Agenda

- What is Angular?
- Angular **(vs)** Angular2 **(vs)** Angular4
- Setup for local environment
- What is Typescript?

What is Angular?

- Angular is a very powerful open source and absolutely free **JavaScript library**.
- Angular is a framework for building client applications in HTML and either JavaScript or a language like Typescript that compiles to JavaScript.
- AngularJS is a **structural framework** for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly.
- Angular's **data binding and dependency injection** eliminate much of the code you would otherwise have to write. And it handles all of the **DOM** and **AJAX** code you once wrote by hand and puts it in a **well-defined structure**.
- History: AJAX → jQuery / Prototype → MVC Frameworks (Knockout / Backbone) → AngularJS1.X → Angular2 → Angular4.
- Even though Google is working hard on Angular 2.0, there's a dedicated team working full-time on Angular 1.X and regularly providing updates too.
- Features of AngularJS / Angular4
  - It's based on **Model View Controller** design pattern.
  - It's an **extension of HTML DOM** and has additional attributes which reduces code and makes it easy to program dynamic and responsive applications.
  - Its features include everything we need to **build a CRUD** app: data-binding, basic templating, directives, form validation, routing, deep-linking, reusable components, dependency injection.
  - Angular is built around the belief that **declarative code** is better than imperative when it comes to building UIs and wiring software components together, while imperative code is excellent for expressing business logic.
  - Its views are pure html pages, and controllers written in JavaScript do the business processing.
  - AngularJS applications can run on all major browsers and smart phones, including Android and IOS based phones/tablets.
  - It's majorly used for developing **Single Page Applications** (SPA).
- Over and above AngularJS 1.X, Angular providers
  1. Code is simpler to write and read.
  2. Better performance
  3. Improved Modularity

4. Improved dependency injection
5. Better support for native mobile – iOS and Android.
6. Improved testability

- It’s a bit complicated to learn when compared to Angular 1 but is easy to implement.
- Its built using TypeScript (Typed JavaScript) and hence TypeScript is ideal choice for programming Angular2 applications.
- It uses component based approach.

Angular1.X (vs) Angular2 (skip if new to Angular)

First thing to be clear is **Angular2** is not an upgrade of **Angular,** it is completely different from **angular1.X.**

Let us see some of the comparisons:

Angular JS 1.X	Angular 2
Setting up application is very easy, need to just add the reference of angular.js	Setting up application is quite complex, and it takes lot of work because it’s dependent on **Gulp** or **Grunt.**
Angular JS is not built with mobile support in mind.	Whereas Angular2 was built for specially taking care of mobile support, so it’s mobile oriented.
Angular JS core concept was **$scope,** without scope we can’t communicate between views and controller.	Angular2 is using **Zone.js** for detecting changes made in the **DOM.**
For communicating with **views/templates** it require **controllers**.	It’s build with **components**. **Template Directives + controllers = components**
Performance is less when compared to angular2.	Unidirectional tree based change detection increases its performance. As per **ng-conf** angular 2 is **five times faster than angular1.**
Angular1.X has **ES5, ES6 and Dart**	Whereas angular2 has more choice, it can use any of the languages from **ES5, ES6, and typescript/dart.**
Ng-repeat is used for iterating through list of objects.	Structural directives syntax is completely changed. **Ng-repeat** is replaced with ***ngFor**
Mostly **built-in directives** are used for events and element properties. Ex: **ng-src, ng-href, ng-click** etc.	It directly uses the HTML DOM element properties. Ex: **(click)**
Ng-bind is used for one-way binding	Whereas **ng-bind is replaced with [property],** where property is HTML DOM element property.
It has two ways to bootstrap the application, one is **automatic bootstrapping (ng-app) and manual bootstrapping.**	It has **only one way** which is manual bootstrapping.

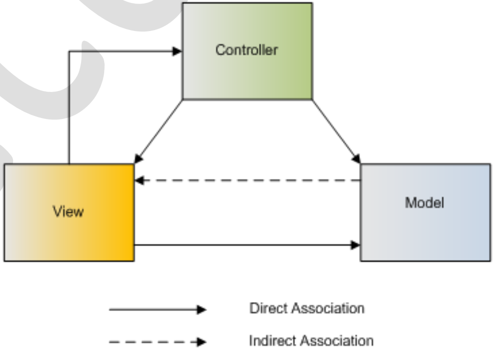
For routing it uses **$routeProvider.when()**	It uses **@RouteConfig {{…}}**

Angular2 (vs) Angular4 (skip if new to Angular)

Angular 2 and Angular 4 Doesn't have much differences, it's just an update for the features that are existed in angular2.

