Prerequisites

1. Verify you have npm 3+ installed (run `npm -v` to check).
2. Install Yeoman and the aspnetcore-spa generator globally:

   ```
   npm install -g yo generator-aspnetcore-spa
   ```

3. Intall Webpack globally:

   ```
   npm install -g webpack
   ```

Creating a new Angular 2 application

1. From a commandline, create and navigate to a new empty directory.
2. Create a new Angular 2 application using the "aspnetcore-spa" scaffolder:

   ```
   yo aspnetcore-spa
   ```

3. View the application code by typing `code .` to launch Visual Studio Code in the current directory.

Running the Angular 2 application

1. From the commandline, set the ASP.NET Core development mode environment variable:

   ```
   set ASPNETCORE_ENVIRONMENT=Development
   ```

   Note: On OSX this is done using `export ASPNETCORE_ENVIRONMENT=Development`

1. Run the application using the `dotnet watch` tool:

   ```
   dotnet watch run
   ```

2. Navigate to `http://localhost:5000` to view the application.

   Note: Leave the application running and the browser window open for the remainder of the lab.

Experiment with Hot Module Reloading (HMR)

1. Navigate to the Counter page in running web application.
2. In Visual Studio Code, edit the Counter template (\ClientApp\components\counter\counter.html) to change the H2 heading text.
3. Edit the counter implementation ((\ClientApp\components\counter\counter.ts) to change the counter increment to 10:

   ```
   export class Counter {
     public currentCount = 0;

     public incrementCounter() {
         this.currentCount+=10;
   ```

```
    }
}
```

4. Observe that the application has refreshed with your changes. View the console output to see the debug messages printed out during the updates.

Extra

1. Create a Knockout, React, or React + Redux application using the above steps but selecting a different option in the Yeoman scaffolder.

2. Create a production build by setting `ASPNETCORE_ENVIRONMENT=Production`, then running the following:

   ```
   webpack --config webpack.config.vendor.js
   webpack
   ```

Then inspect the front-end resources in browser tools and verify that the resources are minified.