Angular Workshop Overview

Overview

Angular is one of the most popular client-side JavaScript frameworks. It is used to create dynamic, interactive and responsive cross platform applications. It is a full-featured Single Page Application (SPA) framework.

This workshop is designed for software professionals who want to learn the basics of Angular (version 8) and its building blocks in simple and easy steps. It follows a hands-on approach. It is structured around a small sample application. Different concepts will be explained in detail as they are introduced in the application.

Participants' Profile

The participant should have a good working knowledge of HTML, CSS and JavaScript. Knowledge of Bootstrap is a plus, but not mandatory.

Benefits

At the end of this course, the participant will:

- Understand the key building blocks of an Angular application
- Learn to build interactive, single page applications using Angular
- Understand and appreciate the application of emerging concepts like MVC, MVVM, DI, REST, etc.
- Be able to use various Angular features including modules, components, directives, services, pipes and routers

Topics Covered

1. Need for frameworks (Day 1)
2. Introducing Angular (Day 1)
3. Architecture overview (Day 1)
4. TypeScript (Day 1)
5. Setting up Development Environment (Day 2)
6. Components & Templates (Day 2)
7. Data Binding (Day 2)
8. Directives (Day 3)
9. Pipes (Day 3)
10. Services & Dependency Injection (Day 3)
11. Introduction to Unit Testing using Jasmine & Karma (Day 4)
12. Understanding Observables (Day 4)
13. Forms & Validation (Day 4)
14. Building Single Page Apps using Routing (Day 5)

15. Server Communication using HttpClient (Day 5)

Software Requirements

1) Node.js (https://nodejs.org/en/)
   a) Required for installing JSON Server and Angular CLI mentioned below
2) Angular CLI (https://cli.angular.io/)
   a) A command line interface for Angular
3) JSON Server (https://www.npmjs.com/package/json-server)
   a) Allows us to expose JSON data as REST API
   b) This is required for demonstrating client-server communication
   c) Install it globally using "npm install -g json-server" command
   d) Check the URL for more information
4) Code Editor (any one)
   a) Visual Studio Code (https://code.visualstudio.com/)
   b) Sublime Text (https://www.sublimetext.com/)
   c) Brackets (http://brackets.io/)
   d) Atom (https://atom.io/)
5) Browser - Google Chrome
   a) Preferred because of easier debugging
6) Bootstrap (http://getbootstrap.com/)

Detailed Content

1. Need for frameworks
    1.1. Why do we need a framework?
    1.2. Benefits of a framework
2. Introducing Angular
    2.1. What is Angular?
    2.2. Angular versions
    2.3. Advantages of Angular
    2.4. Traditional web app – Request & response
    2.5. Angular app – Request & response
    2.6. Where does Angular fit within a modern web app?
3. Architecture overview
    3.1. Introduction to key building blocks of Angular
4. TypeScript
    4.1. What is TypeScript?
    4.2. Why TypeScript?
5. Setting up Development Environment
    5.1. Introduction to Angular CLI
    5.2. Setting up Angular
    5.3. Creating an app using Angular CLI
    5.4. Setting up Bootstrap for styling
    5.5. How an Angular app gets loaded and started?
6. Components & Templates
    6.1. What is a Component? What are its benefits?
    6.2. The Root component
    6.3. What are Decorators?
    6.4. Understanding the component decorator
    6.5. Creating and using components
    6.6. Component templates
    6.7. Component styles
    6.8. Lifecycle Hooks
7. Data Binding
    7.1. What is Data Binding?
    7.2. Interpolation
    7.3. Property binding
    7.4. Event binding
    7.5. Passing and using event data
    7.6. Two-way data binding
    7.7. Component interaction
        7.7.1.Parent to child interaction
        7.7.2.Child to parent interaction

8. Directives
    8.1. Understanding Directives
    8.2. ngIf – Outputting data conditionally
    8.3. ngStyle – Styling elements dynamically
    8.4. ngClass – Applying CSS classes dynamically
    8.5. ngFor – Outputting lists
    8.6. Creating custom directives
9. Pipes
    9.1. Introduction to Pipes
    9.2. Using pipes
    9.3. Parameterizing pipes
    9.4. Chaining multiple pipes
    9.5. Creating custom pipes
10. Services & Dependency Injection
    10.1. Need for a Service
    10.2. Creating a service
    10.3. Understanding Dependency Injection (DI) and its benefits
    10.4. Using a service within a component
    10.5. Using a service within another service
    10.6. Cross component interaction using a service
11. Introduction to Unit Testing using Jasmine & Karma
    11.1. Fundamentals of unit testing
    11.2. Setup and tear down
    11.3. Spies
    11.4. Angular testing utilities
    11.5. Working with components
    11.6. Handling component dependencies
    11.7. Testing async operations
12. Understanding Observables
    12.1. Introduction to Reactive Extensions (RxJS)
    12.2. Observables
    12.3. Creating Observables
    12.4. Using Observable operators
    12.5. Transforming Observables
    12.6. Cancelling Subscriptions
13. Forms & Validation
    13.1. Template-driven forms vs Reactive forms
    13.2. Building a form
    13.3. Registering form controls
    13.4. Submitting the form
    13.5. Understanding form state

Final Project

Title

Build a Product List web app in Angular

Objective

Implement CRUD functionality in Angular. The participants will build an Angular app with following features:

1) Create a new product (Product Form)
2) View all products (Product List)
3) View a single product
4) Update a product
5) Delete a product

Topics Covered

The participants will apply the following concepts of Angular for building the app:

1) Components & templates
2) Data Binding
3) Forms and Validation
4) Services & Dependency Injection
5) Server Communication using HttpClient
6) Routing