Introduction to Angular: Objectives and Outcomes

In this lesson you will be given a quick overview of JavaScript frameworks and then introduced to Angular in particular. We will learn some basics of Angular and how to configure an Angular application using the Angular-CLI, the command line tool. At the end of this lesson, you will be able to:

- Get a basic overview of JavaScript frameworks
- Understand the architecture of an Angular application
- Scaffold out a starter Angular application using *angular-cli*, the command line tool

Exercise (Instructions): Getting Started with Angular

Objectives and Outcomes

In this first Angular exercise, you will first install *angular-cli*, the command line tool for scaffolding Angular applications. You will then use the tool to scaffold out a basic Angular application. We will thereafter develop this application into a full-fledged Angular application in the process of doing the exercises in this course. At the end of this exercise you will be able to:

- Install *angular-cli*
- Scaffold out a basic Angular application

Installing *Angular-CLI*

From the Angular-CLI documentation we learn that the Angular CLI makes it easy to create an application that already works, right out of the box. It already follows the best practices suggested by the Angular community!

- To install *angular-cli* globally, type the following at the prompt:

```
npm install -g @angular/cli
```

Use *sudo* on a Mac and Linux

- This will make the command line tool for creating Angular applications. To learn more about the various commands that this CLI provides, type at the prompt:

```
ng help
```

Generating and Serving an Angular Project using Angular-CLI

- At a convenient location on your computer, create a folder named *Angular* and move into that folder.
- Then type the following at the prompt to create a new Angular application named *conFusion*:

```
ng new conFusion -dir=<The path of your Angular folder>/conFusion --style=scss
```

- This should create a new folder named *conFusion* within your *Angular* folder and create the Angular application in that folder.
- Move to the conFusion folder and type the following at the prompt:

```
npm install
ng serve --open
```

- This will compile the project and then open a tab in your default browser at the address http://localhost:4200.
- You can initialize your project to be a Git repository by typing the following commands at the prompt:

```
git init
git add .
git commit -m "Initial Setup"
```

Note: Some of you may find that Angular CLI automatically does the first commit on your computer and initializes the Git repository. Please do a "git status" in the project directory to check if an automatic commit has been done. This doesn't happen on my computer. Hence the above instructions.

- Thereafter you can set up an online Git repository and synchronize your project to the online repository. Make sure that the online Git repository is a ***private*** repository.

Conclusions

In this exercise you installed the Angular CLI tool and created a basic Angular project and served up the compiled project to your browser.

✓ Completado

Exercise (Instructions): Configuring your Angular Application

Objectives and Outcomes

In this exercise we will set up our project to use Angular Material and Angular Flex Layout. We will then introduce our first Angular Material component into our application. At the end of this exercise you will be able to:

- Configure your Angular project to use Angular Material and Flex Layout.
- Start using Material components in your application.

Configure your Angular Project to use Angular Material

Note: This course is designed with Angular Material Beta.3. Before you proceed forward, you may wish to read the detailed information posted in https://www.coursera.org/learn/angular/discussions/all/threads/4yxVk7DXEee0mQrUfDuicA where I have clearly explained about dealing with the newer Beta versions of Angular Material (up to Beta.12). I would strongly suggest that to proceed ahead with the course with minimal disruption, please install the Beta.8 version of Angular Material. With this installation, the course instructions will still work as given.

- To configure your project to use Angular material, type the following at the prompt to install Angular Material, Angular Animations and HammerJS:

```
npm install @angular/material@2.0.0-beta.8 --save
npm install @angular/cdk@2.0.0-beta.8 --save
npm install --save @angular/animations
npm install --save hammerjs
```

Configure to use Material Design Icons

- Next, include the following into the <head> of *index.html* to make use of Material Design icons:

```
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel
  ="stylesheet">
```

Configure your Angular Project to use Flex Layout

- Next, install Angular Flex Layout as follows:

```
npm install --save @angular/flex-layout@latest
```

Updating AppModule

- Then, you need to import the Angular Animations Module, Angular Material Module, Flex Layout Module and hammerjs into your root module (src/app/app.module.ts) as follows:

```
. . .

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { MaterialModule } from '@angular/material';
import { FlexLayoutModule } from '@angular/flex-layout';

. . .

import 'hammerjs';

@NgModule({

  . . .

  imports: [

    . . .,

    BrowserAnimationsModule,
    MaterialModule,
    FlexLayoutModule

  ],

    . . .


})

. . .
```

