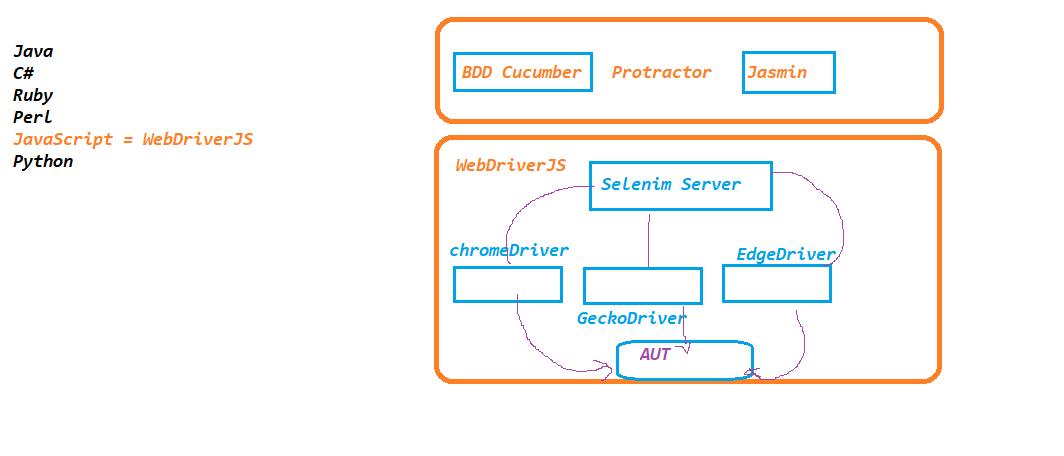
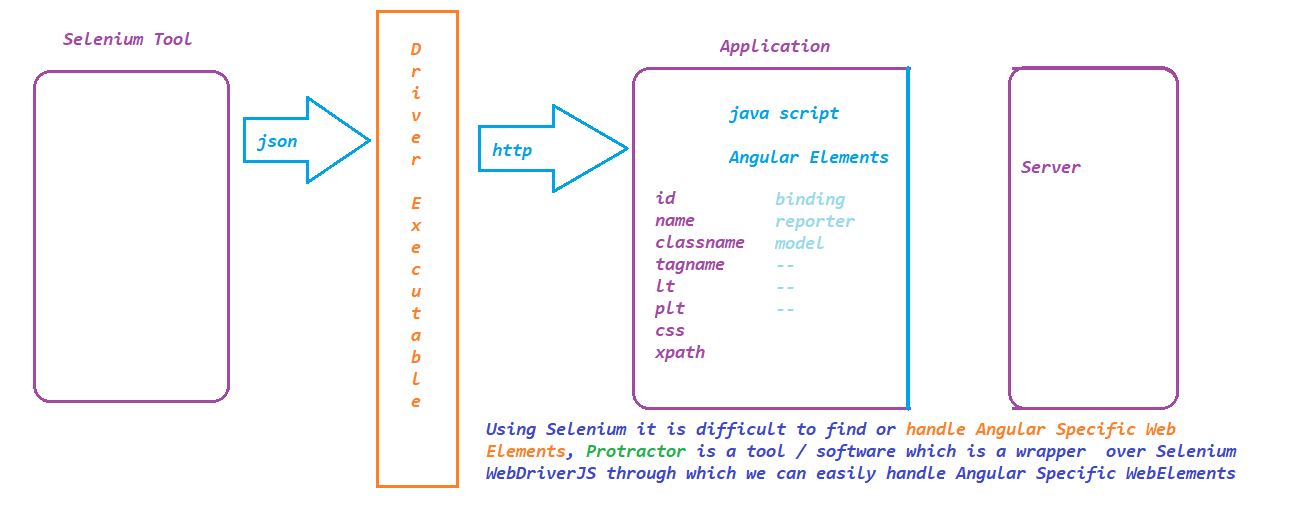
**Protractor - Architecture**

