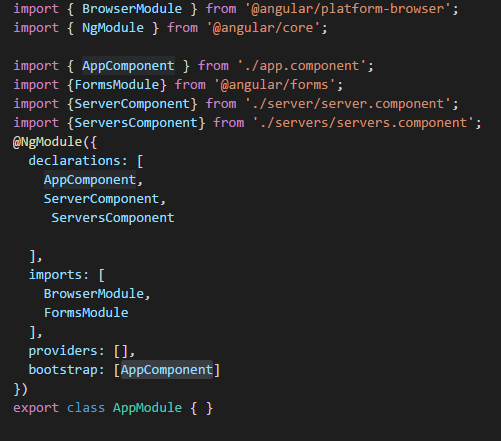
Angular 2 from Udemy

1. @NgComponents:-



1. npm install -g @angular-cli  
   ng new project-name

ng g c nameofComponent

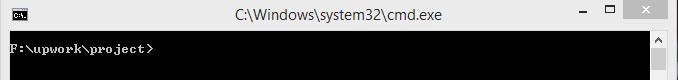
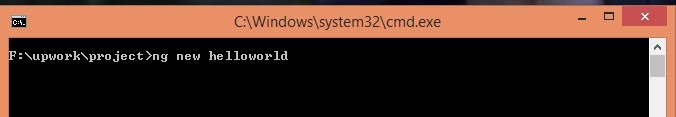
ng g component employee-list

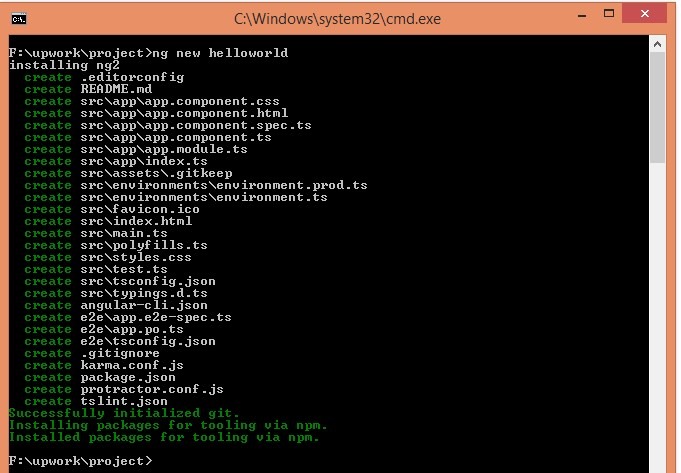
Service Creation:

ng generate service hero

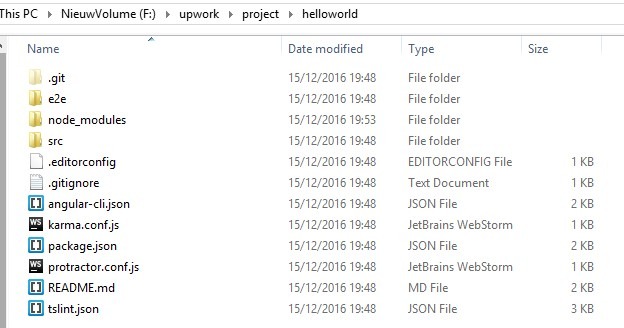
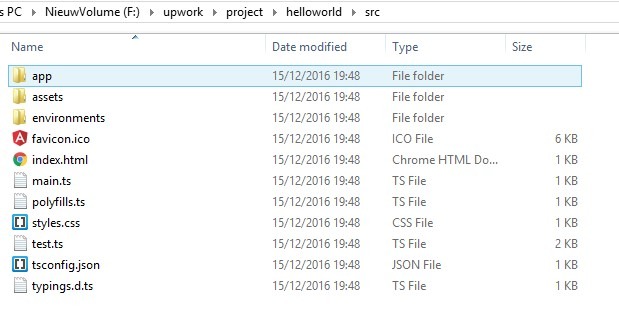
**How to develop your first Angularjs 2 application in steps to render simple text?**

We are going to follow a step by step approach to develop our first Angular2 Application.

