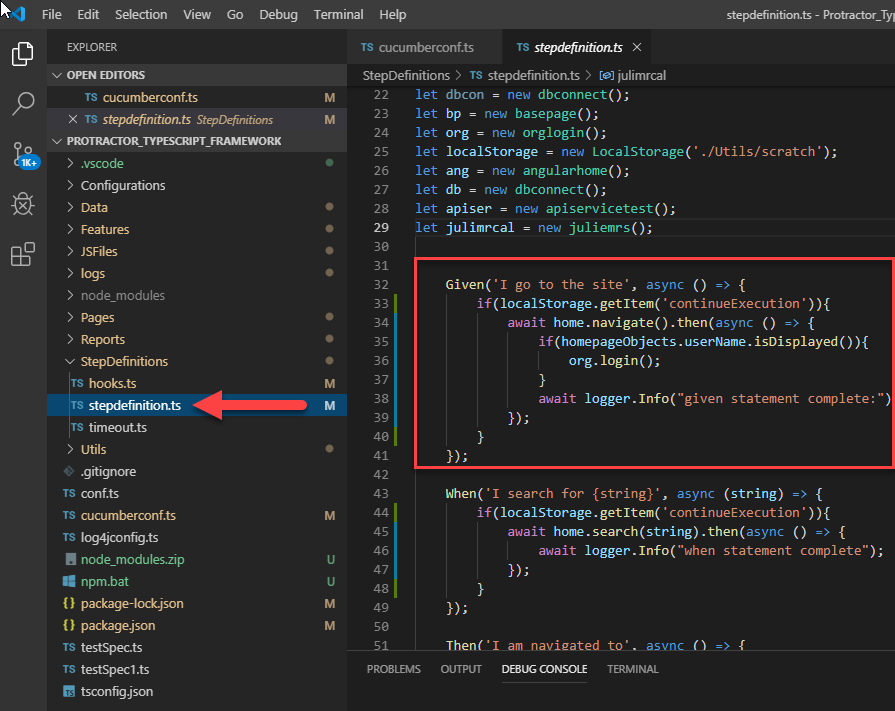
In BDD Cucumber testing, feature files are the business requirement test cases written in plain English. Each feature will have test steps written with gherkin keywords like Given, When, And & Then



Note: Feature file path is given in Cucumberconfig.ts file as mentioned in section [**1.3.2**](#_Cucumberconf.ts)

