Building Software Systems

Lecture 4.2

Introduction to Angular

SAURABH SRIVASTAVA

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

IIT (ISM) DHANBAD

What is Angular?

Angular is a platform/framework for building single-page client applications using HTML and TypeScript

- It is developed and maintained by Google
- It utilizes a hierarchy of components as the primary method for building applications
- Each component encapsulates the logic, template, and data specific to a part of the user interface

Angular enables developers to build applications for any deployment target

- For example, for web, mobile web, native mobile, and native desktop applications
- Angular provides the underlying base for Ionic Framework
- Angular's architecture allows Ionic apps to share a single codebase across multiple platforms
- In Ionic apps, the user interface (UI) is designed using Angular components
- Ionic developers can utilize Angular's features such as modules, data binding and services

We will only be covering Angular as a pre-requisite for Ionic

- Angular itself is quite vast and complex, and it is not in our current scope to cover it in detail

Angular Components and Templates

Components are the building blocks of an Angular application

- Each component controls a patch of screen called a view
- Components encapsulate the data, HTML template, and behaviour into a single reusable unit
- They can be reused throughout the application, promoting code efficiency and consistency
- Components are TypeScript classes decorated with the `@Component` TypeScript decorator
- The decorator can be used to provide information about the component such as selector, template, styles, etc.

Templates are HTML views with Angular directives and binding markup

- They allow displaying information from the components and listen for component events
- They can contain standard HTML elements such as `<div>`, `<span>`, `<p>`, `<h1>` etc.
- Angular supports interpolation using double curly braces `{{ }}` to dynamically display component data
- Property binding allows you to set an element's property to the value of a component's property
- Event binding allows you to listen to events triggered by user interactions (e.g., click, hover) …
- …. and execute methods in the component class

An Example Angular Component

Here,

- The class `GreetingsComponent` is a component
- Its selector is `app-greetings`
- Its template is defined in the file called `greetings.component.html`
- Its styles are defined in the file called `greetings.component.css`
- It has a property/field called `greetingMessage`
- There are two methods – a constructor and a method called `getGreetingBasedOnTime`

```typescript
import { Component } from '@angular/core';

@Component({
  selector: 'app-greetings',
  templateUrl: './greetings.component.html',
  styleUrls: ['./greetings.component.css']
})
export class GreetingsComponent {
  greetingMessage: string;

  constructor() {
    this.greetingMessage = this.getGreetingBasedOnTime();
  }

  getGreetingBasedOnTime(): string {
    const hour = new Date().getHours();
    if (hour < 12) {
      return 'Good Morning';
    } else if (hour < 15) {
      return 'Good Noon';
    } else if (hour < 18) {
      return 'Good Afternoon';
    } else {
      return 'Good Evening';
    }
  }
}
```

An Example Angular Template

Here,

- We the value of the property `greetingMessage` will be pasted inside an `<h1>` tag

```html
<!-- greetings.component.html -->
<div>
  <h1>{{ greetingMessage }}</h1>
</div>
```

Setting up Angular on your Machine

Install Node first

- If you followed the session on ReactJS, you should have `npm` and `ng` already setup on it
- However, the node version we used that time was not an LTS version
- Angular only works with LTS versions
- You may setup an LTS version by downloading and installing node from here: https://nodejs.org/en/download/
- Pick a suitable binary, e.g., *Linux Binaries (x64)*, and then go to: https://github.com/nodejs/help/wiki/Installation#how-to-install-nodejs-via-binary-archive-on-linux

Next, setup Angular

- Use the following command to do so:
  ```
  npm install -g @angular/cli
  ```

Going through the Tutorial Application

It is recommended that you follow the first tutorial of Angular to begin

- Download and extract the content of this app in a directory: https://angular.io/generated/zips/first-app-lesson-00/first-app-lesson-00.zip
- Install the app with its dependencies by issuing the command in the directory:
  `npm install --force`
- There may be a number of warnings, just ignore them
- To check if things are fine, run the following command:
  `ng serve --host=0.0.0.0 --disable-host-check`
- If everything works fine, you will see a URL to open in your browser on the CLI
- Open the same in the browser, it should display a simple page showing **Default** as content
- It may be better to go ahead with the full tutorial, but we will not do so here

Check out the Video Resources for the Lecture

https://drive.google.com/drive/folders/1oCh4z0Ntqo-SyBNvun6jDcdZLn1ulvA1?usp=drive_link

Check out the Video Recording of the Lecture

https://www.youtube.com/watch?v=dyIm0SNBG98