1. Open the command prompt on the workspace folder called ‘project’.
2. 
3. Run command **‘ng new <project name>’**. Lets suppose the project name is ‘helloworld’. This command will create a new project named ‘helloworld’. This may take few minutes.
4. 

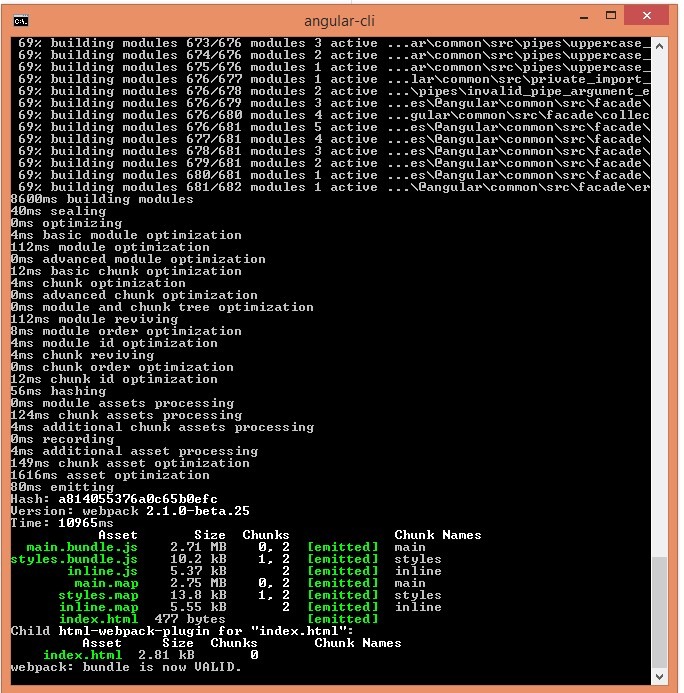
This command will create the main app component and relevant template along with typings, tsconfig, styles and e2e configuration.

This step will also install all necessary required dependency library for the new project.Inside ‘helloworld’ project folder, there will be these following files created.

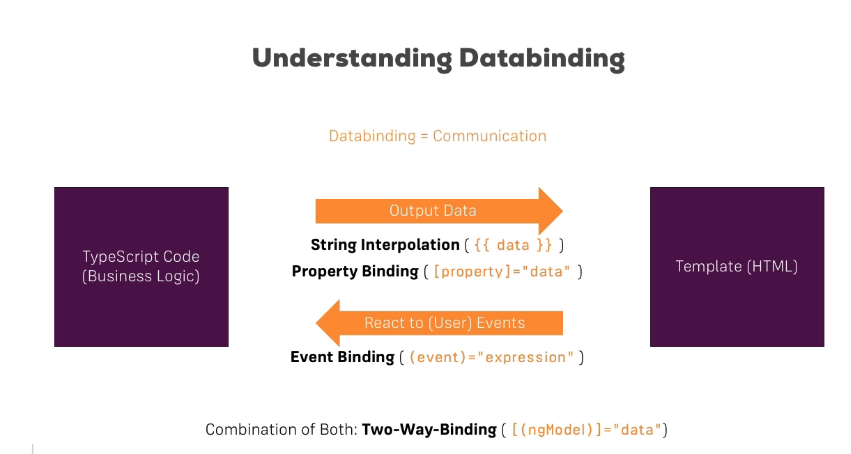
  
The source code for the application are inside ‘src’ folder as following:Index.html is the main html file for the application.

Therefore, in app.component.ts **<app-root>** has been described as the main selector.  
Now as we want to render a simple text. So in our AppComponent, we have taken a variable named ‘title’ and the template url of AppComponent is app.component.html.Next step is to render title in app.component.html as