Angular2	Angular4 / Just Angular
Angular2 application **size is larger** when compared to angular4.	Angular4 application is **smaller and faster** when compared to angular2 app.
***ngIf** is not build with if/else syntax	***ngIf** has **if/else** syntax that make sense when you are making ajax calls we can show loading within the page.
For changing a text to title case, we need to write our own logic.	Angular 4 has introduced a new '**titlecase'** pipe for converting first letter of each word to uppercase **Ex: <h1>{{'hello world' | titlecase}}</h1>**
For creating templates we need to use template property	New directive **ng-template** has introduced.
For animations need to import multiple packages and apply transactions	For animations we now have own package **@angular/platform-browser/animations**
	Generating Source maps when there is an error in the template with a meaningful context.

What is MVC Pattern?



Model: The model is responsible for managing application data. It responds to the request from view and to the instructions from controller to update itself.

View: A presentation of data in a particular format, triggered by the controller's decision to present the data.

Controller: The controller responds to user input and performs interactions on the data model objects. The controller receives input, validates it, and then performs business operations that modify the state of the data model.

Setup for Local Development Environment

Setting up **Angular2/Angular4** is a quite complex, it needs a lot of work when compared with the installation for **Angular1.X** applications.

To setup with the quick start files follow these steps:

Step1: Install Node JS and npm

Step2: Install visual studio

Step3: Configuring External Web Tools.

Step4: Install Typescript

Step5: Download quick start files

Step6: Create ASP.NET Project

Step7: Copy the quick start files into your ASP.NET Project

Step8: Restore Packages

Step9: Build and Run

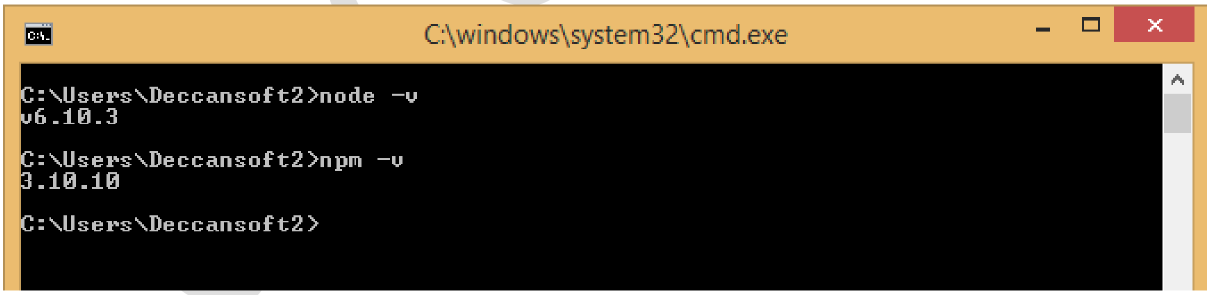
Install Node JS and npm

Node.js and npm are essential to modern web development with Angular and other platforms. Node powers client development and build tools. The *npm* package manager, itself a *node* application, installs JavaScript libraries.

- To work with **Angular2/Angular4** you must have **Node JS version 4.6.X or higher and npm 3.X.X or higher**. Older versions produce errors.
- Before installing make sure whether you have already installed Node JS before. For **checking the Node JS and npm versions** follow these commands in command prompt.

  For Node JS: **node –v**

  For npm: **npm –v**



```
C:\Users\Deccansoft2>node -v
v6.10.3

C:\Users\Deccansoft2>npm -v
3.10.10

C:\Users\Deccansoft2>
```

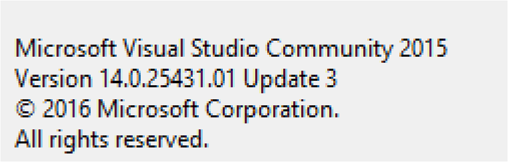
- If the **PC** don't have Node JS installed it will show you like **"node" is not recognized as an internal or external command**, **"npm" is not recognized as an internal or external command**. Or if your PC has older version than the required please update to the latest versions.
- For installing latest version of Node JS and npm go to https://nodejs.org/en/download/

Install Visual Studio

Minimum requirement for developing Angular2/ Angular4 applications is **Visual Studio Update3** because the earlier versions do not follow the best practice for developing applications with Typescript.

