**Protractor – Angular Automation Tool**

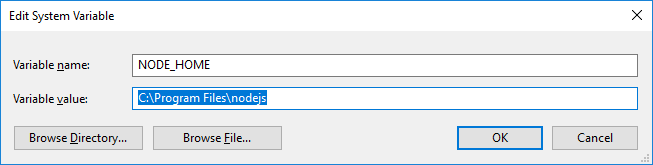
Protractor is an end-to-end test framework for Angular and AngularJS application. Also used to test non-Angular applications.

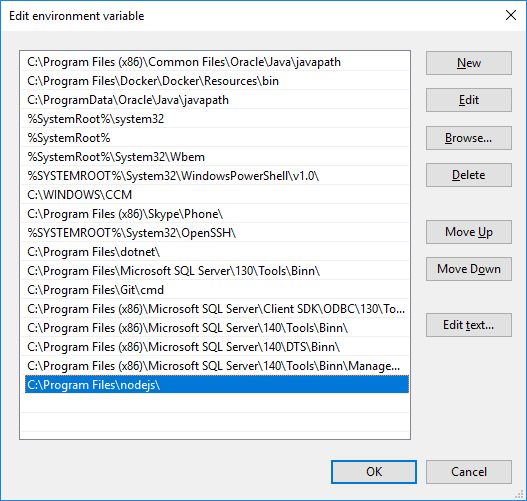
WebDriver + Angular support = Protractor

Protractor is a node.js program built on top of WebDriver JavaScript. So it uses JavaScript as core language to develop tests.

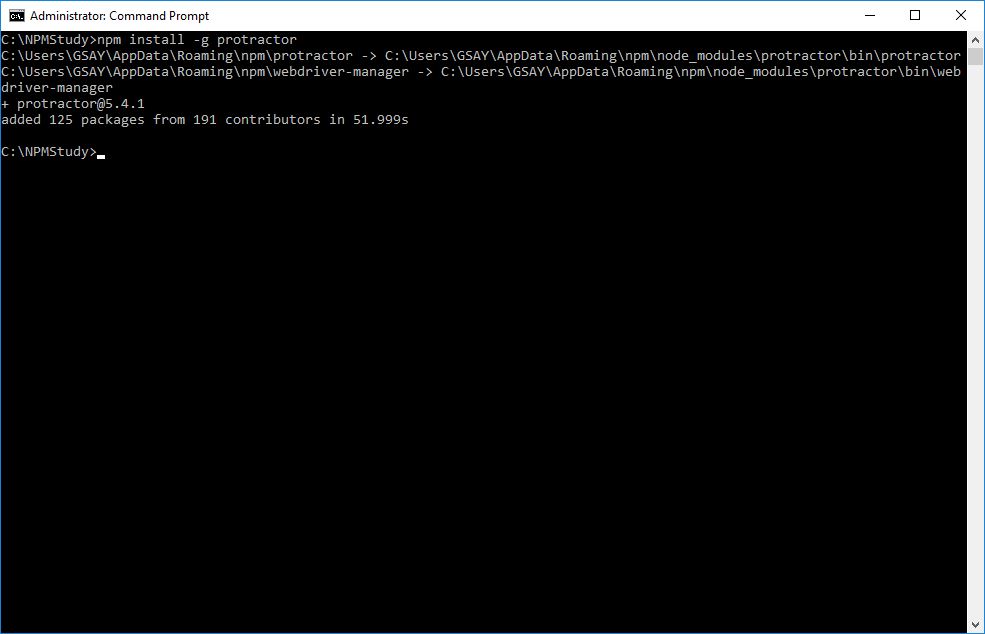
**Protractor Installation**

1. Download Node.js (<https://nodejs.org>) – Download latest stable version.
2. Set Node in system variables





1. Node Package Manager
2. Install Protractor



*C:\Users\GSAY>protractor –version*Version 5.4.1

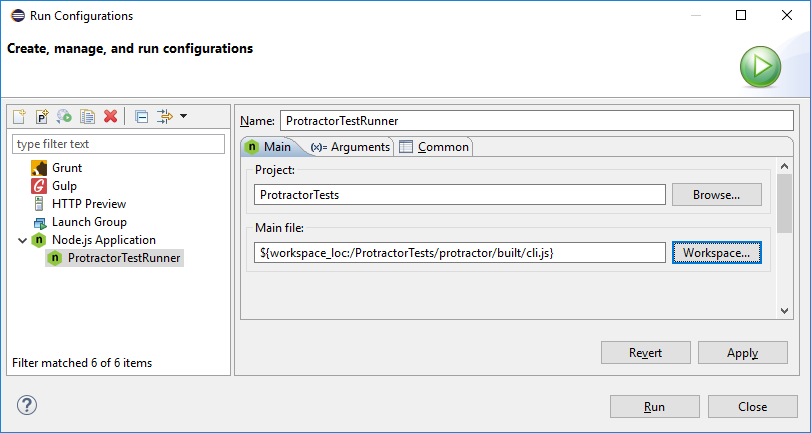
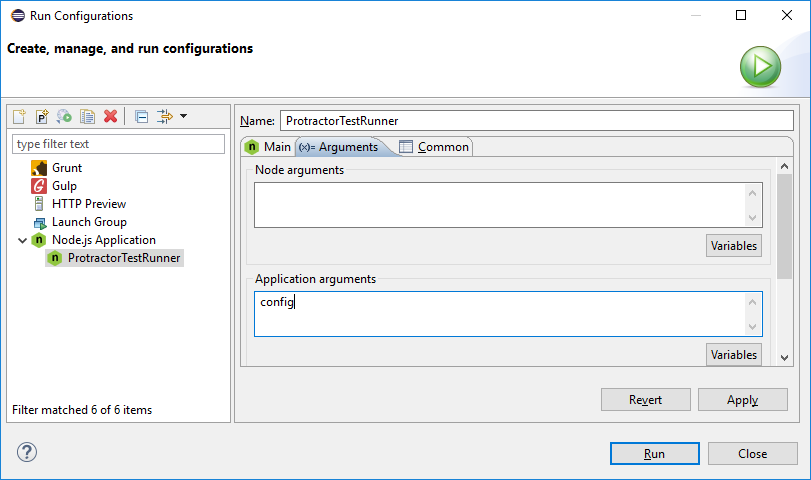
1. WebDriver Manager plugin