Adding a Material Toolbar

- Open app.component.html and replace its contents with the following code:

```
<md-toolbar color="primary"> <span>Ristorante Con Fusion</span> </md-toolbar>
```

Adding Styles

- Add the following styles to styles.scss file:

```
@import '~@angular/material/prebuilt-themes/deeppurple-amber.css';

// some basic resets

body {
  padding: 0;
  margin: 0;
  font-family: Roboto, sans-serif;

}
```

- This will add a built-in Material theme to our application.
- Do a Git commit with the message "Configuring Angular"

Conclusions

In this exercise we learnt to use Angular Material and Flex Layout NgModules in our Angular application.

Introduction to Angular: Additional Resources

PDFs of Presentations

1-JavaScript-Frameworks.pdf
2-Intro-Angular.pdf
3-Angular-Architecture.pdf

Angular Resources

- Angular.io
- Angular CLI
- Angular Material 2
- Angular Material Toolbar
- Angular Flex Layout
- Angular Flex Layout Documentation

Typescript

- Typescript
- Typescript Tutorial
- Typescript: Migrating from JavaScript

Definitions

- Framework
- Hollywood Principle
- Inversion of Control
- Imperative vs Declarative Programming
- Imperative vs Declarative

Blog Articles

- 5 Best JavaScript Frameworks in 2017
- Top JavaScript Frameworks & Topics to Learn in 2017
- Declarative vs. Imperative Programming for the Web
- Intro to Material (2) Components, Custom Styles, Buttons & Indicators—Using Angular Material (2) Components in your Angular (2) Project

Front-End JavaScript Frameworks: An Introduction

Jogesh K. Muppala

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Why JavaScript Frameworks?

- Complexity of managing DOM manipulation and data updates manually
- Well defined application architectures:
  - Model View Controller / Model View View Model / Model View Whatever
  - Binding of model and view: controllers, view models

Software Library

- Collection of implementations of behavior with a well-defined interface by which the behavior is invoked
- Reuse of behavior
- Modularity
- E.g., jQuery

https://en.wikipedia.org/wiki/Library_(computing)

Software Framework

- Abstraction in which software provides generic functionality that can be selectively changed by additional user-written code
- Universal, reusable environment that provides particular functionality as part of a larger software platform
- E.g., Angular, Ember, Backbone

https://en.wikipedia.org/wiki/Software_framework

Library vs Framework

- The following borrowed from AngularJS documentation makes the distinction clear:
  - **a library** - a collection of functions which are useful when writing web apps. Your code is in charge and it calls into the library when it sees fit. E.g., jQuery.
  - **frameworks** - a particular implementation of a web application, where your code fills in the details. The framework is in charge and it calls into your code when it needs something app specific. E.g., Angular, Ember, etc.

https://docs.angularjs.org/guide/introduction

Framework

- Hollywood Principle
  - Don’t call us, we’ll call you!
- Inversion of Control
- Imperative vs Declarative Programming

JavaScript Frameworks

- Single Page Application
  - Rich Internet Applications
- Model-View-Controller (MVC) / Model-View-ViewModel (MVVM) / Model-View-Whatever
  - Data binding, routing
- Scalable, Reusable, Maintanable JS code
- Test driven development

JavaScript Frameworks

- Angular
- Ember
- Backbone
- React
- Aurelia
- Meteor
- Polymer
- Knockout
- Vue
- Mercury

Introduction to Angular

Jogesh K. Muppala

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

Angular History

- AngularJS
  - Designed by Misko Hevery
  - First released in June 2012
  - One of the most popular front-end JS frameworks
- Angular
  - Complete rewrite of the framework
  - Component-based
  - Mobile Support
  - Server-side rendering
  - Powerful templates

What is Angular?

