Table of Contents

Aug 27th Lecture Test Automation with Protractor and Jasmine

Will use this website for the testing project
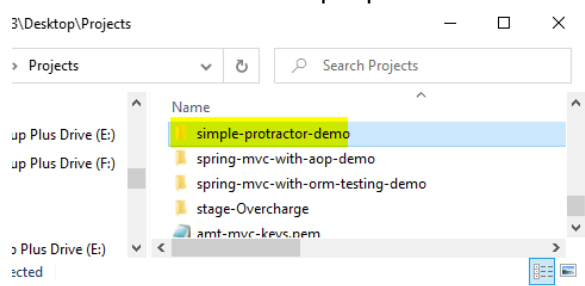https://angular.io
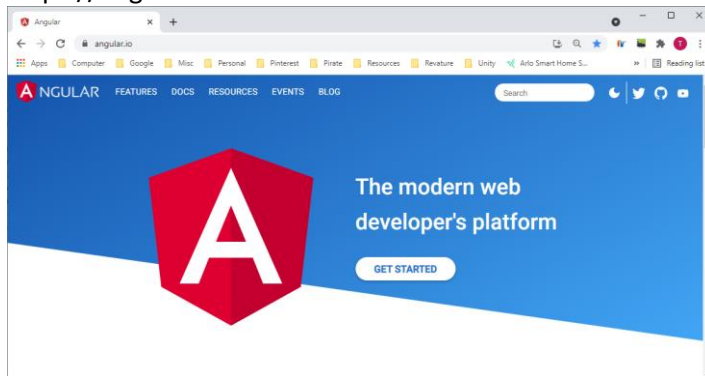
Create a new project folder

In your project folder
Create a folder name: simple-protractor-demo

Inside the simple-protractor-demo project folder open a Git Bash Window
Type the following command

```
npm init
```

Respond to the prompts

```
package name: (simple-protractor-demo) ← Press Enter to accept default
version: (1.0.0) ← Press Enter to accept default
description: A basic demo of protractor ← enter a description
entry point: (index.js) ← Press Enter to accept default
test command: ← Press Enter to accept default
git repository: ← Press Enter to accept default
keywords: ← Press Enter to accept default
author: your name ← enter your name
license: (ISC) ← Press Enter to accept default
About to write to C:\Users\tlw8748253\Desktop\Projects\simple-protractor-demo\package.json:

{
  "name": "simple-protractor-demo",
  "version": "1.0.0",
  "description": "A basic demo of protractor",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "your name",
  "license": "ISC"
}
```

```
Is this OK? (yes) yes  ← enter yes
```

```
MINGW64:/c/Users/tlw8748253/Desktop/Projects/simple-protractor-demo

tlw8748253@TLW8748253-DL13 MINGW64 ~/Desktop/Projects/simple-protractor-demo
$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (simple-protractor-demo)
version: (1.0.0)
description: A basic demo of protractor
entry point: (index.js)
test command:
git repository:
keywords:
author: your name
license: (ISC)
About to write to C:\Users\tlw8748253\Desktop\Projects\simple-protractor-demo
son:

{
  "name": "simple-protractor-demo",
  "version": "1.0.0",
  "description": "A basic demo of protractor",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "your name",
  "license": "ISC"
}

Is this OK? (yes) yes
```

Install Protractor Globally

For more information on Protractor visit
http://www.protractortest.org/#/

Setup

Use npm to install Protractor globally with:

```
npm install -g protractor
```

This will install two command line tools, `protractor` and `webdriver-manager`. Try running `protractor --version` to make sure it's working.

The `webdriver-manager` is a helper tool to easily get an instance of a Selenium Server running. Use it to download the necessary binaries with:

```
webdriver-manager update
```

Now start up a server with:

```
webdriver-manager start
```

This will start up a Selenium Server and will output a bunch of info logs. Your Protractor test will send requests to this server to control a local browser. You can see information about the status of the server at http://localhost:4444/wd/hub.

In the Git Bash window type

```
npm install -g protractor
```

```
MINGW64:/c/Users/tlw8748253/Desktop/Projects/simple-protractor-demo

tlw8748253@TLW8748253-DL13 MINGW64 ~/Desktop/Projects/simple-protractor-demo
$ npm install -g protractor
```

Update the web driver

```
webdriver-manager update
```


Update the package.json file

This update will add the web driver manager commands to the project

Replace:

```
"test": "echo \"Error: no test specified\" && exit 1"
```

With:

```
"test": "webdriver-manager start & protractor configuration.js & webdriver-manager shutdown"
```