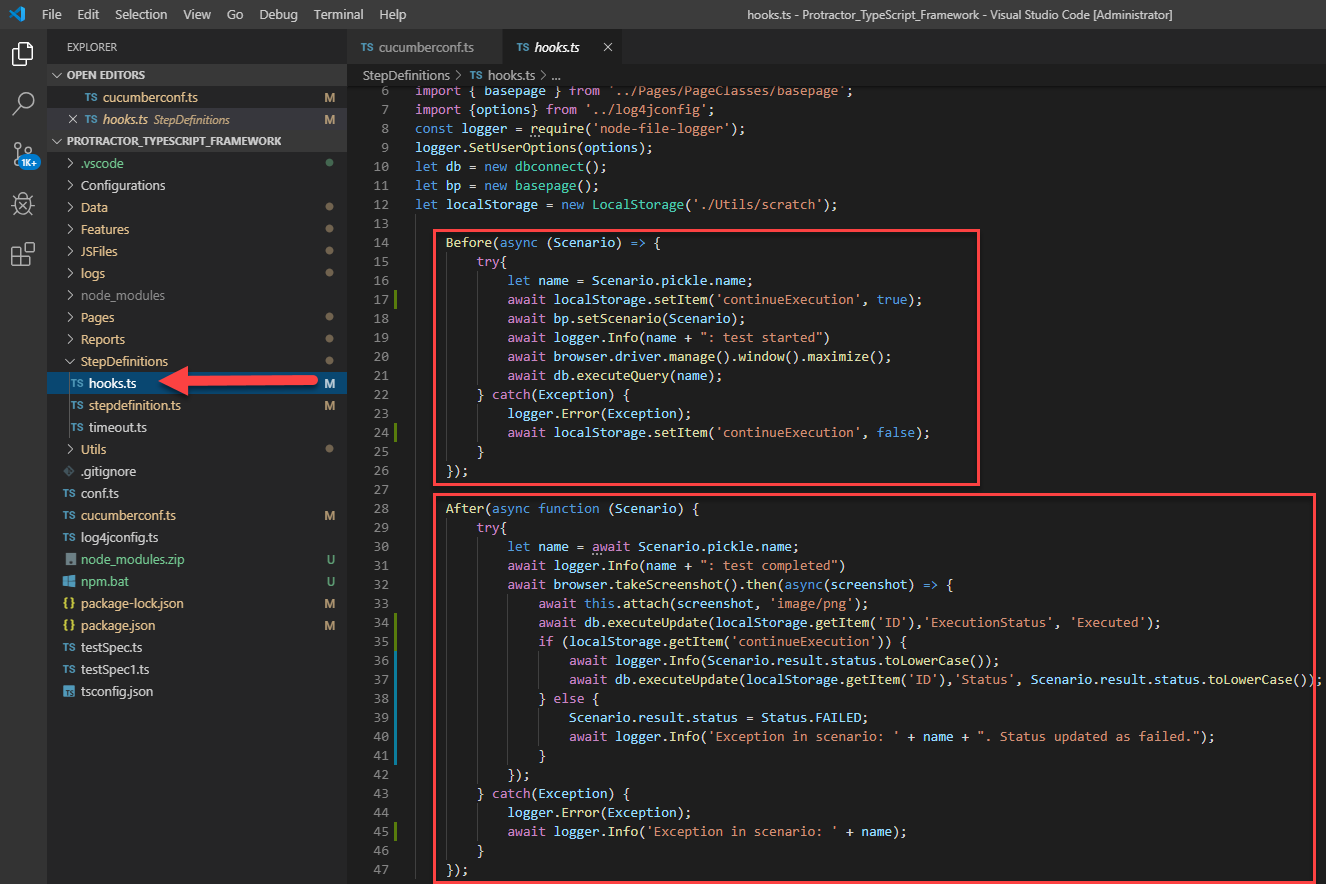
### Stepdefinition.ts

Stepdefinition.ts is where the actual code to test is written based on the feature file.



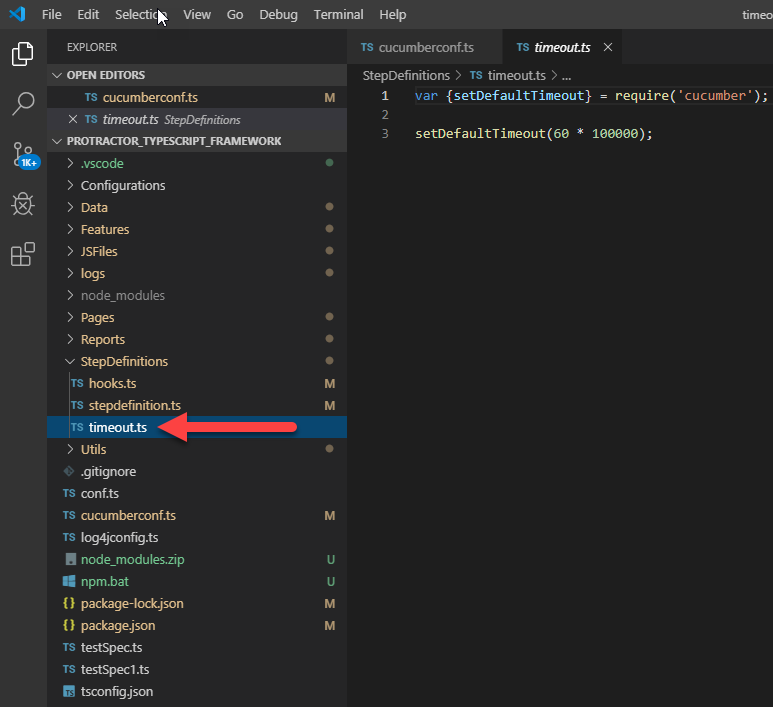
### Hooks.ts

Hooks are like @BeforeTest and @AfterTest in TestNG. As shown in below screenshot, it has a Before and After block which will be executed accordingly to the test code in Step definition file