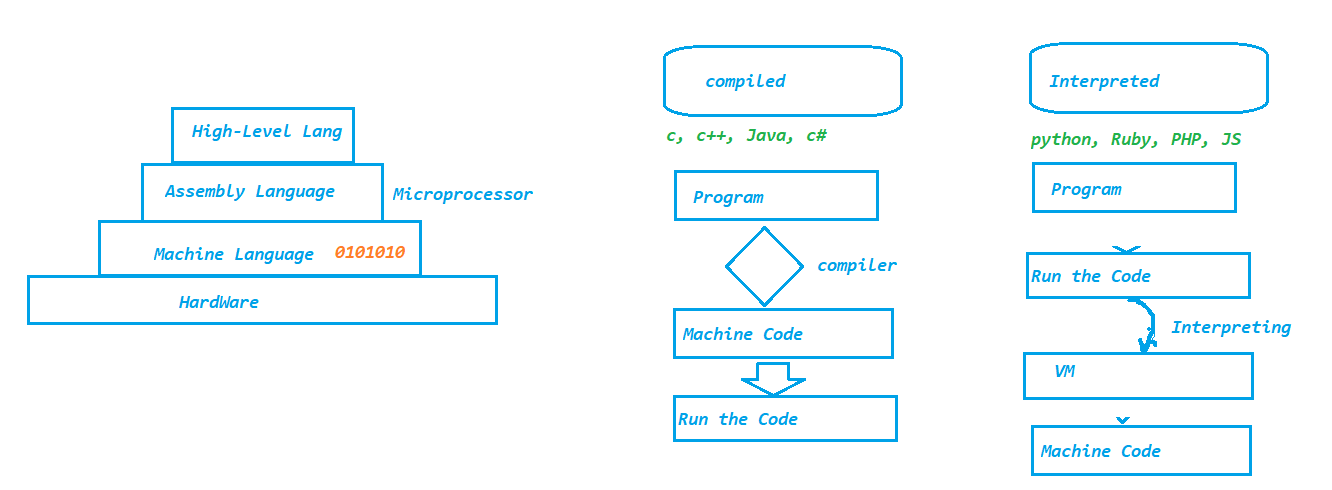
****

**Why Protractor ????**

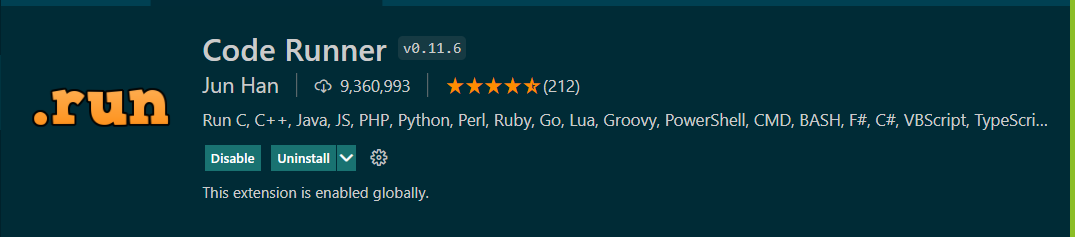
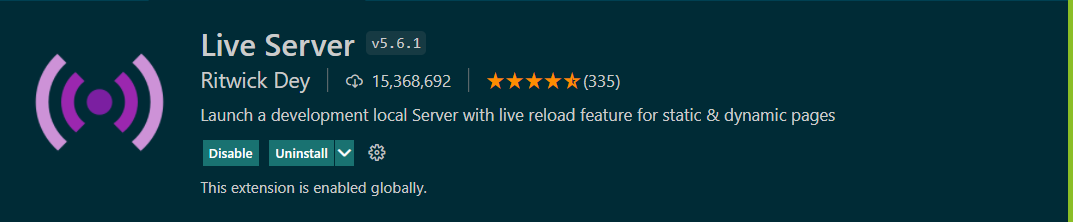
****

