Setting up the environment - Demo

**Demosteps:**

In order to set up Angular environment try out the following demo steps:

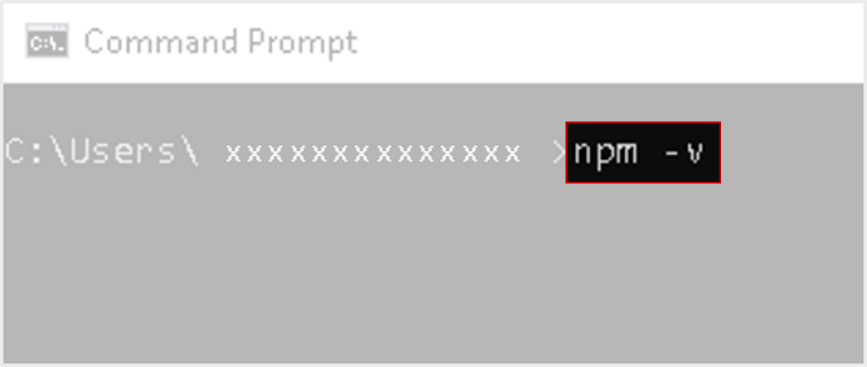
**Step 1:**

Angular CLI is a command line interface tool that is used to create, build and deploy Angular applications.

To install Angular CLI you require Node package manager (npm).

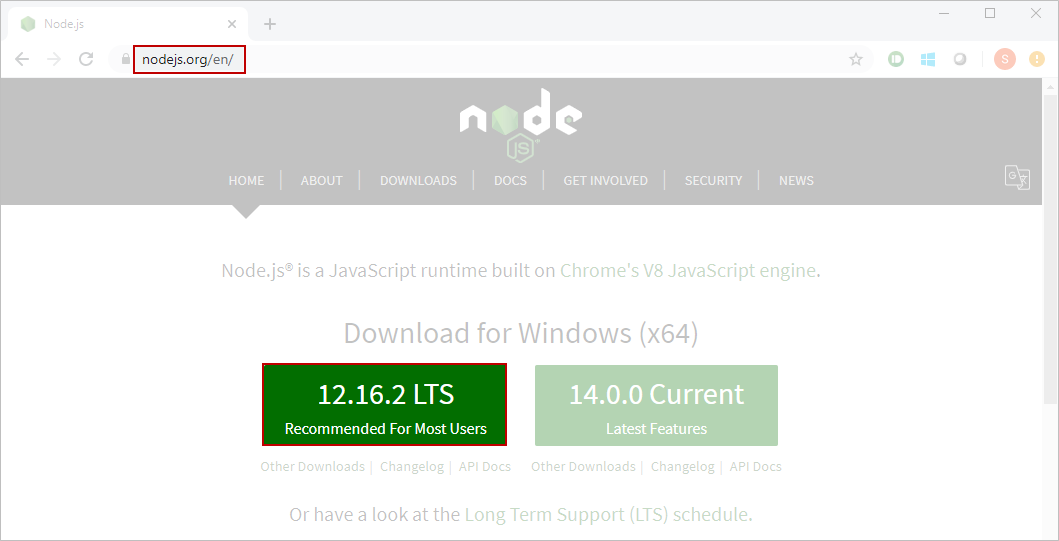
To check if npm is installed type the following command incommand prompt window,

1. npm –v



**Step 2:**

If npm is not installed, open the link( <https://nodejs.org/en/>) in the browser. Download and install the current version(12+) of node.js with latest features



Node.js has a package ecosystem named **npm**(**Node Package Manager**). It is a command line utility that interacts with the repository of open source projects to install the libraries, packages for angular.

**Step 3:**

Search for **'Node.js command prompt'**in your windows search window and open it.

You will be configuring npm to change registry in order to read package info. Type the following command in your command prompt.

1. npm config set registry https:*//infyartifactory.ad.infosys.com/artifactory/api/npm/npm-remote/*

**Note: Use the above command only if you are connected to company network. If you use your personal device to install angular cli the above command is not required**

Once you have set the registry, run the below command for logging in.

