Comprehensive Angular 12 Programming

Course Overview

This Angular 12 Training course is packed with useful and actionable information you can apply to your work right away. Learn the fundamentals of basic Angular 12 development such as single-page browser applications, responsive websites, and hybrid mobile applications.

Get started today with Angular 12 - the most popular platform for building front-end web applications!

Duration

5 days

Course Outline

Students will complete the following modules.

Chapter 1. Introducing Angular

- What is Angular?
- Central Features of the Angular Framework
- Appropriate Use Cases
- Building Blocks of an Angular Application
- Basic Architecture of an Angular Application
- Installing and Using Angular
- Anatomy of an Angular Application
- Running the Application
- Building and Deploying the Application
- Angular for Native Mobile Apps
- Summary

Chapter 2. Introduction to TypeScript

- Programming Languages for Use with Angular
- TypeScript Syntax
- Programming Editors
- The Type System – Defining Variables
- The Type System – Defining Arrays

- Basic Primitive Types
- Type in Functions
- Type Inference
- Defining Classes
- Class Methods
- Visibility Control
- Class Constructors
- Class Constructors – Alternate Form
- Uninitialized Fields
- Interfaces
- Working with ES6 Modules
- var vs let
- Arrow Functions
- Arrow Function Compact Syntax
- Template Strings
- Generics in Class
- Generics in Function
- Summary

Chapter 3. Components

- What is a Component?
- An Example Component
- Creating a Component Using Angular CLI
- The Component Class
- The @Component Decorator
- Registering a Component to Its Module
- Component Template
- Example: HelloComponent Template
- Example: The HelloComponent Class
- Using a Component
- Run the Application
- Component Hierarchy
- The Application Root Component
- The Bootstrap File
- Component Lifecycle Hooks
- Example Lifecycle Hooks
- CSS Styles
- Summary

Chapter 4. Component Templates

- Templates
- Template Location
- The Mustache {{ }} Syntax
- Setting DOM Element Properties
- Setting Element Body Text
- Event Binding
- Expression Event Handler

- Date fields
- Radio Buttons
- Summary

Chapter 7. Reactive Forms

- Reactive Forms Overview
- The Building Blocks
- Import ReactiveFormsModule
- Construct a Form
- Design the Template
- Getting Input Values
- Initializing the Input Fields
- Setting Form Values
- Subscribing to Input Changes
- Validation
- Built-In Validators
- Showing Validation Error
- Custom Validator
- Using a Custom Validator
- Supplying Configuration to Custom Validator
- FormArray - Dynamically Add Inputs
- FormArray - The Component Class
- FormArray - The Template
- FormArray - Values
- Sub FormGroups - Component Class
- Sub FormGroups - HTML Template
- Why Use Sub FormGroups
- Summary

Chapter 8. Services and Dependency Injection

- What is a Service?
- Creating a Basic Service
- The Service Class
- What is Dependency Injection?
- Injecting a Service Instance
- Injectors
- Injector Hierarchy
- Registering a Service with the Root Injector
- Registering a Service with a Component's Injector
- Register a Service with a Feature Module Injector
- Where to Register a Service?
- Dependency Injection in Other Artifacts
- Providing an Alternate Implementation
- Dependency Injection and @Host
- Dependency Injection and @Optional
- Summary

Chapter 9. HTTP Client

- The Angular HTTP Client
- Using The HTTP Client - Overview
- Importing HttpClientModule
- Service Using HttpClient
- Making a GET Request
- What does an Observable Object do?
- Using the Service in a Component
- The PeopleService Client Component
- Error Handling
- Customizing the Error Object
- Making a POST Request
- Making a PUT Request
- Making a DELETE Request
- Summary

Chapter 10. Pipes and Data Formatting

- What are Pipes?
- Built-In Pipes
- Using Pipes in HTML Template
- Chaining Pipes
- Internationalized Pipes (i18n)
- Loading Locale Data
- The date Pipe
- The number Pipe
- Currency Pipe
- Create a Custom Pipe
- Custom Pipe Example
- Using Custom Pipes
- Using a Pipe with ngFor
- A Filter Pipe
- Pipe Category: Pure and Impure
- Pure Pipe Example
- Impure Pipe Example
- Summary

Chapter 11. Introduction to Single Page Applications

- What is a Single Page Application (SPA)
- Traditional Web Application
- SPA Workflow
- Single Page Application Advantages
- HTML5 History API
- SPA Challenges
- Implementing SPA's Using Angular
- Summary