**JavaScript:**

**1995 - Brendan Eich**

****

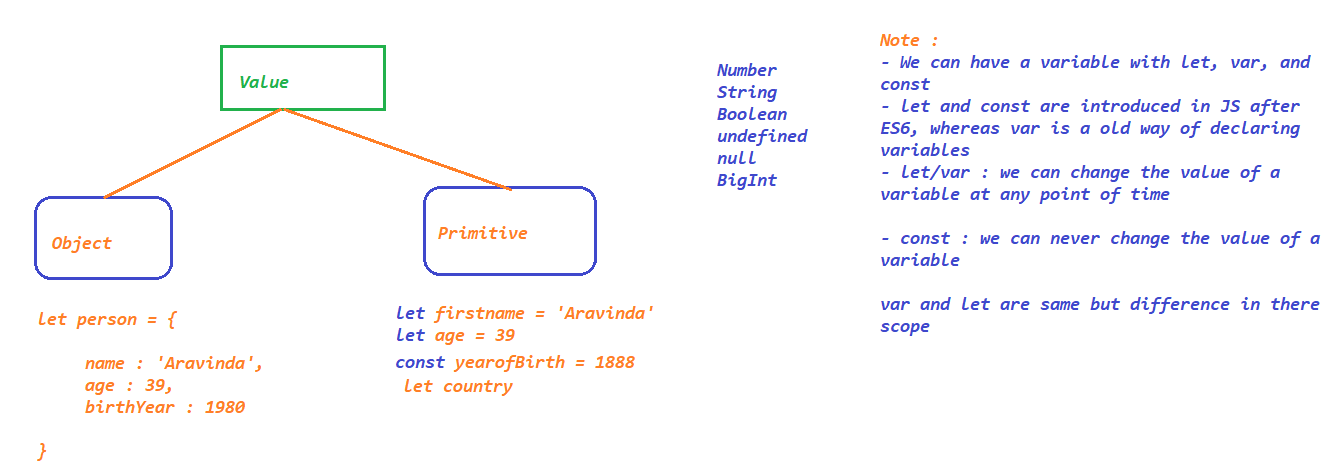
**Editor Installation for Protractor :**

1. **Download and install** [**https://code.visualstudio.com/download**](https://code.visualstudio.com/download)
2. **Install Extensions**
3. ****
4. 
5. 

**3. Install Node software is installed on your machine**

1. **Download and install node js from** [**https://nodejs.org/en/download/**](https://nodejs.org/en/download/)
2. **Once the installation is complete, open a new terminal and execute**
   1. **node –v**
   2. **npm –v**