1. npm login

Now you will be asked to enter your credentials. Provide the credentials as shown below.

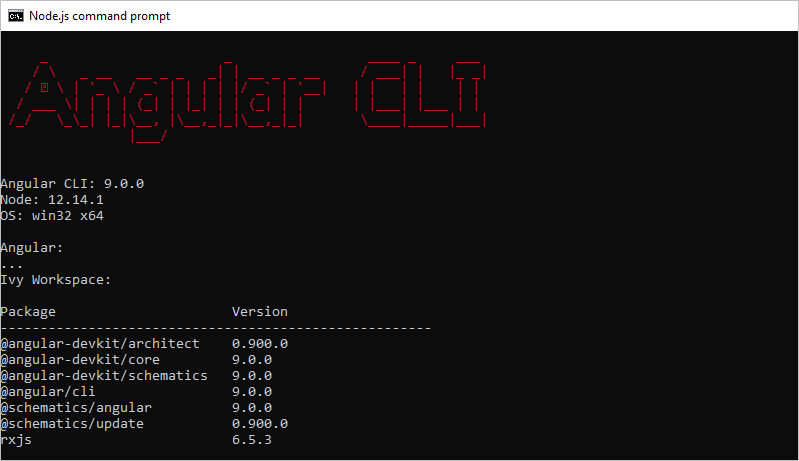


Type the following command to start the installation.

1. npm install –g @angular/cli@9.0.0

Note: If the version is not specified it will install the latest version of cli

Once installation is complete the following message will be displayed



Note: In case the above message is not displayed run any one of the following command in the prompt