### Timeouts.ts

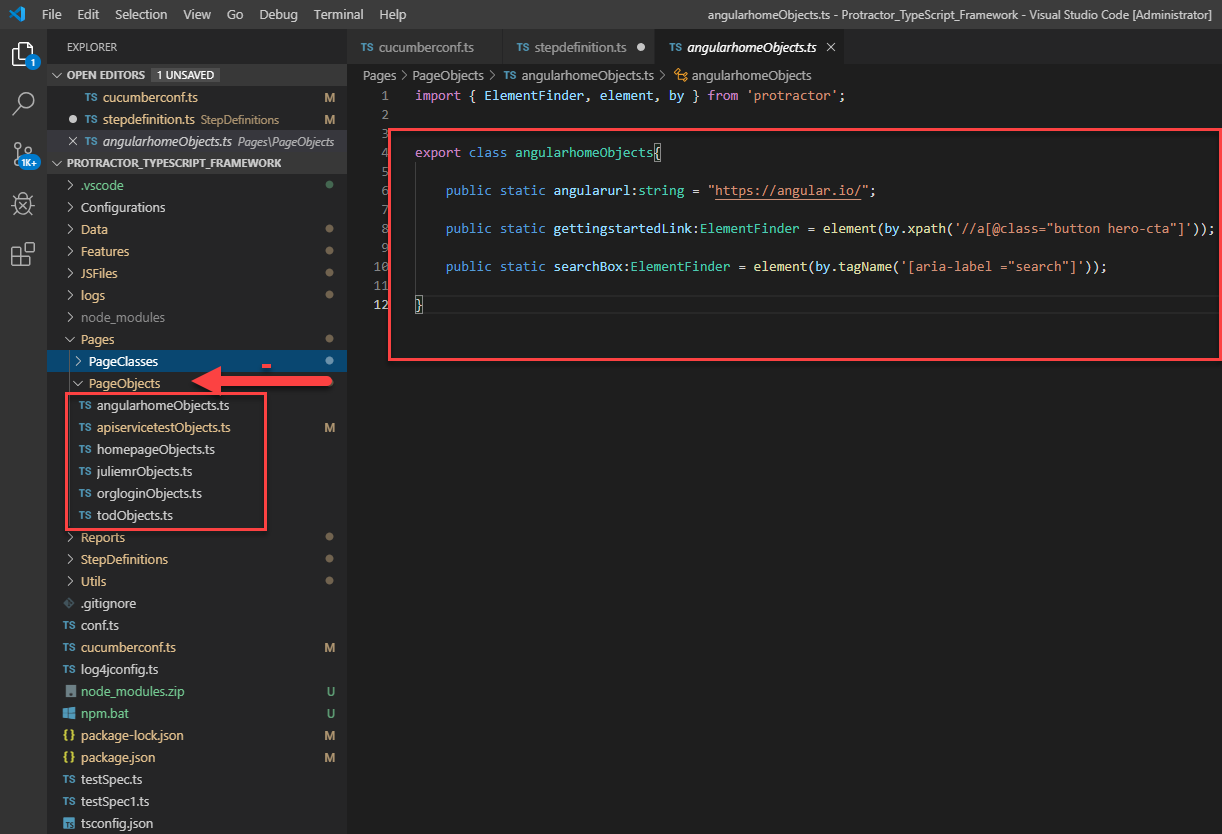
Timeout.ts file has the default timeout set to 60 seconds