**Variables in JS**

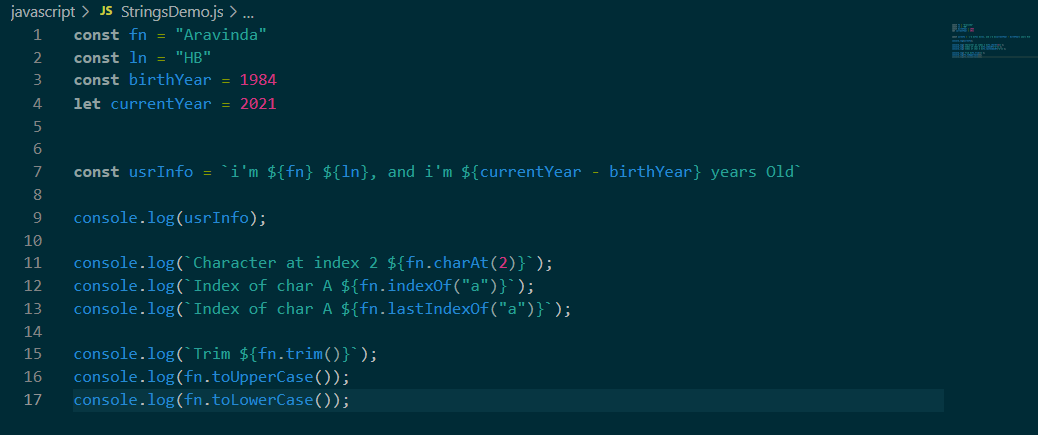
****

**Operators in JS**

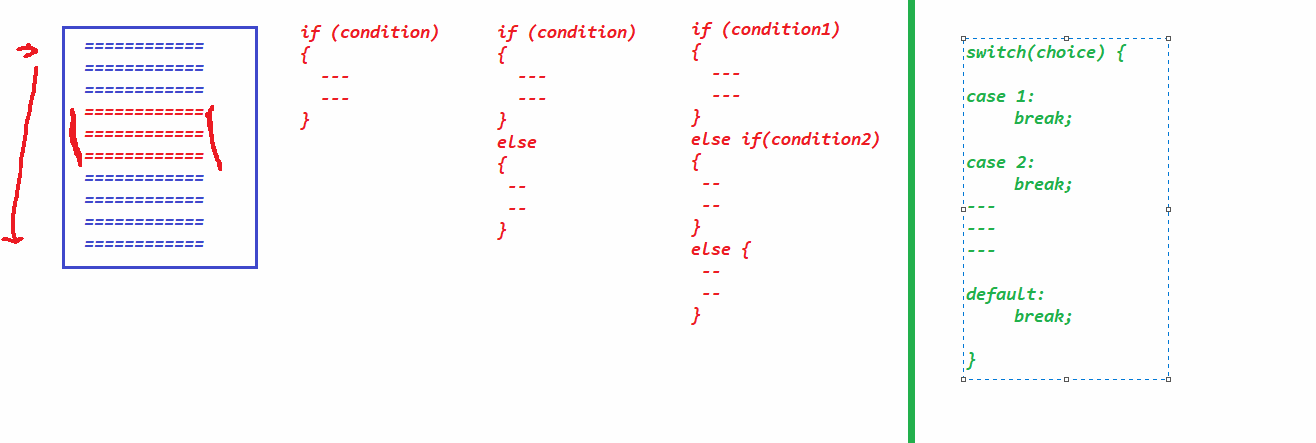
* **Arithmetic Operators**
  + **Addition +**
  + **Subtraction -**
  + **Multiplication \***
  + **Division /**
  + **Module %**
  + **Exponential \*\***
  + **Increment ++**
* **Assignment Operators**
  + **Assign = x=y**
  + **Add and Assign += x+=y x = x+y**
  + **Subtract and assign -= x-=y x= x-y**
  + **Multiply and assign \*= x\*=y x= x\*y**
  + **Divide and assign /= x/=y x= x/y**
  + **Module and assign %= x%=y x= x%y**
  + **Exponential and assign \*\*= x\*\*=y x= x\*\*y**
* **Comparison Operators**
  + **equal to ==**
  + **equal to value and type ===**
  + **not equal to !=**
  + **not equal value or type !==**
  + **greater than >**
  + **less than <**
  + **greater than or equal >=**
  + **less than or equal <=**
  + **ternary operator (condition? “first statement” : “second statements”)**
* **Logical Operators**
  + **Logical AND && -> true if both conditions are true**
  + **Logical OR || -> true when any one conditions is true**
  + **Logical NOT ! -> true if the result is false**
* **Type Operators**
* **BitWise Operators**

**Strings in JS**

**Strings are immutable in nature, we can create string literal or String object.**

****

**Conditional Statements**

****

**Loops**

* **for**
* **for in**
* **for of**
* **for each**
* **while**
* **do, while**

**Arrays**

* **length**
* **push**
* **pop**
* **shift**
* **unshift**
* **delete**
* **slice**
* **splice**
* **join**

**Functions:**

* **function with definition**
* **Function Expression or Anonymous Function (Function with out name)**
* **Arrow Functions**