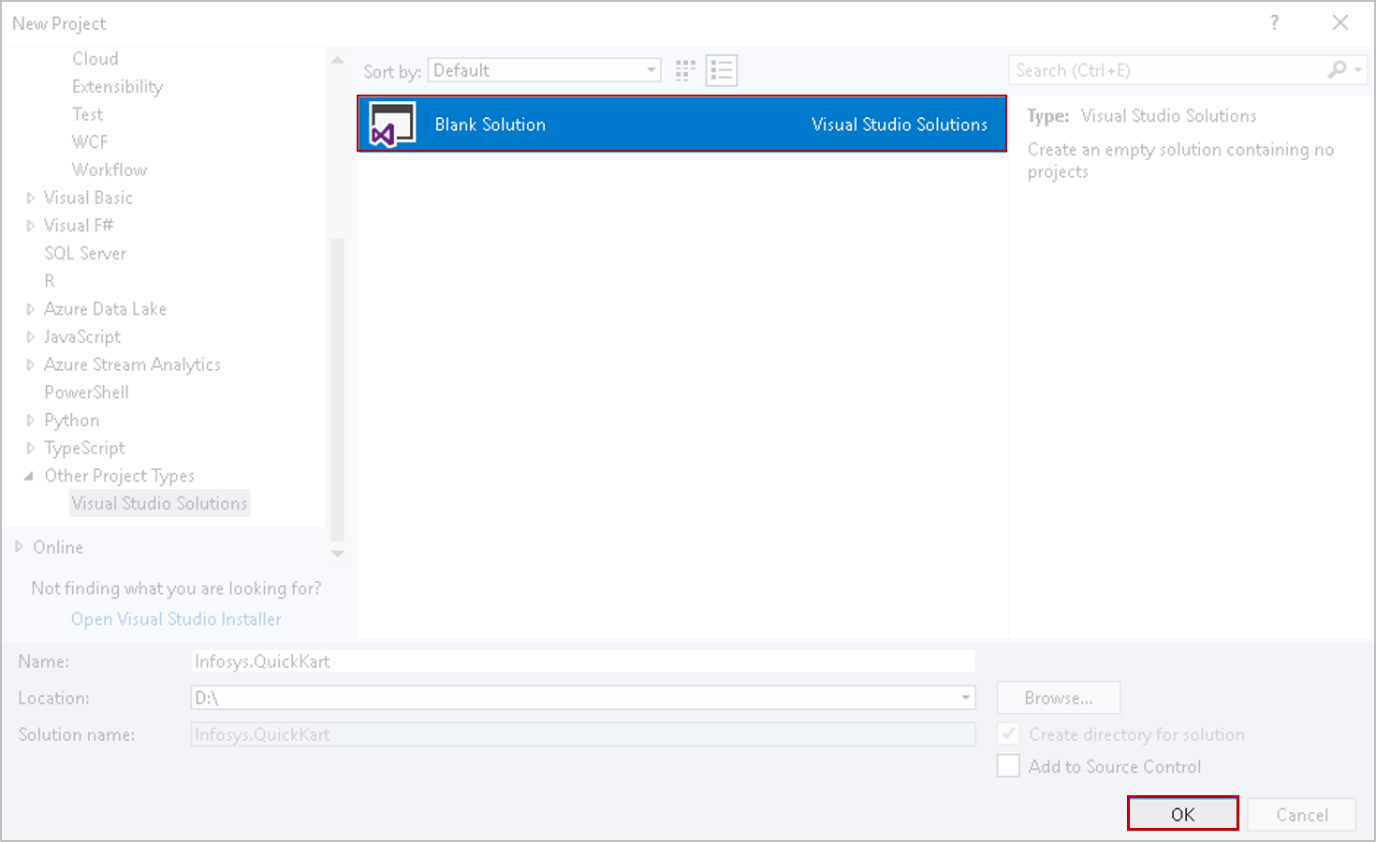
1. ng v
2. ng version

This should display the above message giving the version of the cli installed.

**Step 4:**

To create a new project we first need to create a blank solution in visual studio.

Open Visual Studio, go to File--> New--> Project. In the New Project window pane go to Other Project Types--> Visual Studio Solutions and select the Blank Solution template. Name this solution as Infosys.QuickKart and save it in your local path as shown below.



**Step 5:**

In your Node.js command prompt go to the directory in which you created the blank solution.

To create a new angular project in the directory type the following cli command in the command prompt window and press enter

1. ng new QuickKartApp



**ng** – to call Angular CLI.

**new** - one of the many commands issued to the CLI.

**QuickKartApp** is the name of the project.

Select No when be prompted to add Angular routing. Select CSS for the stylesheet format.

Angular CLI installs the necessary packages, creates the project files and populates the project with a default app.

**Step 6:**

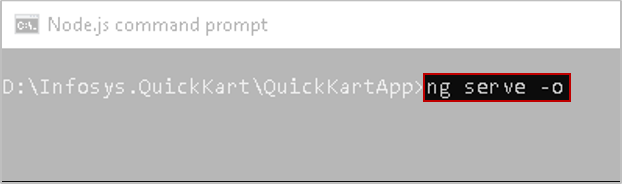
Once CLI has generated the new project, you should navigate into the project by typing the command in command prompt

1. cd QuickKartApp

**Step 7:**

In order to build and launch the application in the browser type the following command.