- To check the version of Visual Studio 2015

  Open Visual Studio 2015 → go to **Help** → click on **About Microsoft Visual Studio**



Microsoft Visual Studio Community 2015
Version 14.0.25431.01 Update 3
© 2016 Microsoft Corporation.
All rights reserved.

- If you don’t have **update3** go to the following links (or) you can directly install from visual studio go to **Tools** → **Extensions and Updates**

Visual Studio 2017 ships with a built-in version of **TypeScript**.

Configuring External Web Tools

Configure Visual Studio to use the global external web tools instead of the tools that ship with Visual Studio:

a. Open the **Options** dialog with `Tools` | `Options`

b. In the tree on the left, select `Projects and Solutions` → Web Package Management → `External Web Tools`.

c. On the right, move the `$(PATH)` entry above the `$(DevEnvDir)` entries. This tells Visual Studio to use the external tools (such as npm) found in the global path before using its own version of the external tools.

d. Click OK to close the dialog.

e. Restart Visual Studio for this change to take effect.

Note: Visual Studio now looks first for external tools in the current workspace and if it doesn't find them, it looks in the global path. If Visual Studio doesn't find them in either location, it will use its own versions of the tools.

Install Typescript 2 + for Visual Studio:

To develop Angular applications we need **Typescript 2.2.0 or its higher versions**, as Visual studio Update3 doesn’t have such specifications we need to install it externally.

Before installation we can check our Typescript version in the Visual Studio

Open Visual Studio 2015 → go to **Help** → click on **About Microsoft Visual Studio**

- To get the latest version of **Typescript for Visual Studio 2015**
  - https://www.microsoft.com/en-us/download/details.aspx?id=48593
- To get the latest version of **Typescript for Visual Studio 2017**
  - https://www.microsoft.com/en-us/download/details.aspx?id=55258

Download Quick Start Files

You can download the quick start files from Git hub https://github.com/angular/quickstart and then extract the files.

1. Download QuickStart seed and restore in d:\QuickStartSeed folder.

   Download Link: https://github.com/angular/quickstart/archive/master.zip

2. Copy Seed to d:\...\DemoProject folder
3. Open Command window and execute the following commands in above folder
   a. npm install
   b. npm start

Note that the browser is auto started and index.html output is displayed.

Create ASP.NET Project

1. Create a New ASP.NET MVC Project
   a. In Visual Studio, select `File` | `New` | `Project` from the menu.
   b. In the template tree, select `Templates` | `Visual C#` (or `Visual Basic`) | `Web`.
   c. Select the `ASP.NET Web Application` template, give the project a name, and click OK.
   d. Select the desired ASP.NET 4.5.2 template and click OK.
2. Copy the QuickStart files into the ASP.NET project folder
   a. Copy the QuickStart files we downloaded from github into the folder containing the `.csproj` file.
   b. Include the files in the Visual Studio project as follows:
   c. Copy the quick start files into your project, but all the files are not necessary to be included into your project. So copy only selected files from the extracted folder.

      Copy the following files/folders into your **.csproj**

      - **src**
      - **bs-config.json**
      - **package.json**
      - **tslint.json**

Restore the packages required for your application:

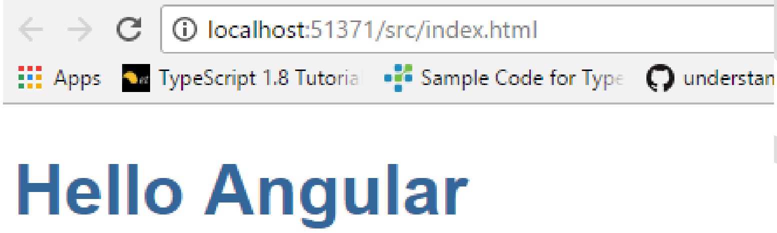
- For restoring packages right click on **package.json** → select **Restore Packages.** This uses **npm** to install all the packages defined in **package.json.**

  In between you can open the Output window to watch the npm commands that are executed. Simply we can **ignore the warnings**.