**Collections**

* **Map**
* **Set**

**Objects and classes in JS:**

**Protractor :**

**Official Website :** [**https://www.protractortest.org/#/**](https://www.protractortest.org/#/)

**Protractor is an end-to-end framework or testing tool for Angular and Angular JS Applications**

**Protractor is a node.js program built on top of WebdriverJS**

**Node.js : platform to develop serverside and networking applications . its a opensource free sw**

**Installation:**

**1. Download and install node.js**

**2. Install protractor**

npm install -g protractor

**3. Update Webdriver manager**

webdriver-manager update

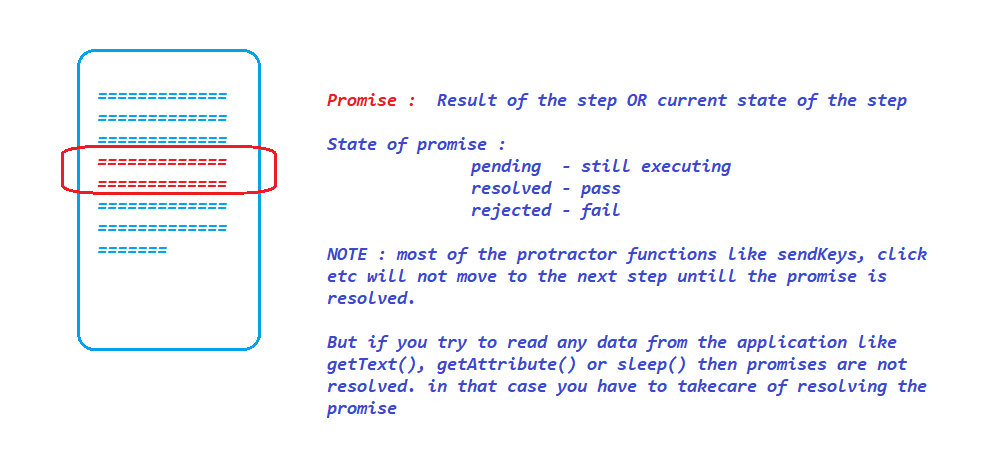
C:\Users\<<USER\_DIR>>\AppData\Roaming\npm\node\_modules\protractor\node\_modules\webdriver-manager\selenium

**4. Write the Test**

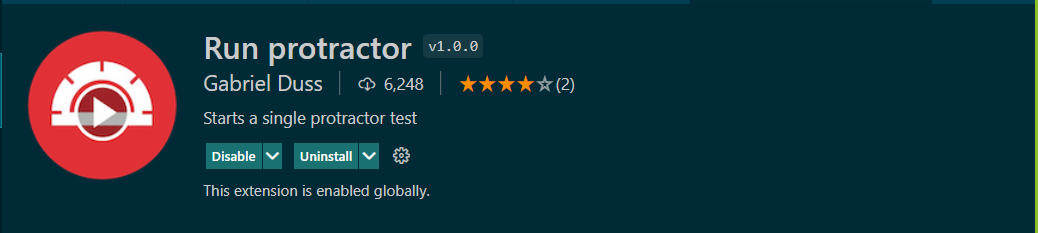
**a. copy the sample code from C:\Users\<<USERDIR>>\AppData\Roaming\npm\node\_modules\protractor\example**

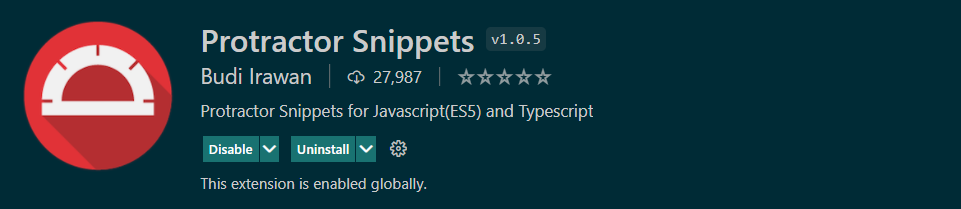
**b. paste it in the newly created folder Ex: protractor**