### PageObjects

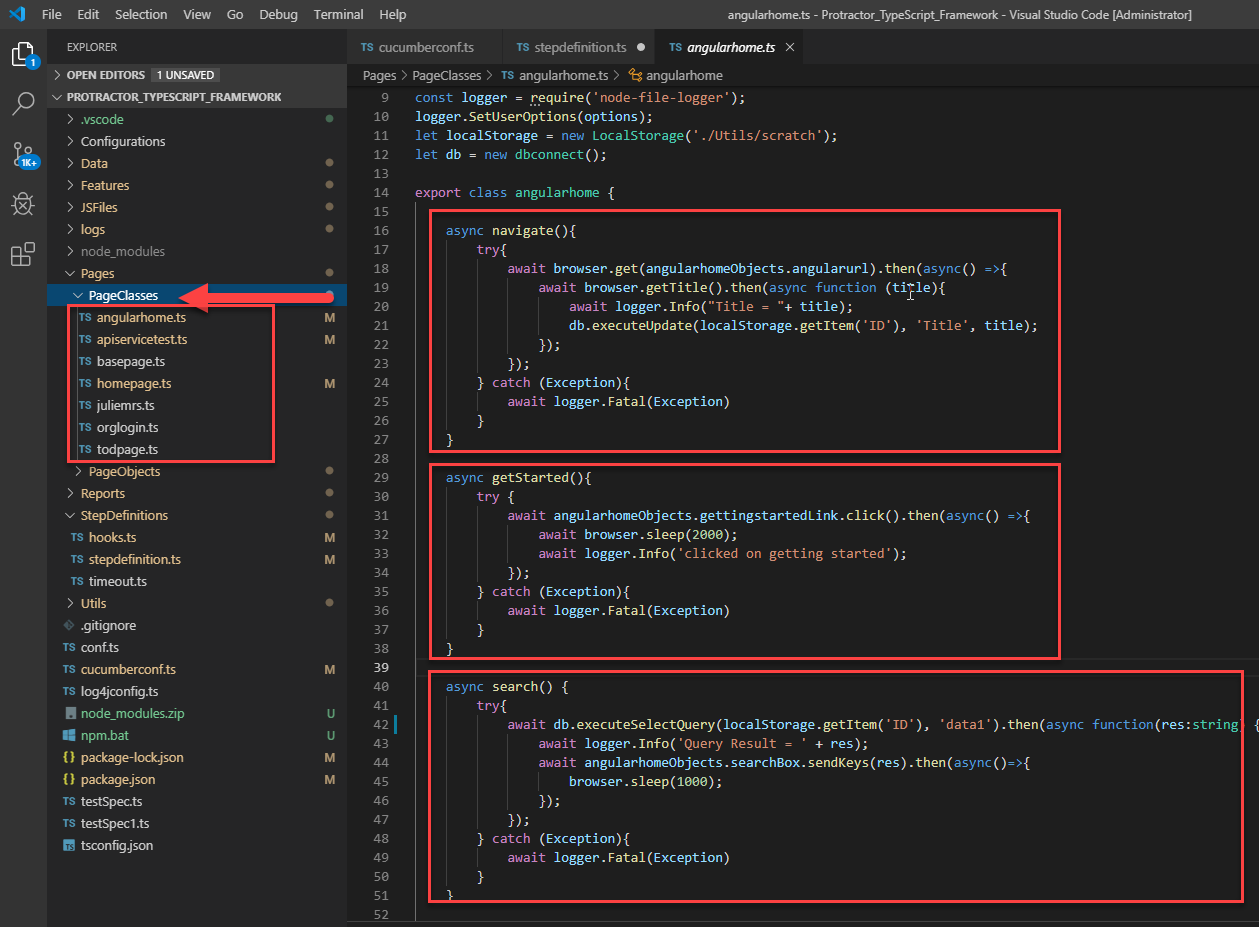
Page Object Model methodology is followed in this framework, where the page objects otherwise known as web elements for all the pages are initialized in this folder.

The reason for following this approach is to make the maintenance easy, in future if there is any change in object. It need not be updated in the code directly rather update in the page object files in this folder



### PageClasses

All methods are defined in Page Classes for each Application to test. Page Classes folder acts as library with all the methods which can be reused. Page Objects mentioned above is imported to this class. In Stepdefinition.ts objects for Page Class names are created and all corresponding Methods and Variables are initialized.