Install Jasmine

In the Git Bash window type

```
npm install @types/jasmine
```


Open the project in VS Code
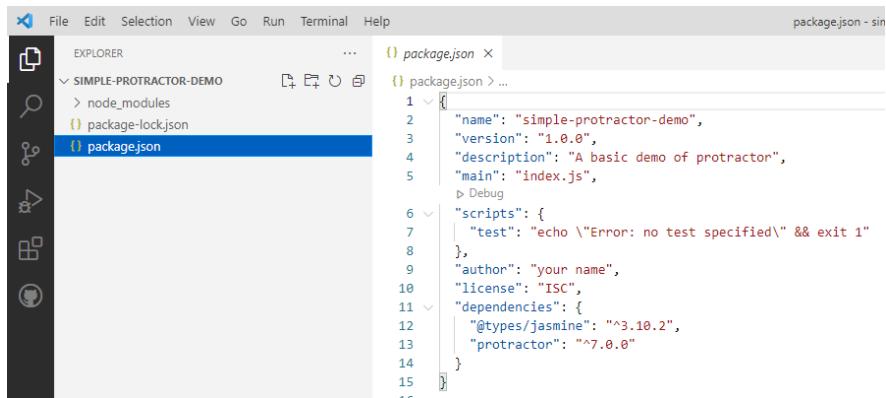
Create a configuration file

```
configuration.js
```
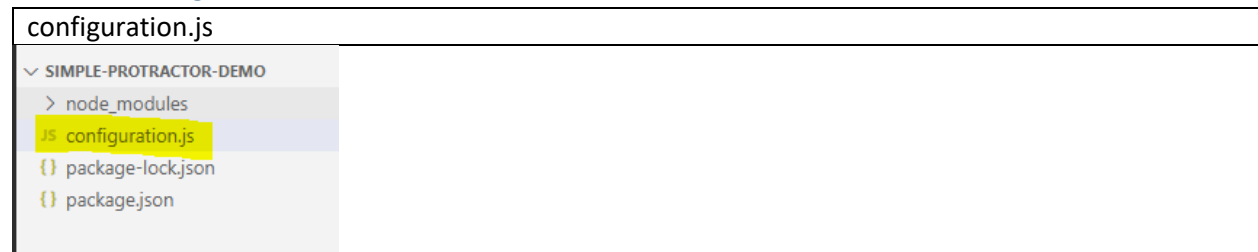

Add the test configuration to the configuration.js file

```javascript
module.exports.config = {
    seleniumAddress: 'http://localhost:4444/wd/hub',
    capabilities: {
        'browserName': 'chrome',
        'shardTestFiles': true,
        'maxInstances': 2
    },

    framework: 'jasmine',
    specs: [
        './specs/spec1.js'
    ],

    jasmineNodeOpts: {
        defaultTimeoutInterval: 30000
    }
};
```
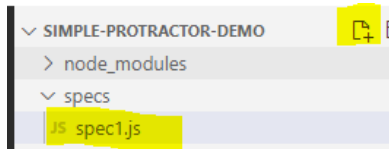
Create a new folder

Add a folder call specs
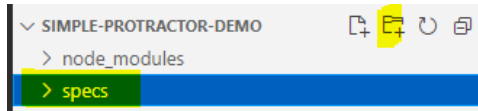
We call the folder specs because the tests inside the files we will create are considered a spec, so two test in a single file are considered two specs.

Create a spec file spec1.js

Add code to the spec1.js file

```javascript
const { browser } = require('protractor');


// This is Jasmine syntax for writing tests in JavaScript
// You can think of Jasmine as being similar to JUnit for Java
describe('this will be for a collection of tests', function() {
    it('this is test 1', function() {
        // browser is coming from protractor, which is similar to Selenium
        browser.get('https://angular.io');
    });

    it('this is test 2', function() {
        browser.close();
    });
})
```

Run the project

For some reason trying to run the project through the definition in the project files did not work. So the lecturer did manual steps to run the project. The manual steps are defined here and the steps that did not work are left out.

You might get a Windows Defender alert with the following commands if so click "Allow access"
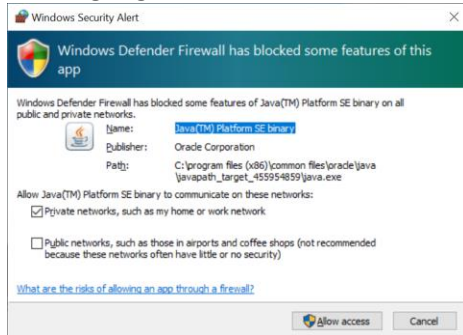
Open a Git Bash terminal inside VS Code

Enter the update command

```
webdriver-manager update
```


Enter the start command

```
webdriver-manager start
```


In the Git Bash Window in the Project Folder

Enter the test command

```
protractor configuration.js
```

The browser defined in the spec1.js file will pop up briefly. This indicates that the program is coded and configured correctly.
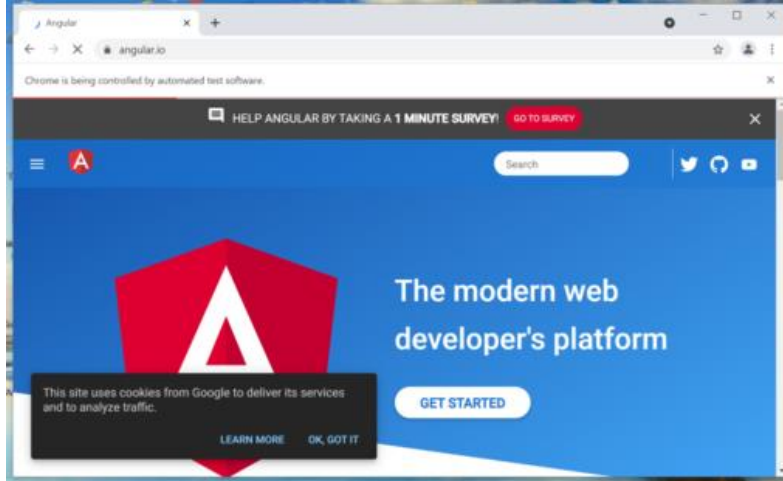
The Final Project

The lecture ends stating the project will be continued on Monday Aug 30th. However this project is not updated on Aug 30th. Comparing the files in the training repository to what was done in this document, no further updates were performed. This is the final project with Protractor.