*C:\Users\GSAY>webdriver-manager update*

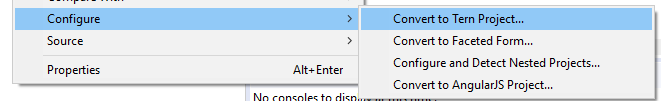
1. Download Eclipse JavaScript Tool
2. Importance of Jasmine framework for Protractor tests - <https://jasmine.github.io/>
3. Understand specs and configuration file
4. Describe and It blocks importance in Protractor test
5. Preparing Protractor Run configuration file
6. Running Protractor first test

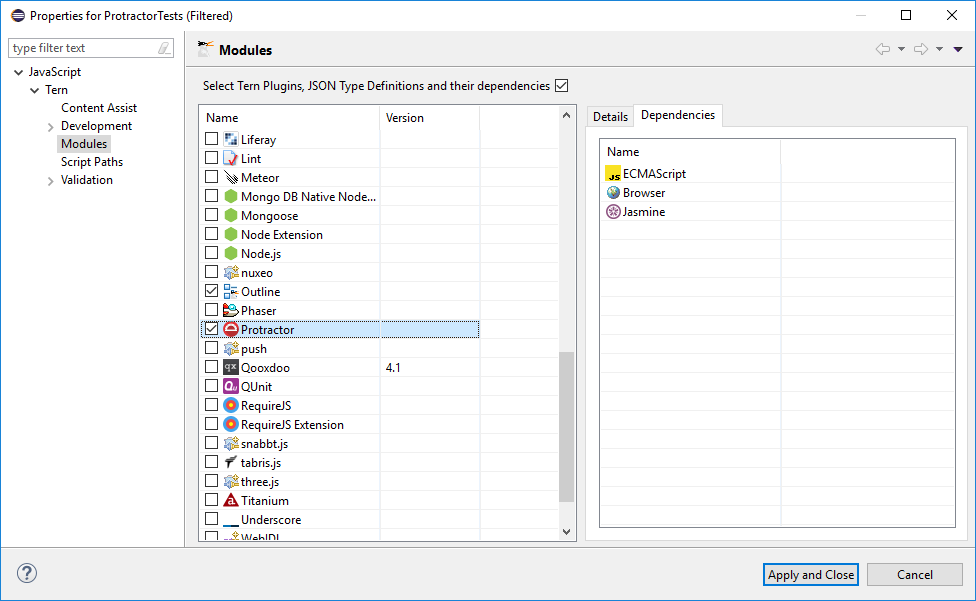
C:\Users\GSAY\AppData\Roaming\npm\node\_modules\protractor\built\cli.js

Copy Protractor folder into your project and add configuration as below:

* Right click the project and select Run Configuration. Select Node.js application
* 
* 

1. Install Angular JS plugins – Intern project (Eclipse Market place >> AngularJS Eclipse)





JavaScript is Asynchronous. Every JavaScript step return Promise

**Promise resembles state of your steps**:

1. Pending 2. Resolved 3. Rejected

**Synchronous**: You will move to next step only after promise for current step is either resolved or rejected.

**Asynchronous**: JavaScript moves to next step even if promise is pending

90% (all of the actions performed on browser) of Protractor API will not move to next step until promise is resolved.

If you want to retrieve anything from browser, then that particular functionality related methods do not have support of promise resolving. (e.g. getText(), getTitle(), getAttribute() etc.)

Links to read more on promises:

* <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 Basics:**

1. Printing in the Console

console.log(“Hello World!”);

1. JS Data Types

var a = 4; // dynamic type  
var b = “hello”;  
console.log(a);  
console.log(b);

1. JS Conditional Statement

If (var1==var2)  
{  
 console.log(“1”);

} else if

{  
 console.log(“2”);  
} else  
{  
 console.log(“3”);  
}

1. JS Loops
2. JS Operators
3. JS Functions
4. JS Arrays
5. JS Strings
6. Importance of Objects in JS
7. Ways of creating JS objects

Demo Project Walkthrough:

**Protractor Framework building from Scratch – 1**

JS is an object-based language. JS is template based not class based. We don’t create class to get the object, but we direct create objects.

A JS object is an entity having state and behaviour.

**Page object Mechanism for Protractor tests:**

**Git:**

<https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html>

1. Create Repository in GitHub
2. >git config --global user.name "Sanjay"

>git config --global user.email [sanjay.adav@clarksons.com](mailto:sanjay.adav@clarksons.com)

3. How to push code to remote repository

Go to the folder

>git init

*Add files to stash (staging)*

>git add \*

>git status

>git commit –m “Commit Message”

>git remote add origin <server>

Here server like <https://github.com/sanjayadav/myGit.git>

>git push –u origin master

*How to clone repository?*

>git clone <server>

>git status

>git add \*

Commit and push

*How to get latest code?*

>git pull origin master

*Branching*

>git checkout –b develop (create new branch and checkout to new branch)

>git branch

>git checkout develop (checkout to new branch)

*How to merge develop branch into Master branch?*

*Switch to master branch first*

>git checkout master

>git pull origin master

>git merge develop