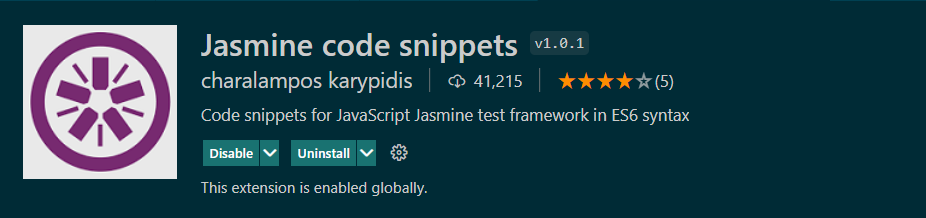
**Asynchronous nature of JS**

****

**Extensions:**

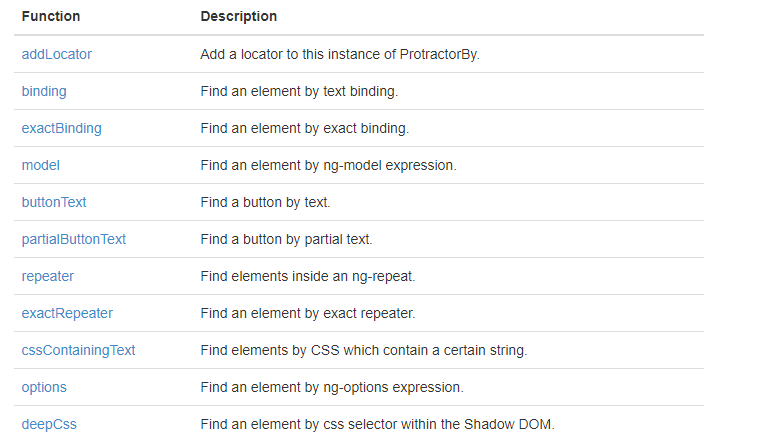
****

****

****

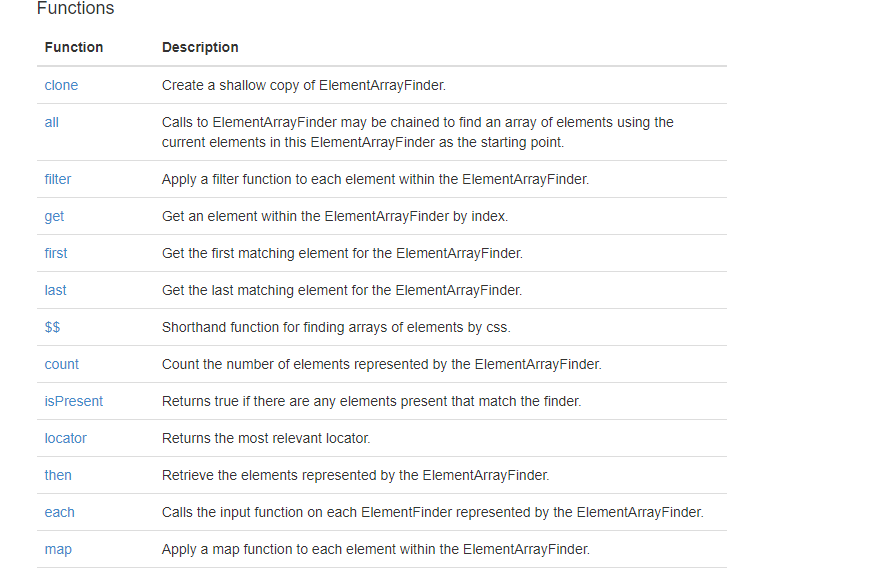
**- Install What Runs Plugin in Chrome to know what technology is used to build the application**

**Finding Elements in Protractor :**

****

**Chain of locators :**

**Finding one element from the existing or another element is called chain of locators**

****

**Capturing Screen shots on failure :**

**1. Go to npm package manager site :** [**https://www.npmjs.com/**](https://www.npmjs.com/)

**2. search for protractor-jasmine2-screenshot-reporter :** [**https://www.npmjs.com/package/protractor-jasmine2-screenshot-reporter**](https://www.npmjs.com/package/protractor-jasmine2-screenshot-reporter)

**3. Install the plugin using** npm i protractor-jasmine2-screenshot-reporter

**4. Update the conf.js with the information provided in the official document.**

**5. Execute the test**

**HTML Reports in Protractor – Allure Reports**

**1. Go to npm package manager site :** [**https://www.npmjs.com/**](https://www.npmjs.com/)

**2. search for jasmine allure report** [**https://www.npmjs.com/package/jasmine-allure-reporter**](https://www.npmjs.com/package/jasmine-allure-reporter)

**3. Install the plugin using** npm i jasmine-allure-reporter

**4. Update the conf.js with the information provided in the official document.**

**5. Execute the test and verify xml output is generated**

**6. Install allure command line by** npm i allure-commandline

**7. Execute allure serve “Location of allure xml result files”**

**HTML Reports using protractor html reporter :**

**1. Go to npm package manager site :** [**https://www.npmjs.com/**](https://www.npmjs.com/)

**2. search for protractor-html-reporter-2 :** [**https://www.npmjs.com/package/protractor-html-reporter-2**](https://www.npmjs.com/package/protractor-html-reporter-2)