- Structural framework for dynamic web applications:
  - HTML only does static documents
  - Angular fills in the gap to support dynamic applications
    - Solving the impedance mismatch
    - Designed with CRUD applications (data-driven) in mind
    - Declarative approach

Angular Vocabulary

- One-way/Two-way Data Binding
- Components
- Directives
- Templates
- Routing
- Modules
- Service
- Provider
- Testing

Angular

- Is it Angular 2 or just Angular?
- Semantic Versioning
  - <Major Version>.<Minor Version>.<Patch>

Exercise: Getting Started with Angular

- Install *angular-cli*
- Scaffold out a basic Angular application

Angular Application Architecture Overview

Jogesh K. Muppala

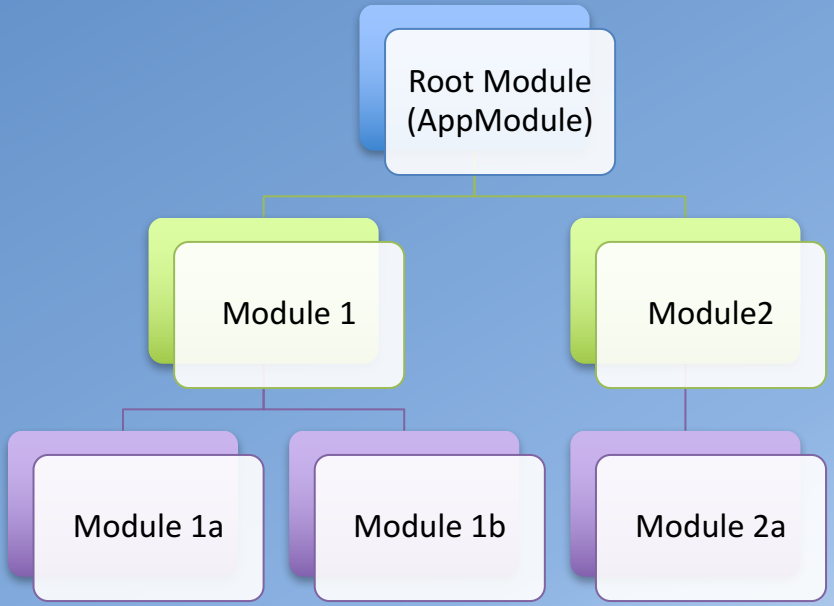
THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

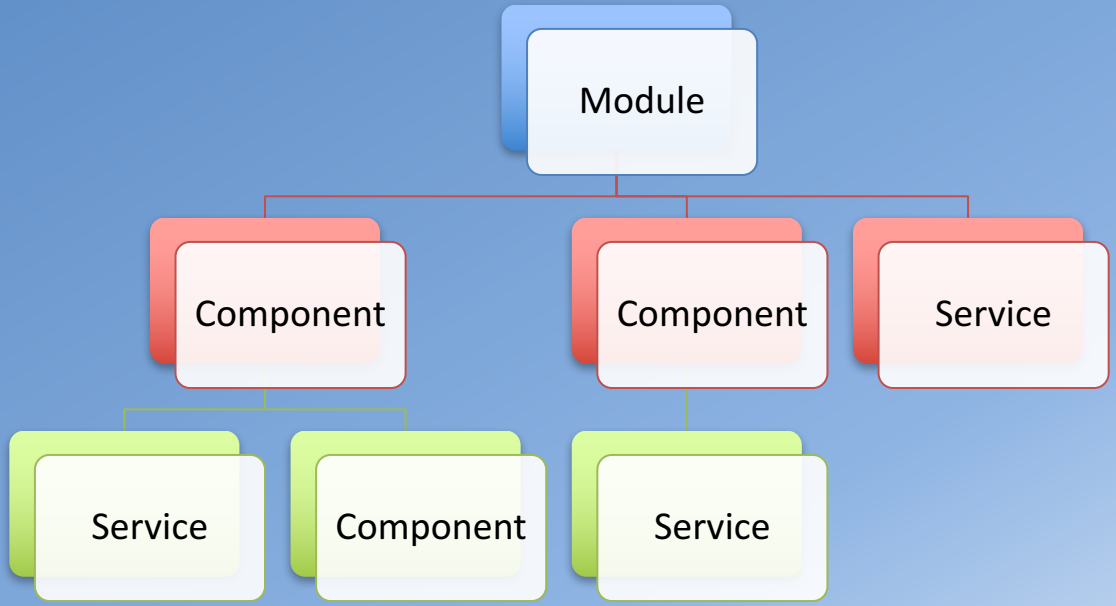
Angular Architecture

- Modular
- Component-based with Templates
- Services

Modular Architecture



Module


Module
Component
Component
Service
Service
Component
Service