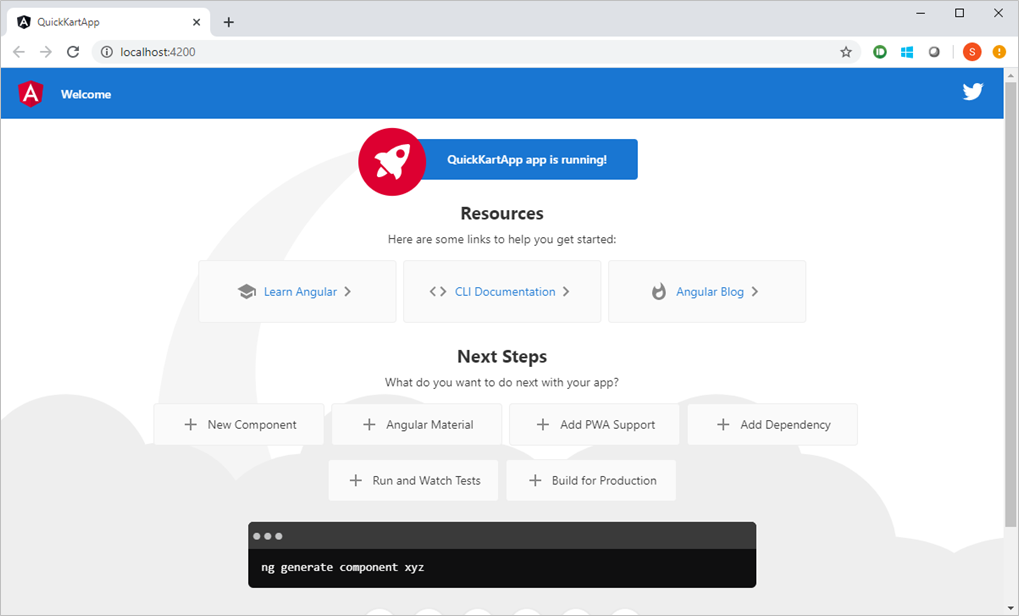
1. ng serve -o



**ng serve** command launches the server

**--o or --open** option will automatically open the browser on <http://localhost:4200/>

You can view your app in the browser



# Understanding Angular flow - Demo

Demosteps:

In order to understand the flow of execution of QuickKartApp application, try out the following demo steps:

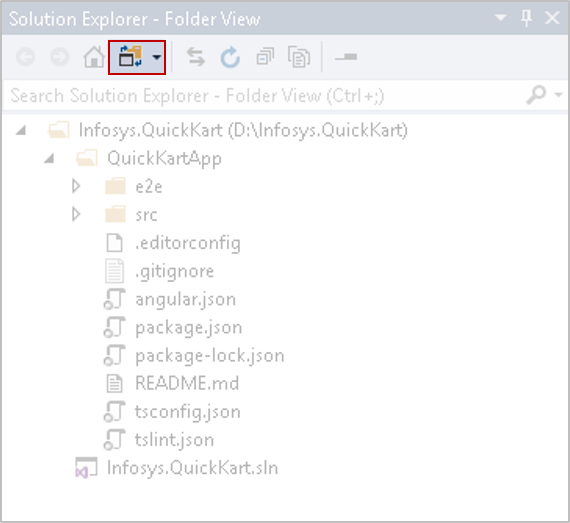
**Step 1:**

Open Visual Studio and select File -> Open -> Project/Solution.

Browse to Infosys.QuickKart and select the Infosys.QuickKart.sln file to open in Visual studio.

**Step 2:**

Open the Solution Explorer in Folder View and you should be able to view the QuickKarApp  as shown below.



Note : This base project structure is created because you have created the project using Angular CLI.

You will understand the various files in the project folder.

**src -**all application related files will be stored inside this folder.

**angular.json** - configuration for Angular CLI. In this file you can set several defaults and also configure what files are included when your project is built.

**package.json -**node configuration file which contains all dependencies required for your project. Node.js creates a folder **node-modules**and puts all the modules listed here in that folder. You can browse to node-modules folder present inside Infosys.QuickKart--> QuickKartApp in File Explorer and observe it.

**tsconfig.json -**typeScript compiler configuration for your IDE to pick up and give you helpful tooling.

**tslint.json -**used when running **ng lint**. Linting helps keep your code style consistent.

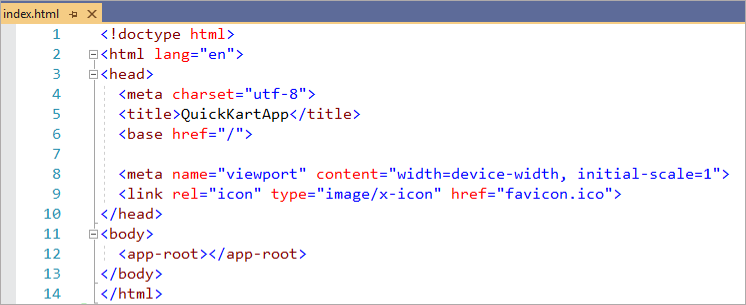
**README.md -**basic documentation for your app.

**Step 3 :**

When the QuickKartApp project is built and executed, the first thing to be loaded is index.html file present in the QuickKartApp --> src folder.

The Angular CLI automatically adds the .ts and .css files while building the app so you don’t need to add it manually.