**3. Install the plugin using** npm i protractor-html-reporter-2

**4. Update the conf.js with the information provided in the official document**

onPrepare: **function** () {

    jasmine.getEnv().addReporter(reporter);

**var** jasmineReporters = require('jasmine-reporters');

    jasmine.getEnv().addReporter(new jasmineReporters.JUnitXmlReporter({

      consolidateAll: true,

      savePath: './',

      filePrefix: 'xmlresults'

    }));

**var** fs = require('fs-extra');

    fs.emptyDir('screenshots/', **function** (err) {

      console.log(err);

    });

    jasmine.getEnv().addReporter({

      specDone: **function** (result) {

        if (result.status == 'failed') {

          browser.getCapabilities().then(**function** (caps) {

**var** browserName = caps.get('browserName');

            browser.takeScreenshot().then(**function** (png) {

**var** stream = fs.createWriteStream('screenshots/' + browserName + '-' + result.fullName + '.png');

              stream.write(new Buffer(png, 'base64'));

              stream.end();

            });

          });

        }

      }

    });

    jasmine.getEnv().addReporter(new AllureReporter({

      resultsDir: 'allure-results'

    }));

    jasmine.getEnv().addReporter(new AllureReporter());

    jasmine.getEnv().afterEach(**function** (done) {

      browser.takeScreenshot().then(**function** (png) {

        allure.createAttachment('Screenshot', **function** () {

          return new Buffer(png, 'base64')

        }, 'image/png')();

        done();

      })

    });

  }

onComplete: **function** () {

**var** browserName, browserVersion;

**var** capsPromise = browser.getCapabilities();

    capsPromise.then(**function** (caps) {

      browserName = caps.get('browserName');

      browserVersion = caps.get('version');

      platform = caps.get('platform');

**var** HTMLReport = require('protractor-html-reporter-2');

      testConfig = {

        reportTitle: 'Protractor Test Execution Report',

        outputPath: './',

        outputFilename: 'ProtractorTestReport',

        screenshotPath: './screenshots',

        testBrowser: browserName,

        browserVersion: browserVersion,

        modifiedSuiteName: false,

        screenshotsOnlyOnFailure: true,

        testPlatform: platform

      };

      new HTMLReport().from('xmlresults.xml', testConfig);

    });

  }

**5. Run the Test**

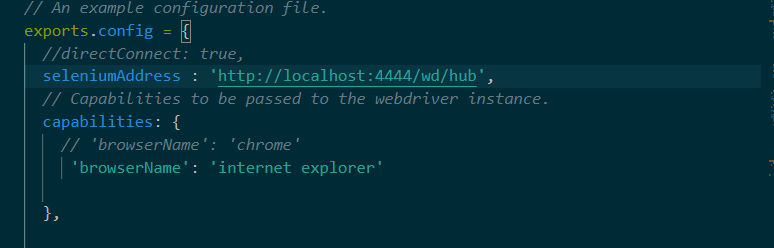
**Executing tests on a Different Browser :**