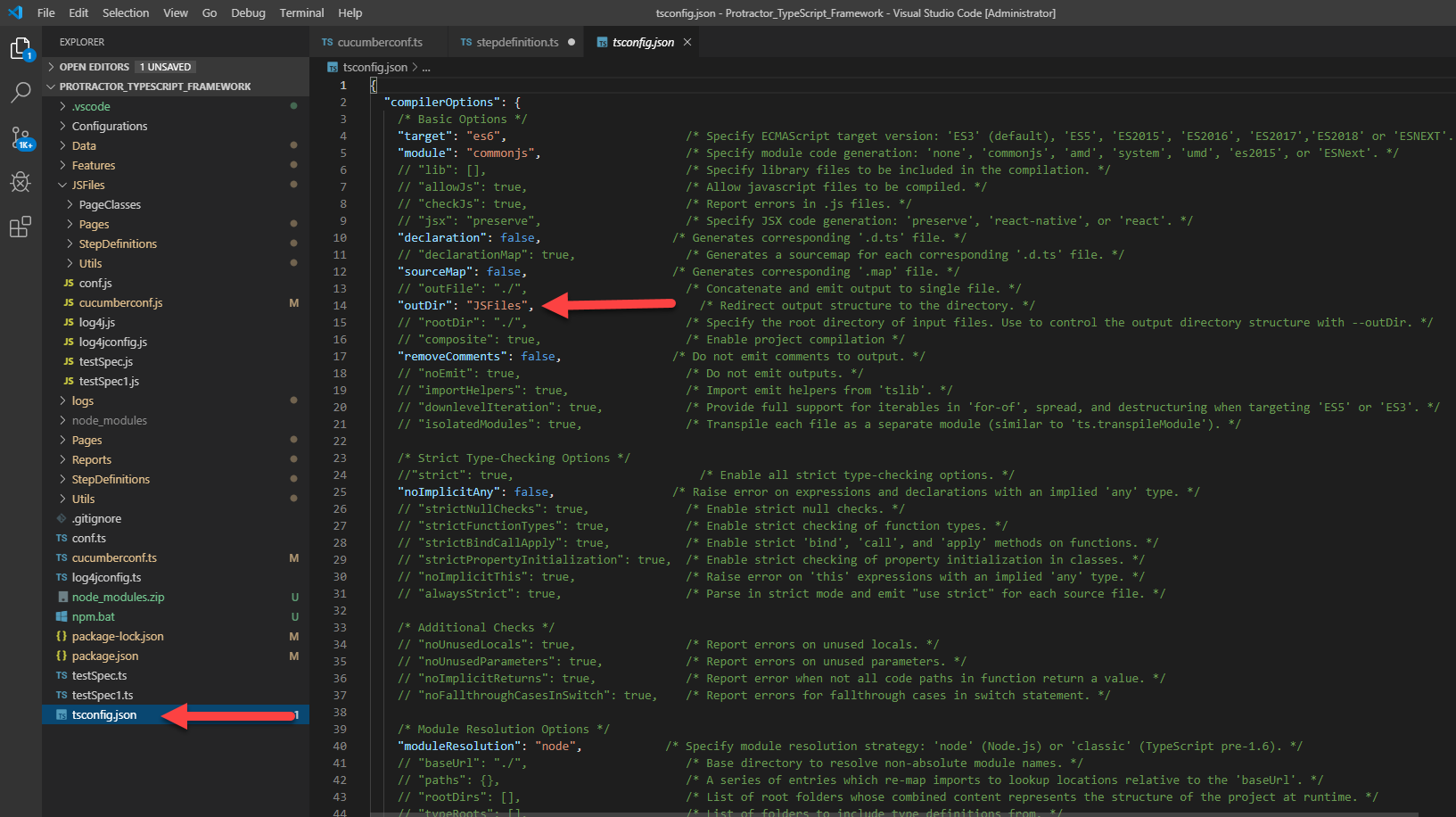
### Tsconfig.ts

Tsconfig.ts file is created by the Automation Architect with the command “**tsc –init**”

Note: No action required from the user, users can read through the commented section in green to understand how it functions. Type script configuration file majorly provides the Java Script Folder Path (JSFile) where the converted (Typescript to JavaScript) code is stored.

As a pre-requisite, when tester enters the command ‘**tsc’** to convert all Typescript files to JavaScript. All the JavaScript files will be stored in JSFiles folder mentioned in **Tsconfig**.ts file

Note: Remember protractor can only understand JavaScript. So, it is mandatory to execute **“tsc”** command before execution, so that all files in .ts files converted to .js code files

