CST-391 Activity 2 Guide

Contents

[Part 1: Tools Installation 2](#_Toc127347537)

[Supplemental Tutorial 3](#_Toc127347538)

[Appendix: Implementation Notes 4](#_Toc127347539)

# Part 1: Tools Installation

**Overview**

In this activity, students will install the development tool and validate the tools installation with a simple test application running on Angular.

The students should refer to the implementation notes in the Appendix for this activity.

**Execution**

Execute this assignment according to the following guidelines:

1. Install Angular CLI:
   1. On Windows, open up a Command Prompt, or on Mac, open up the Terminal application and run the following command (**Note:** on Mac, you may need to run as sudo):

npm install -g @angular/cli

* 1. Verify the version of Angular by running the following command and then take a screenshot:

ng version

1. Create a test application to validate the environment:
   1. In the Terminal, navigate to the location where you want the test application to be created.
   2. Run the following Angular CLI command and select the Angular route and CSS options during the installation:

ng new testapp

* 1. Create a VS Code Workspace by performing the following steps:
     1. Open VS Code. Close the existing Workspace if one is opened.
     2. Click the App Workspace Folder hyperlink from the VS Code start screen.
     3. Select the testapp directory where the test application was created.
     4. Select the File > Save Workspace As menu options and save your Workspace so VS Code can be opened again in the future.
  2. Open the internal Terminal in VS Code and run the following command to start the Node server. Your browser should automatically be started with the application running. Take a screenshot.

ng serve -o

* 1. Open the Component *app.component.js* file in VS Code. Notice that the default *index.html* page is rendering this Component by using the *<app-root>* tag, which is specified in the Components *selector* property and *templateUrl* properties that are both specified in the @Component decorator. Also notice that the *title* variable from the Component is referenced in the Component Template using the syntax {{ title }}. Change the value of the *title* variable. Validate that the application is automatically refreshed in your browser. Take a screenshot.
  2. Modify the Component *app.component.js* file by adding a new class member variable named *message* and set this value to your name. Render this value in an h3 Html tag in the Component Template. Take a screenshot.

1. Research:
2. Inspect the default test project structure created in the Activity. Describe the purpose for each of the folders of the following in the project structure: node\_modules, src, src/app, src/assets, and src/environments. Also, describe the purpose for each of the following files in the project: angular.json, package.json, and tsconfig.json,
3. Inspect the page source for the default page displayed when running the test project. Explain how the resultant page was generated by Angular by providing a brief overview and purpose for each of the following files: main.ts, app.component.css, app.component.html, app.component.ts, and app.module.ts.

**Submission**

Submit the following in a Microsoft Word document as directed by the instructor:

1. All screenshots (including captions).
2. Research questions and write-up.

# Supplemental Tutorial

Reference the following tutorial as necessary when completing the activity:

* **Angular Tutorials**: <https://angular.io/tutorial>

# Appendix: Implementation Notes

**Note:** if you ever need to rebuild the *node\_modules* directory in a project, just delete the directory, and run the *npm install* command within the projects *root* directory.

**Part 1 Implementation Notes**



Figure 1 *Component Class*



Figure 2 *Component View Template*



Figure 3 *Application Page*