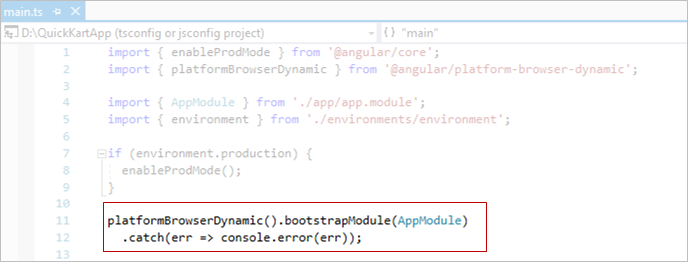
Open the index.html file, observe the code.



**Step 4:**

Similar to all other programs your Angular app also has a main entry point which is the **main.ts** file

You can view the main.ts file in the QuickKartApp-> src folder



The main.ts file bootstraps(starts) the application’s root module i.e. the AppModule

**Step 5:**

AppModule is the root module of the application

If you open the app.module.ts file in the QuickKartApp->src->app folder you'll notice that, AppComponent is added to the **bootstrap** array of the module definition

Hence the AppModule in turn makes a call to the **AppComponent**, which is the root component of the application and makes it ready for use.



The **@NgModule**decorator identifies AppModule as an Angular Module class

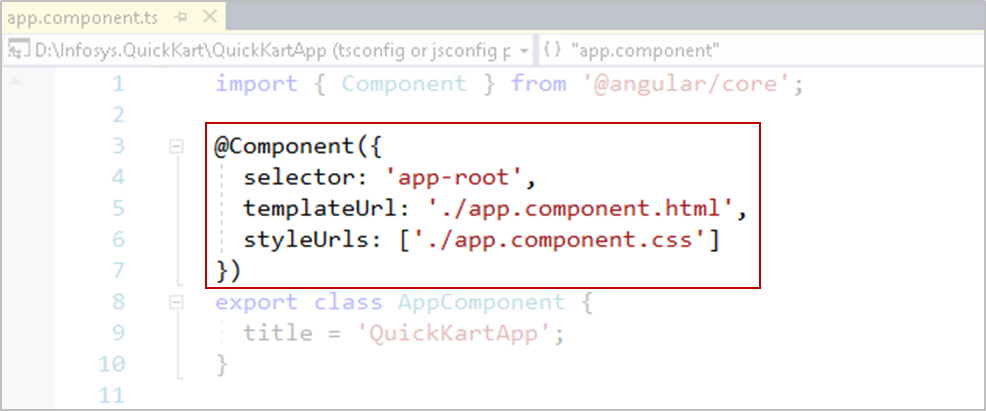
The **declarations**array tells Angular which components belong to AppModule. As you create more components they need to be added to this array else an error message will be displayed while loading the component in the browser.

The **imports**array consists of various modules whose features are used in the application. For example, every application that runs in a browser requires the BrowserModule therefore, it is included in the imports array.

The **bootstrap**array tells Angular which component to execute when the application is launched. Here the AppComponent is bootstrapped when the application launches.

**Step 6:**

Open app.component.ts file in QuickKartApp->src->app folder and observe the code.



**@Component** is a decorator which helps to identify a class as a component.

Observe the selector i.e. **app-root** and templateUrl with a path in AppComponent class.

**selector -**custom tag which is used to render the html of the associated component.

**templateUrl -**contains the path of the html file that is associated with the current component.

**styleUrls -** Specifies the stylesheet file which contains CSS styles to be applied to the template.

**Step 7:**

Once the AppComponent is ready for use. The flow of execution comes to the body tag written in index.html file.

Here it encounters the app-root custom html tag. Modify the code in app-root tag as follows and then run the app once again.

1. <app-root>Loading...</app-root>

This app-root tag is recognized as a selector for AppComponent. Hence the templateUrl ie. app.component.html will get executed.

Angular app will show a small unstyled Loading... at the top left corner of the browser when AppComponent is slowly getting loaded or when there is an error in AppComponent.

And at last **QuickKartApp app is running!** is rendered by the browser.