Note: After you see the message **Installing Packages Complete, DO NOT** include the `node_modules` folder in the project. Let it be a hidden project folder.

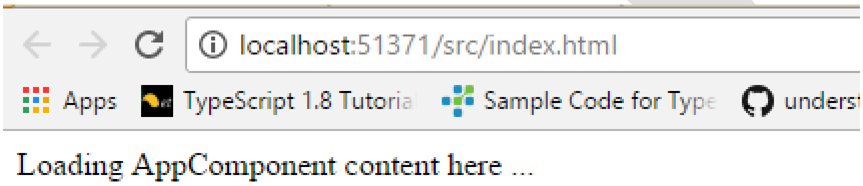
Build and Run

- After Installing packages **build the Application.**
- To run the application open **Command Prompt → go to your Project Directory →** execute command **npm start.**
- When this command is executed,
  - ➔ firstly it will **launch typescript compiler** and compiles the application with the following command **tsc –p src/**
  - ➔ Now it will start the **lite-server** and launches the browser where you can see the Output: **Hello Angular**



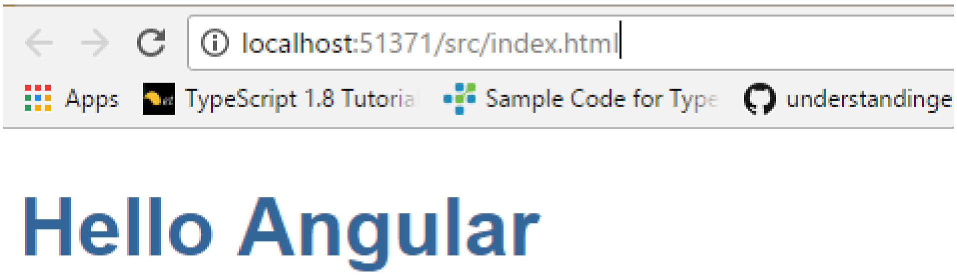
Hello Angular

- But if you run the **index.html** by pressing **F5 in visual studio** it will not show the same output.



Loading AppComponent content here ...

- **To Run in VS with F5 follow these steps**
  1. In \src**index.html**, change **base href** from **<base href="/"> to <base href="/src/">.**
  2. Also in **index.html**, change the scripts from **node_modules** to **/node_modules** with a slash
  3. In **src/systemjs.config.js**, near the top of the file, change the '**npm:**' '**node_modules/**' to **path** '**npm:**' '**/node_modules/** (with a slash).
  4. After these changes npm start command no longer works, you must choose to configure either for **F5 with IIS** or for **npm start with the lite-server**.
  5. Now run the application using F5



Hello Angular

What is Typescript?

1. Typescript is a typed superset of JavaScript. JavaScript that scales. It's not replacement of JavaScript nor does it add any new feature of JavaScript.
2. It's **compiled** to generate JavaScript.
3. It was designed by **Anders Hejlsberg** (founder of C#) at Microsoft. Its open source and can be used in any place where we would need JavaScript.
4. Its object oriented and supports core features like interfaces and classes.
5. Typescript generated JavaScript can reuse all of the existing JavaScript frameworks, tools, and libraries.
6. Also any valid **.js** file can be renamed to **.ts** and compiled with other Typescript files.
7. The reader of this tutorial needs to have good knowledge on **Object Oriented Programming** and basic knowledge on **JavaScript**.
8. It's **not mandatory** to strongly type everything when we are type scripting.
9. There are almost 40 languages which are superset of JavaScript. On which they generate **.js** on compiling, typescript is one of these languages.

   Few of the languages are: **typescript, purescript, coffeescript, livescript**

Versions:

1. Typescript 0.8 was released on Oct 2012
2. Typescript 0.9 was released in 2013
3. Typescript 1.0 was released in 2014
4. Typescript 2.0 was released on Sep 22 2016

Benefits of TypeScript:

1. It compiles the code and **generate syntax errors** if any. This helps to highlight errors before the script is executed.
2. Because of Object Oriented features, it's **reusable and easier to manage** in large and complicated projects.
3. **Angular 2 framework is written in Typescript** and it's recommended that developers use the language in their projects as well.
4. Due to the static typing, code written in Typescript is more predictable, and is generally **easier to debug**.

Small example:

Ex: Hello world

```
function sample() {
  var message: string = "Hello world"
console.log(message);
}
sample();
```