**Angular – 4 Documentation**

**Step -1**

Prerequisites:

1. Node

Node comes with a tool called **Node Package Manager** or **NPM**, which is used for installing third-party libraries.

**Verify that you are running at least node**6.9.x**and npm**3.x.x by running node -v and npm -v in a terminal/console window. Older versions produce errors, but newer versions are fine.

## Step -2 : Set up the Development Environment

## npm install –g angular/cli

## Step – 3 : Create new project

## ng new my-app(project name) It will create your new project folder with node-modues and src folder.

## Structure of the Angular projects

* **e2e**: includes end-to-end tests.
* **node\_modules**: all the third-party libraries that our project is dependent upon.
* **src**: the actual source code of our Angular application.

**angular-cli.json:**a configuration file for Angular CLI. We use this file to import third-party stylesheets or define additional environments (eg testing environment) for our application.

**package.json:**a standard file for Node-based projects. It contains metadata about our project, such as its name, version as well as the list of its dependencies.

**protractor.conf.js:**Protractor is a tool for running end-to-end tests for Angular projects. We hardly need to modify this file.

**karma.conf.js:**Karma is a test runner for JavaScript applications. This file contains some configuration for Karma. We rarely need to modify this file.

**tsconfig.json:**includes setting for the TypeScript compiler. Again, we hardly, if ever, need to modify this file.

**tslint.json:**includes the settings for TSLint which is a popular tool for linting TypeScript code. That means it checks the quality of our TypeScript code based on a few configurable parameters.

## 

## Inside src folder

## 

## Step – 4 : Serve the application

## Go to the project directory and launch the server.

## ng serve –o

## Using the --open (or just -o) option will automatically open your browser on <http://localhost:4200/>

## Main.ts is the file which starts the project or compile first

## Browser understands ECMA 5 but the development uses typescript ECMA 6, to fill the gap between these polyfills.ts will be used

## In order to add bootstrap and font-awesome to your project

1. **add dependencies in package.json under “**devDependencies**”**

"font-awesome": "^4.7.0",

"bootstrap": "^3.3.7"

1. In .angular.cli.json under "styles"

"styles": [

"styles.css",

"../node\_modules/bootstrap/dist/css/bootstrap.css",

"../node\_modules/font-awesome/css/font-awesome.css"

]

And then do npm install.

## Generating a Component Using Angular CLI

## ng g c product(component name)

## 

## class is a fundamental building block of many object-oriented programming languages. It’s a container for a bunch of related functions and variables.

**Constructor** is a reserved keyword in TypeScript. A method by that name is a special method in a class. This method is called automatically when we create an instance of that class.

**ngOnInit** is a special method in Angular. Angular calls this method when it creates an instance of this component and displays it to the user in the browser.

## Property binding – [] used for the attribute associated with html element

## Ex: <img [src]=’img’>

## Declaring variables through class or by interface

## Interface user { username:string;}

## <unput type=”text” #cname>

## #cname – local variable

## Visual studio code - <https://code.visualstudio.com/docs/?dv=winzip>

## Angular CLI - <https://www.npmjs.com/package/angular-cli>

## Routing - <https://code.tutsplus.com/tutorials/beginners-guide-to-angular-4-routing--cms-29676>

## Table - <https://www.npmjs.com/package/ngx-datatable>

## Form vlaidations:

## Template driven forms: directives clubbed with html

## Ng-touched : control has been visited : enabled

## Ng-untouched : control has not been visited : disabled

## Ng-dirty : value is changed

## Ng-pristine : value is new

## Ng-valid : control is valid

## Ng-invalid : control is In valid

## ngForm : exposes the entire form into the local variable

## ngModel : exposes the input control to the local var

**HTTP Module:** Async calls

Observables [RxJs: Reactive extensions for java script: external JS library]

Sequence of event(s) [array] that will arrive in future

News paper agency -> News center(web api/DB)

Use the news available

[conversations in appropriate (news format) <- (news) : formays / real time]

As an end customer we have subscribe to the agency

Component(Customer : UI) -> angular service(HTTP Protocol : REQUEST) -> web api / DB

Component(subscribe) <- convert into JSON format <- HTTP RESPONSE : real time

**Life cycle**

These emthods will be called once in lifecycle

**Init()** – called after the component

**Destroy()**

**OnChanges()** – invoked in parent and child

**Common Angular Errors**

## Error 1: ‘ng’ is not recognized

When creating a new project with Angular CLI, you may receive the following error:

'ng' is not recognized as an internal or external command.

This error is simply telling you that Angular CLI is either not installed or not added to the PATH. To solve this error, first, make sure you’re running **Node 6.9** or higher. A lot of errors can be resolved by simply upgrading your Node to the latest stable version.

Open up the Terminal on macOS/Linux or Command Prompt on Windows and run the following command to find out the version of Node you are running:

node --version

If you’re running an earlier version of Node, head over to [nodejs.org](http://nodejs.org/) and download the installer for the latest stable version.

Once you have installed Node 6.9+, you need to install Angular CLI globally:

npm install -g @angular/cli

Note the -g flag here. This tells NPM that you want to install this package globally so you can run it from any folders on your machine.