**1. Download the browser driver by Webdriver-manager update –ie**

**2. Start the server manually - Webdriver-manager start**

**3. Update the conf.js,**

**- remove directConnect and add seleniumAddress**

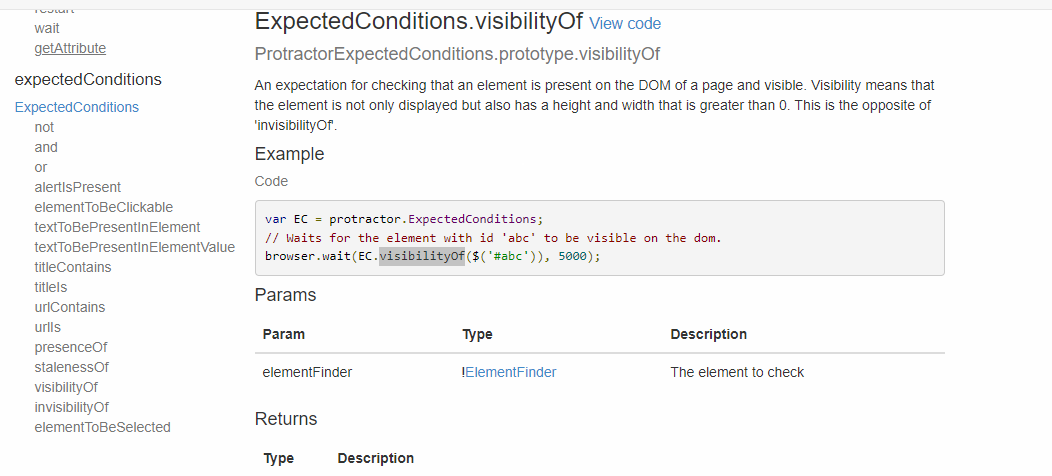
**- update the browsernaame **

**Automate Non angular applications**

**Handling Sync issues in Protractor**

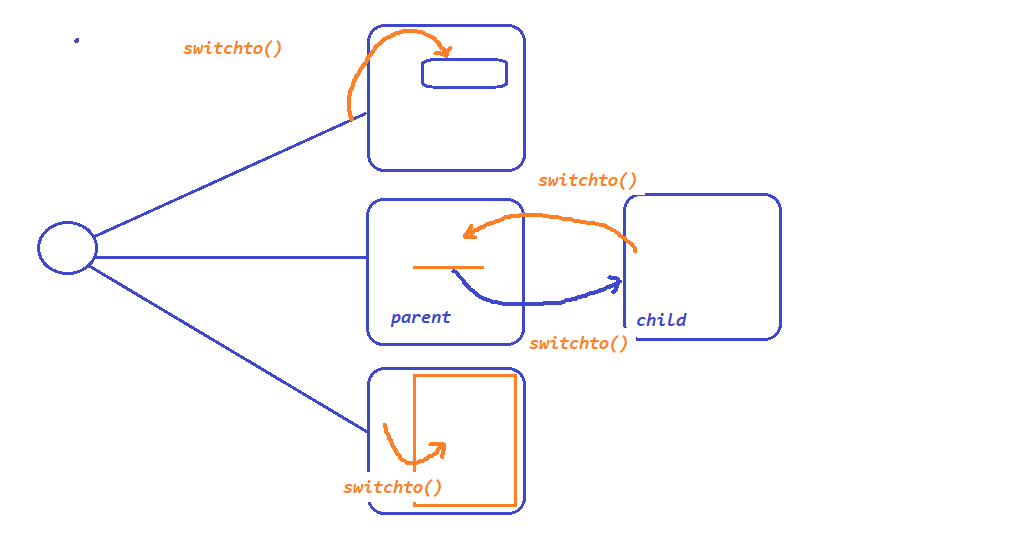
**We can use ExpectedConditions from protractor object to handle the sync issues while automating non-Angular applications**

**var** ec = protractor.ExpectedConditions;

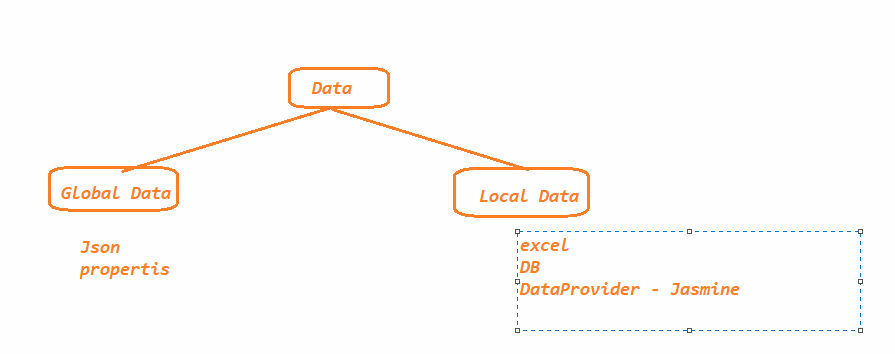


Actions :

SwitchTo



Data Driven Testing in protractor



Dataprovider :

1. go to npm library and search from jasmine data provider <https://www.npmjs.com/package/jasmine-data-provider>

2. install the plugin - npm i jasmine-data-provider

3.

Executing Test Suite :

POM