Chapter 12. The Angular Component Router

- The Component Router
- View Navigation
- The Angular Router API
- Creating a Router Enabled Application
- Hosting the Routed Components
- Navigation Using Links and Buttons
- Programmatic Navigation
- Passing Route Parameters
- Navigating with Route Parameters
- Obtaining the Route Parameter Values
- Retrieving the Route Parameter Synchronously
- Retrieving a Route Parameter Asynchronously
- Query Parameters
- Supplying Query Parameters
- Retrieving Query Parameters Asynchronously
- Problems with Manual URL entry and Bookmarking
- Summary

Chapter 13. Advanced HTTP Client

- Request Options
- Returning an HttpResponse Object
- Setting Request Headers
- Creating New Observables
- Creating a Simple Observable
- The Observable Constructor Method
- Observable Operators
- The map and filter Operators
- The flatMap() Operator
- The tap() Operator
- The zip() Combinator
- Caching HTTP Response
- Making Sequential HTTP Calls
- Making Parallel Calls
- Customizing Error Object with catchError()
- Error in Pipeline
- Error Recovery
- Summary

Chapter 14. Angular Modules

- Why Angular Modules?
- Anatomy of a Module Class
- @NgModule Properties
- Feature Modules
- Example Module Structure
- Create a Domain Module

- Create a Routed/Routing Module Pair
- Create a Service Module
- Creating Common Modules
- Using One Module From Another
- Summary

Chapter 15. Advanced Routing

- Routing Enabled Feature Module
- Using the Feature Module
- Lazy Loading the Feature Module
- Creating Links for the Feature Module Components
- More About Lazy Loading
- Preloading Modules
- routerLinkActive binding
- Default Route
- Wildcard Route Path
- redirectTo
- Child Routes
- Defining Child Routes
- for Child Routes
- Links for Child Routes
- Navigation Guards
- Creating Guard Implementations
- Using Guards in a Route
- Summary

Chapter 16. Unit Testing Angular Applications

- Unit Testing Angular Artifacts
- Testing Tools
- Typical Testing Steps
- Test Results
- Jasmine Test Suites
- Jasmine Specs (Unit Tests)
- Expectations (Assertions)
- Matchers
- Examples of Using Matchers
- Using the not Property
- Setup and Teardown in Unit Test Suites
- Example of beforeEach and afterEach Functions
- Angular Test Module
- Example Angular Test Module
- Testing a Service
- Injecting a Service Instance
- Test a Synchronous Method
- Test an Asynchronous Method
- Using Mock HTTP Client
- Supplying Canned Response

- Testing a Component
- Component Test Module
- Creating a Component Instance
- The ComponentFixture Class
- Basic Component Tests
- The DebugElement Class
- Simulating User Interaction
- Summary

Chapter 17. Debugging

- Overview of Angular Debugging
- Viewing TypeScript Code in Debugger
- Using the debugger Keyword
- Debug Logging
- What is Angular DevTools?
- Using Angular DevTools
- Angular DevTools - Component Structure
- Angular DevTools - Change Detection Execution
- Catching Syntax Errors
- Summary

Lab Exercises

Lab 1. Introduction to Angular
Lab 2. Introduction to TypeScript
Lab 3. Introduction to Components
Lab 4. Component Template
Lab 5. Create a Photo Gallery Component
Lab 6. Template Driven Form
Lab 7. Create an Edit Form
Lab 8. Reactive Form
Lab 9. Develop a Service
Lab 10. Develop an HTTP Client
Lab 11. Use Pipes
Lab 12. Basic Single Page Application Using Router
Lab 13. Build a Single Page Application (SPA)
Lab 14. Advanced HTTP Client
Lab 15. Using Angular Bootstrap
Lab 16. Lazy Module Loading
Lab 17. Advanced Routing
Lab 18. Unit Testing
Lab 19. Debugging Angular Applications

Comprehensive Angular 12 Programming

Objectives

In this Angular 12 training, attendees will learn how to:

- Develop single page Angular applications using Typescript
- Set up a complete Angular development environment
- Create Components, Directives, Services, Pipes, Forms and Custom Validators
- Handle advanced network data retrieval tasks using Observables
- Consume data from REST web services using the Angular HTTP Client
- Handle push-data connections using the WebSockets protocol
- Work with Angular Pipes to format data
- Use advanced Angular Component Router features
- Test and debug Angular applications using built in tools
- Work with Angular CLI

Pre-requisites

Web development experience using HTML, CSS and JavaScript is required to get the most out of this Angular course. Knowledge of the browser DOM is also useful. Prior Angular experience, with AngularJS or the current version of Angular, is not required.