Angular Outline

duration 5 days

In this training, attendees will learn how to:

* Create single page web applications using Angular
* Understand how to write and organize Angular code
* Program Angular View Components and Services
* Use Angular output, two-way, event and property bindings
* Exchange data with network servers using the Angular HTTP Client
* Consume data from REST web services
* Consume WebSocket data
* Work with Angular Pipes to format data
* Create and work with Angular Forms
* Angular Component Router
* Advanced Router Usage
* Angular Animations
* Testing and debugging Angular applications
* Angular CLI
* Best practices for Angular application style and organization

Prerequisites

Trainees should have some prior understanding of web development using, HTML, CSS and JavaScript. Experience developing with Angular 2-6 (the prior versions) is not required.

Course Outline

New topics:

1. Incorporation of Angular Bootstrap into the base template of angular
2. Whole concepts of ngrx and its hands-on.

1. Introducing Angular

* What is Angular?
* Central Features of the Angular Framework
* Why Angular?
* Building Blocks of an Angular Application
* Basic Architecture of an Angular Application
* Installing and Using Angular
* A Basic Angular Application
* Anatomy of a Basic Application
* The Main Component File
* The Application Module File
* The index.html File
* The Bootstrap File
* Running the Application
* Building the Application
* Summary

2. Development Setup of Angular

* Angular Development Lifecycle
* Angular is Modular
* What is Node.js?
* Installing Node.js and NPM
* Node Package Manager (NPM)
* Package Descriptor File - package.json
* Semantic Version Numbering
* Package Version Numbering Syntax
* Updating Packages
* Uninstalling Packages
* Installing Angular Packages
* Angular CLI
* Creating a New Project
* Angular
* Development
* Dependencies
* TypeScript Typings
* Testing Tools
* Development Web Server
* Configuring the Web Server
* The Build System
* Configuring the Build
* Summary

3. Introduction to TypeScript and ES6

* Programming Languages for Use with Angular
* TypeScript Syntax
* Programming Editors
* The Type System – Defining Variables
* The Type System – Defining Arrays
* Type in Functions
* Type Inference
* The Type System – Classes & Objects
* Class Constructors
* Class Constructors – Alternate Form
* Interfaces
* Working with ES6 Modules
* Visibility Control
* Arrow Functions
* Arrow Function Compact Syntax
* Arrow Function and Caller Context
* Template Strings
* Template Strings – Variables and Expressions
* Template Strings – Multiline
* Generics - Class
* Generics - Methods
* Generics - Restricting Types
* Generics - Restricting Types: Example
* TypeScript Transpilation
* Summary

4. Components in Angular

* What is a Component?
* An Example Component
* The Component Class
* Adding a Component to Its Module
* Creating a Component Using Angular CLI
* Developing a Simple Login Component
* Component Template
* Login Component: Add HTML
* The HTML Component Template
* The templateUrl property
* Login Component: Add CSS Styling
* Login Component: Hook Up Input Fields and Button
* Login Component: Fields & Button in the Component Class
* Component Decorator Properties
* Component Hierarchy
* The Application Root Component
* Using a Regular Component
* The Build System
* Component Lifecycle Hooks
* Using a Lifecycle Hook:
* OnInit
* Summary

5. Data and Event Binding

* Binding Syntax
* One-Way Output Binding
* Binding Displayed Output Values
* Setting Component Properties
* More About Setting Properties
* Setting DOM Element Properties
* Event Binding
* Binding Events Examples
* Firing Event from a Component
* @Output() Example - Child Component
* @Output() Example - Parent Component
* Two-Way Binding of Input Fields
* Input Binding Examples
* Two Way Binding in a Component
* Use Two Way Binding
* Breaking Down ngModel
* Summary

6. Attribute Directives

* What are Attribute Directives
* Apply Styles by Changing Classes
* Changing Classes – Example
* Applying Styles Directly
* Applying Styles Directly - Example
* Controlling Element Visibility
* Setting Image Source Dynamically
* Setting Hyperlink Source Dynamically
* Writing a Custom Attribute Directive
* Using a Custom Attribute Directive
* Supplying Input to a Directive
* Handling Event from a Custom Directive
* Summary

7. Structural Directives

* Structural Directives
* Adding and Removing Elements Dynamically
* If-Else Syntax of ngIf
* Looping Using ngFor
* ngFor - Basic Example
* Creating Tables with ngFor
* ngFor Local Variables
* Manipulating the Collection
* Example - Deleting an Item
* Swapping Elements with ngSwitch
* ngSwitch - Basic Syntax
* Summary

