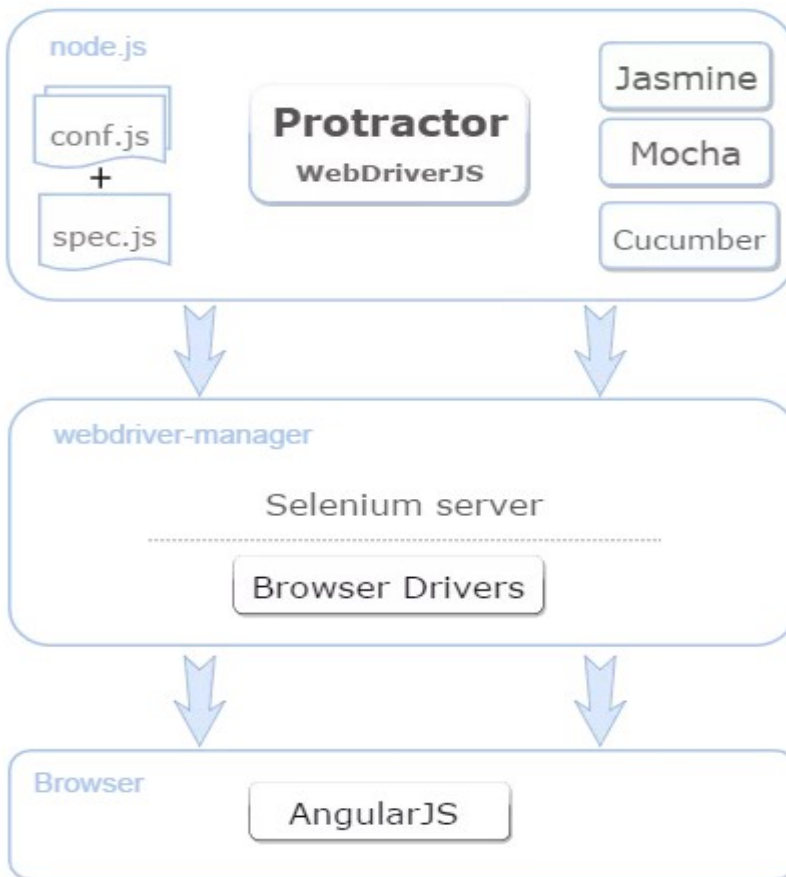


## Protractor Architecture



Create a folder in driver and open this folder using visual studio code

Download and install node.js software in the system

➤ **Npm -v** ( This command will give installed node version)

Now run **npm init**

**Now we have to provide some details.** Then after Package. json file will be created

Open package.json file and add dependencies and dev-Dependencies

```
"dependencies": {
  "@types/jasmine" : "^2.8.8"
},
"devDependencies": {
```

```

"@types/node" : "^10.5.2",
"jasmine": "^3.1.0",
"protractor": "^5.3.2",
"typescript": "^2.9.2"
}

```

If you want to .ts file we don't have any main method like java . So we have to follow below approach

Inside package.json file

```

"scripts": {
  "tsc": "tsc",
  "pretest": "npm run tsc",
  "test" : "protractor outputjs/conf.js"
},

```

Now open terminal/command prompt and run **npm install**

A folder with name node\_modules folder will get created

Now navigate to this path and run command

node\_modules\typescript\bin\tsc -init

tsconfig.json file was successfully created

add below details to tsconfig.json file

```

"outDir": "outputjs",
"types": ["jasmine", "node"],
},
"exclude": [
  "node_modules/*"
]

```

```

import { browser, element, by } from "protractor";
import { log4jsconfig } from '../config/log4jsconfig'

describe("Calculator test", function(){

  beforeEach(function(){
    browser.get("https://juliemr.github.io/protractor-demo/");

```

```

    })

    it("Launch url check", function(){
        expect(browser.getTitle()).toContain("Super");
        //console.log("Browser Title :-" + browser.getTitle());
        let browserTitle = browser.getTitle();

        browserTitle.then(function(txt){
            console.log("Browser Title :-" + txt);
            log4jsconfig.Log().debug("Browser Title :- " + txt);
        });

    })

    it("Add 2 numbers", function(){
        element(by.model("first")).sendKeys("12");
        element(by.model("second")).sendKeys("13");
        element(by.id("gobutton")).click();
        browser.sleep(3000);

        expect<any>(element(by.xpath("//table/tbody/tr[1]/td[3]")).getText()).toEqual('25');
    })

})