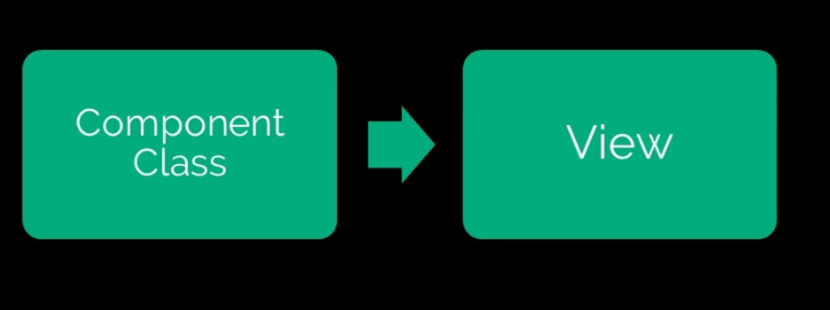
|  |  |
| --- | --- |
| 1  2  3 | <h1>     {{title}}  </h1> |

1. The next step is to run the project with the command **‘ng serve’.**a. Go to the ‘helloworld’ folder by running  ‘cd helloworld’ command  
   b. Run ‘ng serve’ in command prompt
2. From browser run ‘http://localhost:4200’and the output would be as following:

* **Data Binding**

****

* **Property binding ([]):**

****

Defining element property values in the view from the component class.

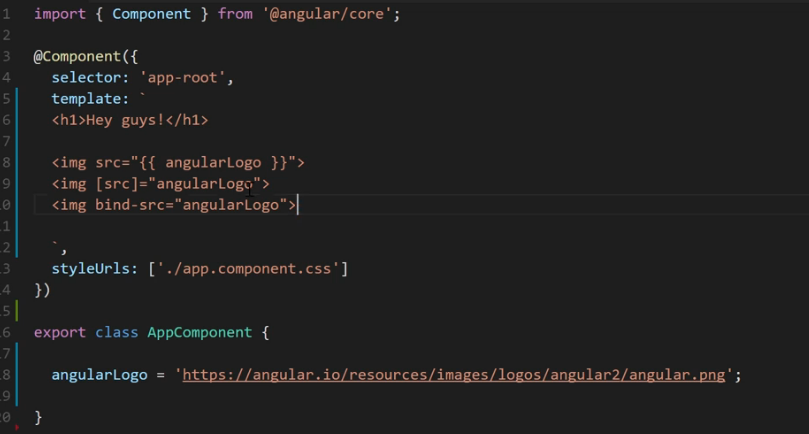
Property binding is one way, in that the data is transferred **from** the component **to** the class.

Eg: <button [disabled]=”buttonVal”>MyButton</button>

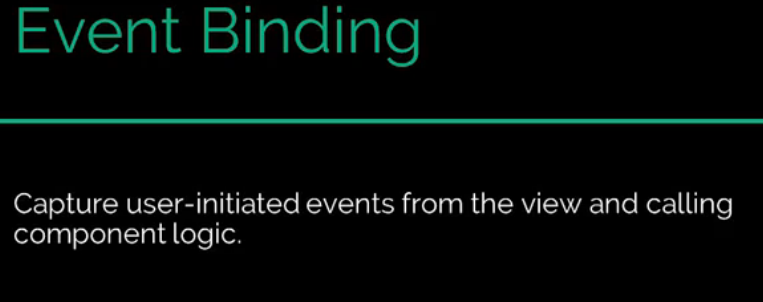
Class Component{

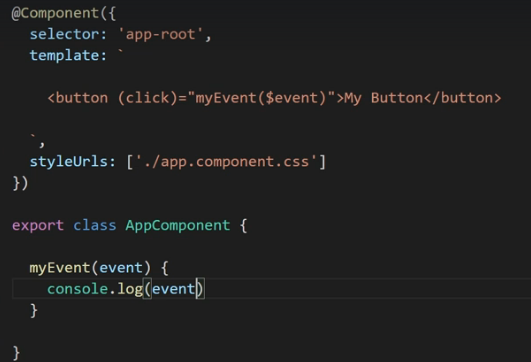
buttonVal=true;