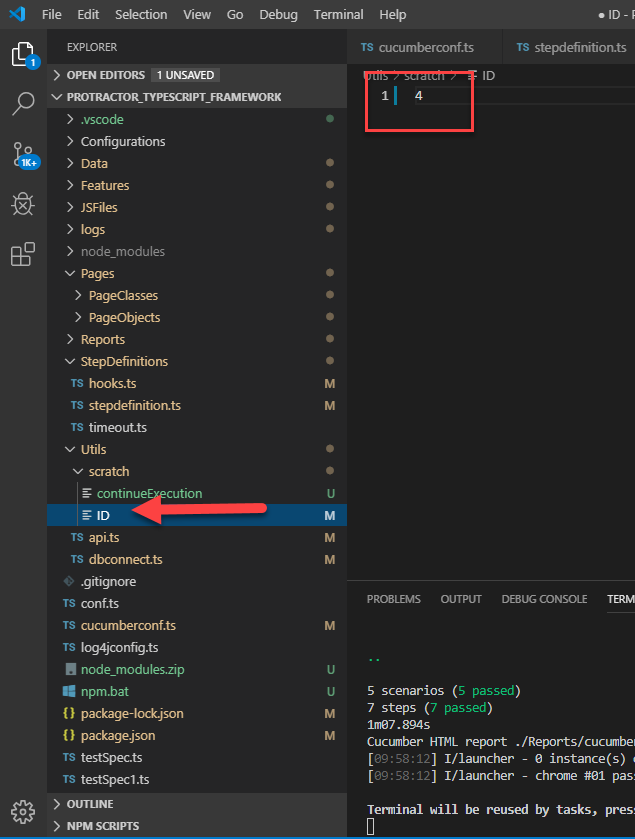


### JSFiles Folder

JSFile has all the files converted into Javascript (.js extension) files generated from typescript (.ts) based on the command **‘tsc’**

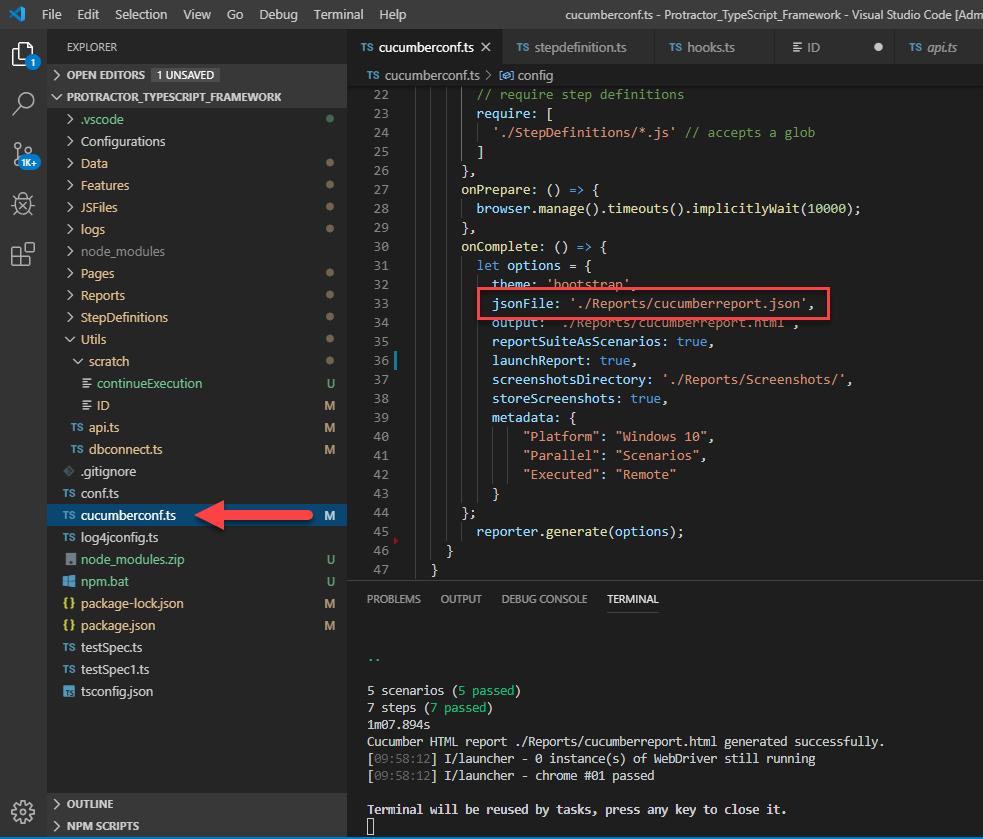
### Utils

In database table testdata, each scenario is provided an id. In Utils/Scratch/ID file, the #ID number is provided as shown in below picture. Following test case will be run accordingly