```

```

import {Config, browser} from 'protractor';
var screenreporter = require('util/screenreporter.js');

export let config: Config = {
    framework : "jasmine2",

    jasmineNodeOpts:{
        showColors : true,
        silent : true,
        defaultTimeoutInterval: 36000,
        print: function(){}
    },

    capabilities:{
        browserName : 'chrome',
        //browserName : 'firefox',
        //marionette : true,

```

```

    acceptSslCerts : true
  },

  suites: {
    calc : './specs/calculator.js',
    bank : './testspec/banktest.js',
    bothtest : ['./specs/calculator.js',
      './testspec/banktest.js']
    //to run each suite
    //protractor conf.js calc
    //protractor conf.js bank
    //protractor conf.js calc,bank
    //protractor conf.js bothtest
  },

  specs:['./testspec/banktestpom.js'],

  seleniumAddress: 'http://localhost:4444/wd/hub',

  onPrepare:()=>{
    var os = require('os');
    //browser.manage().window().maximize();
    browser.manage().timeouts().implicitlyWait(4000);
  }
}

```

It is all about related to logging mechanism

```

it("Launch url check", function(){
  expect(browser.getTitle()).toContain("Super");
  //console.log("Browser Title :-" + browser.getTitle());
  let browserTitle = browser.getTitle();

  browserTitle.then(function(txt){
    console.log("Browser Title :-" + txt);
    log4jsconfig.Log().debug("Browser Title :- " + txt);
  });

})

```

```

{
  "name": "protractordemo",

```

```

"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
  "tsc": "tsc",
  "pretest": "npm run tsc",
  "test": "node_modules/protractor/bin/protractor outputjs/conf.js"
},
"author": "",
"license": "ISC",
"dependencies": {
  "@types/jasmine": "^2.8.8",
  "eslint": "^5.12.1"
},
"devDependencies": {
  "@types/node": "~10.5.2",
  "jasmine": "3.1.0",
  "protractor": "~5.3.2",
  "typescript": "~2.9.2",
  "log4js": "3.0.1"
}
}

```

## Log4js.json

```

{
  "appenders": {
    "console" : {
      "type": "console",
      "category": "console"
    },
    "file" : {
      "category": "test-file-appender",
      "type": "file",
      "filename": "log_file.log",
      "maxLogSize": 10240,
      "backups": 3,
      "pattern": "%d{dd/MM hh:mm} %-5p %m"
    }
  },
  "categories": {
    "default" :{"appenders": ["console", "file"], "level": "DEBUG"},

```

```

    "file" : {"appenders": ["file"], "level": "DEBUG"}
  }
}

```

## Log4jsconfig.ts

```

export class log4jsconfig{

  static Log(): any {
    var log4js = require('log4js');
    log4js.configure('./config/log4js.json');
    //var logger = log4js.getLogger(); //for both console and file
    let log = log4js.getLogger("default");
    return log;
  }
}

```

```

import { browser, element, by } from "protractor";
import { log4jsconfig } from '../config/log4jsconfig'

describe("Calculator test", function(){

  beforeEach(function(){
    browser.get("https://juliemr.github.io/protractor-demo/");
  })

  it("Launch url check", function(){
    expect(browser.getTitle()).toContain("Super");
    //console.log("Browser Title :-" + browser.getTitle());
    let browserTitle = browser.getTitle();

    browserTitle.then(function(txt){
      console.log("Browser Title :-" + txt);
      log4jsconfig.Log().debug("Browser Title :- " + txt);
    });

  })

  it("Add 2 numbers", function(){
    element(by.model("first")).sendKeys("12");
    element(by.model("second")).sendKeys("13");
    element(by.id("gobutton")).click();
  })
}

```

```

        browser.sleep(3000);

expect<any>(element(by.xpath("//table/tbody/tr[1]/td[3]")).getText()).toEqual('25');
    })

})

