Course duration

- 5 days

Course Benefits

- 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

Course Outline

1. Introducing Angular
    1. What is Angular?
    2. Central Features of the Angular Framework
    3. Appropriate Use Cases
    4. Building Blocks of an Angular Application
    5. Basic Architecture of an Angular Application
    6. Installing and Using Angular
    7. Anatomy of an Angular Application
    8. Running the Application
    9. Building and Deploying the Application
    10. Angular for Native Mobile Apps
2. Introduction to TypeScript
    1. Programming Languages for Use with Angular
    2. TypeScript Syntax
    3. Programming Editors
    4. The Type System – Defining Variables
    5. The Type System – Defining Arrays
    6. Basic Primitive Types
    7. Type in Functions
    8. Type Inference
    9. Defining Classes
    10. Class Methods
    11. Visibility Control

12. Showing Validation Error
13. Custom Validator
14. Using a Custom Validator
15. Supplying Configuration to Custom Validator
16. FormArray - Dynamically Add Inputs
17. FormArray - The Component Class
18. FormArray - The Template
19. FormArray - Values
20. Sub FormGroups - Component Class
21. Sub FormGroups - HTML Template
22. Why Use Sub FormGroups

8. Services and Dependency Injection
    1. What is a Service?
    2. Creating a Basic Service
    3. The Service Class
    4. What is Dependency Injection?
    5. Injecting a Service Instance
    6. Injectors
    7. Injector Hierarchy
    8. Registering a Service with the Root Injector
    9. Registering a Service with a Component's Injector
    10. Register a Service with a Feature Module Injector
    11. Where to Register a Service?
    12. Dependency Injection in Other Artifacts
    13. Providing an Alternate Implementation
    14. Dependency Injection and @Host
    15. Dependency Injection and @Optional
9. HTTP Client
    1. The Angular HTTP Client
    2. Using The HTTP Client - Overview
    3. Importing HttpClientModule
    4. Service Using HttpClient
    5. Making a GET Request
    6. What does an Observable Object do?
    7. Using the Service in a Component
    8. The PeopleService Client Component
    9. Error Handling
    10. Customizing the Error Object
    11. Making a POST Request
    12. Making a PUT Request
    13. Making a DELETE Request
10. Pipes and Data Formatting
    1. What are Pipes?
    2. Built-In Pipes
    3. Using Pipes in HTML Template
    4. Chaining Pipes
    5. Internationalized Pipes (i18n)

6. Loading Locale Data
7. The date Pipe
8. The number Pipe
9. Currency Pipe
10. Create a Custom Pipe
11. Custom Pipe Example
12. Using Custom Pipes
13. Using a Pipe with ngFor
14. A Filter Pipe
15. Pipe Category: Pure and Impure
16. Pure Pipe Example
17. Impure Pipe Example

11. Introduction to Single Page Applications
    1. What is a Single Page Application (SPA)
    2. Traditional Web Application
    3. SPA Workflow
    4. Single Page Application Advantages
    5. HTML5 History API
    6. SPA Challenges
    7. Implementing SPA's Using Angular
12. The Angular Component Router
    1. The Component Router
    2. View Navigation
    3. The Angular Router API
    4. Creating a Router Enabled Application
    5. Hosting the Routed Components
    6. Navigation Using Links and Buttons
    7. Programmatic Navigation
    8. Passing Route Parameters
    9. Navigating with Route Parameters
    10. Obtaining the Route Parameter Values
    11. Retrieving the Route Parameter Synchronously
    12. Retrieving a Route Parameter Asynchronously
    13. Query Parameters
    14. Supplying Query Parameters
    15. Retrieving Query Parameters Asynchronously
    16. Problems with Manual URL entry and Bookmarking
13. Advanced HTTP Client
    1. Request Options
    2. Returning an HttpResponse Object
    3. Setting Request Headers
    4. Creating New Observables
    5. Creating a Simple Observable
    6. The Observable Constructor Method
    7. Observable Operators
    8. The map and filter Operators
    9. The flatMap() Operator

10. The tap() Operator
11. The zip() Combinator
12. Caching HTTP Response
13. Making Sequential HTTP Calls
14. Making Parallel Calls
15. Customizing Error Object with catchError()
16. Error in Pipeline
17. Error Recovery

14. Angular Modules
    1. Why Angular Modules?
    2. Anatomy of a Module Class
    3. @NgModule Properties
    4. Feature Modules
    5. Example Module Structure
    6. Create a Domain Module
    7. Create a Routed/Routing Module Pair
    8. Create a Service Module
    9. Creating Common Modules
    10. Using One Module From Another
15. Advanced Routing
    1. Routing Enabled Feature Module
    2. Using the Feature Module
    3. Lazy Loading the Feature Module
    4. Creating Links for the Feature Module Components
    5. More About Lazy Loading
    6. Preloading Modules
    7. routerLinkActive binding
    8. Default Route
    9. Wildcard Route Path
    10. redirectTo
    11. Child Routes
    12. Defining Child Routes
    13. for Child Routes
    14. Links for Child Routes
    15. Navigation Guards
    16. Creating Guard Implementations
    17. Using Guards in a Route
16. Unit Testing Angular Applications
    1. Unit Testing Angular Artifacts
    2. Testing Tools
    3. Typical Testing Steps
    4. Test Results
    5. Jasmine Test Suites
    6. Jasmine Specs (Unit Tests)
    7. Expectations (Assertions)
    8. Matchers
    9. Examples of Using Matchers

10. Using the not Property
11. Setup and Teardown in Unit Test Suites
12. Example of beforeEach and afterEach Functions
13. Angular Test Module
14. Example Angular Test Module
15. Testing a Service
16. Injecting a Service Instance
17. Test a Synchronous Method
18. Test an Asynchronous Method
19. Using Mock HTTP Client
20. Supplying Canned Response
21. Testing a Component
22. Component Test Module
23. Creating a Component Instance
24. The ComponentFixture Class
25. Basic Component Tests
26. The DebugElement Class
27. Simulating User Interaction

17. Debugging
    1. Overview of Angular Debugging
    2. Viewing TypeScript Code in Debugger
    3. Using the debugger Keyword
    4. Debug Logging
    5. What is Angular DevTools?
    6. Using Angular DevTools
    7. Angular DevTools - Component Structure
    8. Angular DevTools - Change Detection Execution
    9. Catching Syntax Errors

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Angular class:

- Web development experience using HTML, CSS, and JavaScript is required to get the most out of this Angular course.

Experience in the following *would be useful* for this Angular class:

- Knowledge of the browser DOM is also useful. Prior Angular experience, with AngularJS or the current version of Angular, is not required.