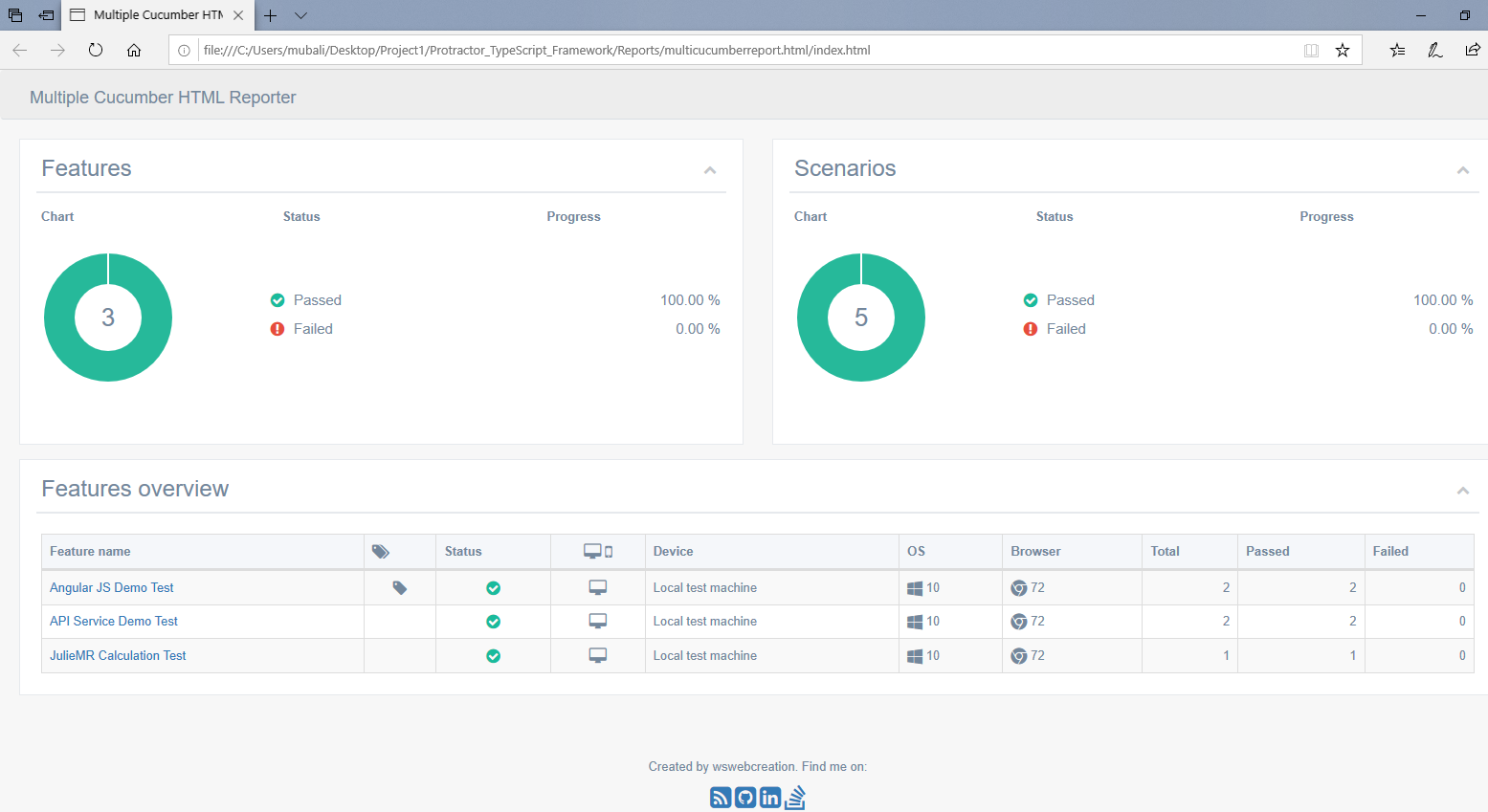


### Cucumber HTML Reports

Based on the file path provided in Cucumberconf.ts highlighted in below screenshot. Report will be generated with status Pass/Fail/Skipped for all targeted scripts



Go to ./Reports path find the html report generated like below



# Run Book

## Framework Flow

**Test Execution**

**Cucumberconf.ts**

**Database setup**

**Package.json**

Run the command in VS Terminal

>> npm install  
  
All dependencies will be downloaded and installed in the system

Associated file paths are provided in this file. (Configuration file is the heart of execution)

Run the following commands in VS Terminal

>> tsc

>> Protractor JSFiles/cucumberconf.js

Execute the database queries

1. Run INSERT Query (one-time activity)

2. Run UPDATE Query

3. In Utils/ID provide the scenario to execute

**Hooks.ts**

**Stepdefinitions.ts**

**Feature File**

**Hooks.ts**

After hooks are executed

* Database updated
* Close db Connection
* Logs are updated

Executes the feature files provided in cucumberconf.js

* Initializes Page Objects and Page Class
* Test Case Execution
* Screenshots are taken