}

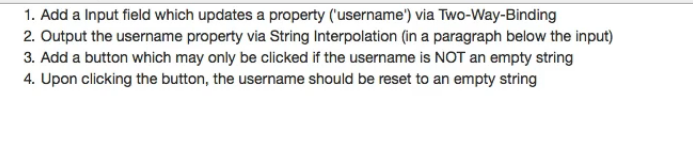


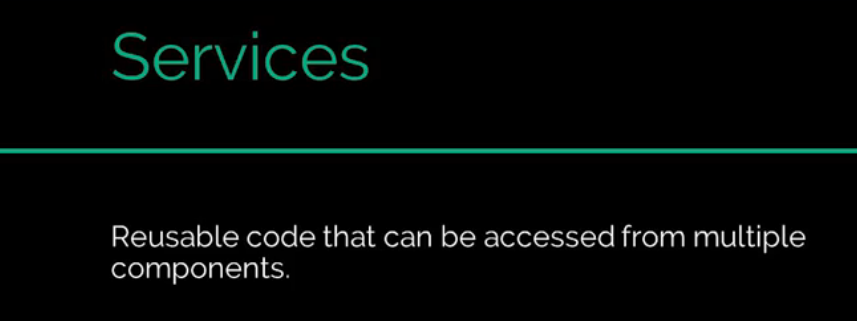
* Event Binding:



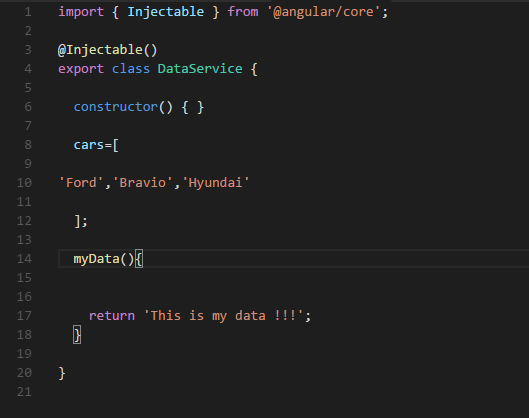


1.Assignments:-

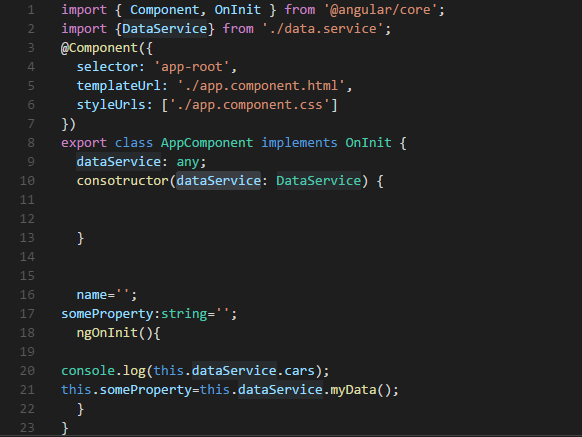


**Services:** 

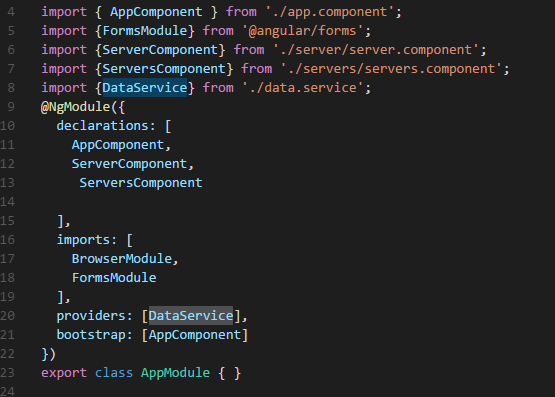
Ex: **data.Service.ts**



**App.component.ts**



App.module.ts



## What is a Stream?

A stream in the RxJS world simply represents **values over time**. Users sending chat messages, a user clicking around on a page, a user filling out different formfields in a form; these all represent the basic concept of values (or events) that take place over a period of time.

## Observables, Observers & Subscriptions

* An observable is a function that produces a stream of values to an observer over time.
* When you subscribe to an observable, you are an observer.
* An observable can have multiple observers.

## Let's Create an Observable

In the project we created from the previous tutorial, open up /src/code.ts and specify the following:

import { Observable } from "rxjs/Observable";

var observable = Observable.create();