```

Working with web-elements

<http://www.globalsqa.com/angularJs-protractor/BankingProject/#/login>

working with alerts

```

function VerifyAndCloseAlert(text: String){
    let EC = protractor.ExpectedConditions;
    browser.wait(EC.alertIsPresent(), 4000, "ALert not found");

    let alert = browser.switchTo().alert();
    let alertText = alert.getText();

    alertText.then(function(txt){
        log4jsconfig.Log().debug(txt);
    })

    browser.sleep(2000);
    expect(alertText).toContain(text);
    alert.accept();
}

```

```

element(by.buttonText('Open Account')).click();
let Customers = element(by.model('custId'));
let options = Customers.all(by.tagName('option'));

options.then(function(items){
    log4jsconfig.Log().debug("Dropdown option size " +
items.length);
    for(let i=0 ; i < items.length ; i++){
        items[i].getText().then(function(txt: any){
            log4jsconfig.Log().debug(txt);
            if(txt == "ABC XYZ"){
                log4jsconfig.Log().debug("Item found on the
list");

```

```

        items[i].click();
    }
    })
}
})

```

Static drop down handling

```

element(by.model('currency')).$('[value="Dollar"]').click(); // $ = by.css

```

Working with tables

```

element(by.buttonText('Customers')).click();
    let rows = element.all(by.repeater
('cust in Customers | orderBy:sortType:sortReverse |
filter:searchCustomer'));
    browser.sleep(10000);

    rows.each(function(row: any){
        let cells = row.$$('td'); //all(by.css)
        cells.get(0).getText().then(function(txt: any){
            if(txt == 'ABC'){
                cells.get(4).$('button').click();
            }
        })
    })
})

browser.sleep(10000);

```

Data-Driven testing

Create prop1.ts file

```

module.exports = {
    "siteurl" :
"http://www.globalsqa.com/angularJs-
protractor/BankingProject/#/manager/addCust",
    "customers" :{
        "firstname" : "John",
        "lastname" : "Carry",
        "postalcode" : "87654"
    }
}

```



```
}  
}
```

```
prop1 = require("../testdata/prop1");  
firstname = this.prop1.customers.firstname;  
    lastname = this.prop1.customers.lastname;  
    pcode = this.prop1.customers.postalcode;
```

## Run test Suites

Add suites section under config.ts file

```
suites: {  
  calc : './specs/calculator.js',  
  bank : './testspec/banktest.js',  
  bothtest : ['./specs/calculator.js',  
    './testspec/banktest.js']  
  //to run each suite  
  //protractor conf.js calc  
  //protractor conf.js bank  
  //protractor conf.js calc,bank  
  //protractor conf.js bothtest  
},
```

```
"test": "node_modules/protractor/bin/protractor outputjs/conf.js --suite calc"
```

## Work with Non-angular Sites

```
browser.ignoreSynchronization = true;  
browser.get("http://qavalidation.com/demo/");  
browser.sleep(1000);
```

```
onPrepare():=>{  
  (global as any).isAngularSite = function(flag: boolean){  
    browser.ignoreSynchronization = !flag;  
  }  
}
```

```
it("invoke non angular app",function(){  
  //browser.ignoreSynchronization = true;  
  (global as any).isAngularSite(false);  
})
```

## Handling multiple window using protractor

<https://skpatro.github.io/demo/links/>

```
import { log4jsconfig } from '../config/log4jsconfig'

describe("Handle multiple windows", function(){

  beforeEach(function(){
    browser.ignoreSynchronization = true;
    browser.get("https://skpatro.github.io/demo/links/");
  })

  it("Multiple windows handle", function(){
    browser.getTitle().then(function(txt){
      console.log("Main browser title :- " + txt);
    })

    browser.findElement(by.name("NewWindow")).click();

    let windowHandles = browser.getAllWindowHandles();
    let parentHandle, childHandle;
```

```
    windowHandles.then(function(handles){
      parentHandle = handles[0];
      childHandle = handles[1];
      console.log("Total Handles :- " + handles.length);

      browser.switchTo().window(childHandle).then(function(){
        browser.getTitle().then(function(txt){
          console.log("Child browser title :- " + txt);
          browser.close();
        })
      })

      browser.switchTo().window(parentHandle).then(function(){
        browser.getTitle().then(function(txt){
          console.log("Main browser title :- " + txt);
        })
      })
    })
  })
})
```

<https://github.com/sunilpatro1985/Protractor-typescript>

<http://qavalidation.com/category/protractor/>