Before hooks are executed

* Db connections
* Logs

**Cucumber Report Generation**

## Pre-requisite Database Queries to Run

Open MySQL and run the following commands

### Creates following database with below command:

Create DATABASE protractor\_framework

### Creates following table with below command:

CREATE TABLE `protractor\_framework`.`testdata`

(

`ID` INT PRIMARY KEY NOT NULL auto\_increment,

`Scenario` VARCHAR(200) NOT NULL,

`Test` VARCHAR(200) NOT NULL,

`ExecutionStatus` VARCHAR(200) NOT NULL,

`Status` VARCHAR(200) NULL,

`Title` VARCHAR(200) NULL,

`data1` VARCHAR(200) NULL,

`data2` VARCHAR(200) NULL,

`ResponseCode` VARCHAR(200) NULL

)

### Provide the username and password to the database

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'root';

### Insert values to the table testdata

* INSERT INTO `protractor\_framework`.`testdata` (`ID`, `Scenario`, `Test`, `ExecutionStatus`) VALUES ('1', 'Angular', 'Test', 'NotExecuted');
* INSERT INTO `protractor\_framework`.`testdata` (`ID`, `Scenario`, `Test`, `ExecutionStatus`) VALUES ('2', 'Service', 'Test', 'NotExecuted');
* INSERT INTO `protractor\_framework`.`testdata` (`ID`, `Scenario`, `Test`, `ExecutionStatus`, `data1`) VALUES ('3', 'Angular', 'Test', 'NotExecuted', 'Tutorial');
* INSERT INTO `protractor\_framework`.`testdata` (`ID`, `Scenario`, `Test`, `ExecutionStatus`) VALUES ('4', 'Service', 'Test', 'NotExecuted');
* INSERT INTO `protractor\_framework`.`testdata` (`ID`, `Scenario`, `Test`, `ExecutionStatus`, `data1`, `data2`) VALUES ('5', 'Calculation', 'Test', 'NotExecuted', '45', '54');

Each scenario is linked to an ID in the database, based on the ID provided in Util/Scratch/ID file in framework. Following scenario will be executed

## Execution – Test Run

Cucumberconf.ts – Ensure the path of Feature files to execute along with Step definition path is provided in this file

Go to Visual Studio Code terminal and execute following commands

**>> tsc**

**>> protractor .JSFiles/cucumberconf.js**

Following commands will convert all typescript code in JavaScript and execute the scenarios to test in feature files.

## Post Execution

Once the test execution is complete, go to mysql and execute the following commands

### Run the query to fetch the end result

* use protractor\_framework
* select \* from testdata

Select query will provide the status of the scenarios that are run

### Clears the data in testdata table

* UPDATE `protractor\_framework`.`testdata` SET `ExecutionStatus` = 'NotExecuted', `Status` = '', `Title` = '', `ResponseCode` = '' WHERE (`ID` = '1');
* UPDATE `protractor\_framework`.`testdata` SET `ExecutionStatus` = 'NotExecuted', `Status` = '', `Title` = '', `ResponseCode` = '' WHERE (`ID` = '2');
* UPDATE `protractor\_framework`.`testdata` SET `ExecutionStatus` = 'NotExecuted', `Status` = '', `Title` = '', `ResponseCode` = '' WHERE (`ID` = '3');
* UPDATE `protractor\_framework`.`testdata` SET `ExecutionStatus` = 'NotExecuted', `Status` = '', `Title` = '', `ResponseCode` = '' WHERE (`ID` = '4');
* UPDATE `protractor\_framework`.`testdata` SET `ExecutionStatus` = 'NotEtestdatafilm\_idxecuted', `Status` = '', `Title` = '', `ResponseCode` = '' WHERE (`ID` = '5');

Above Update commands will clear the records in testdata table

# Reference Documents

Please refer to the following documents as well:

1. Protractor Official Website –

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

1. Config.ts Reference Link

<https://github.com/angular/protractor/blob/master/lib/config.ts>

1. Latest Package.json file

<https://github.com/angular/protractor/blob/5.4.1/exampleTypescript/package.json>

1. Hooks

<https://github.com/cucumber/cucumber-js/blob/master/docs/support_files/hooks.md>

1. Cucumber HTML Reports

<https://www.npmjs.com/package/cucumber-html-reporter>

1. Jasmine Assertions

<https://github.com/angular/protractor/blob/5.4.1/exampleTypescript/tsconfig.json>

1. Chai

<https://www.chaijs.com/>

---------------------------------------------------END--------------------------------------------------