8. Template Driven Forms

* Template Driven Forms
* Importing Forms Module
* A Basic Angular Form
* Binding Input Fields
* Accessing the NgForm Directive
* Binding the Form Submit Event
* The Submit Function
* Basic HTML5 Validation - "required" Attribute
* HTML5 vs. Angular Validation
* Angular Validators
* The NgModel Directive
* Controlling when validation is applied
* Displaying
* Form
* Validation State
* Displaying
* Field
* Validation State
* Displaying Validation State Using Classes
* Disabling Submit when Form is Invalid
* Submitting the Form
* Binding to Object Variables
* Binding to Object Variables - Code
* Additional Input Types
* Checkboxes
* Select (Drop Down) Fields
* Rendering Options for Select (Drop Down)
* Date fields
* Radio Buttons
* Summary

9. Reactive Forms

* Reactive Forms Overview
* The Building Blocks
* Import ReactiveFormsModule
* Construct a Form
* Design the Template
* FormControl Constructor
* Getting Form Values
* Setting Form Values
* The Synchronous Nature
* Subscribing to Input Changes
* Validation
* Built-In Validators
* Showing Validation Error
* Custom Validator
* Using a Custom Validator
* Sub FormGroups - Component Class
* Sub FormGroups - HTML Template
* Why Use Sub FormGroups
* Summary

10. Angular Modules

* Why Angular Modules?
* But, We Already Had ES6 Module
* Angular Built-in Modules
* The Root Module
* Feature Modules
* Create Feature Module Using CLI
* The Module Class
* @NgModule Properties
* Using One Module From Another
* Importing BrowserModule or CommonModule
* Lazy-Loaded Modules
* How to Organize Modules?
* Third Party Modules
* Summary

11. Services and Dependency Injection

* What is a Service?
* Creating a Basic Service
* What is Dependency Injection?
* What Dependency Injection Looks Like
* Injectors
* Injector Hierarchy
* Register a Service with a Module Injector
* Registering a Service with the Root Injector
* Registering a Service with a Component's Injector
* Where to Register a Service?
* Dependency Injection in Other Artifacts
* Providing an Alternate Implementation
* Dependency Injection and @Host
* Dependency Injection and @Optional
* Summary

12. HTTP Client

* The Angular HTTP Client
* Using The HTTP Client - Overview
* Importing HttpClientModule
* Simple Example
* Service Using HttpClient
* ES6 Import Statements
* Making a GET Request
* What does an Observable Object do?
* Using the Service in a Component
* The PeopleService Client Component
* Error Handling
* Making a POST Request
* Making a PUT Request
* Making a DELETE Request
* Summary

13. Pipes and Data Formatting

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

14. 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

15. The Angular Component Router

* The Component Router
* View Navigation
* The Angular Router API
* Creating a Router Enabled Application
* Hosting the Routed Component
* 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

16. Advanced HTTP Client

* Request Options
* Returning an HttpResponse Object
* Setting Request Headers
* Creating New Observables
* Creating a Simple Observable
* The Observable.create() Method
* Observable Operators
* More About map
* Piping Operators
* The flatMap() Operator
* The tap() Operator
* The zip() Operator
* Caching HTTP Response
* Making Sequential HTTP Calls
* Making Parallel Calls
* Customizing Error Object with catchError()
* Error in Pipeline
* Error Recovery
* Summary

18. Advanced Routing

* Routing Overview
* Routing Enabled Project
* Routing Enabled Feature Module
* Using the Feature Module
* Lazy Loading the Feature Module
* Creating Links for the Feature Module Components
* More About Lazy Loading
* routerLinkActive binding
* Default Route
* Wildcard Route Path
* redirectTo
* Child Routes
* Defining Child Routes
* <router-outlet> for Child Routes
* Links for Child Routes
* Navigation Guards
* Creating Guard Implementations
* Using Guards in a Route
* Route Animations
* Summary

19. Introduction to Testing Angular Applications

* Unit Testing Angular Artifacts
* Testing Tools
* Testing Setup
* 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

20. Debugging

* Overview of Angular Debugging
* Viewing TypeScript Code in Debugger
* Using the debugger Keyword
* Inspecting Components
* Saving ng.probe Component References
* Modifying Values using Component References
* Debug Logging
* What is Augury?
* Installing Augury
* Opening Augury
* Augury - Component Tree
* Augury - Router Tree.
* Augury - NgModules Tab
* Common Exceptions
* Common Exceptions: 'No such file: package.json'
* Common Exceptions: 'Cant bind to ngModel'
* Common Exceptions: 'router-outlet not a known element'
* Common Exceptions: 'No provider for Router!'
* Summary

Additional Topic

1.      Introduction of a new interface - **UrlSegment**[] to **CanLoad**interface

2.      Introduction of a new interface -  **DoBootstrap**interface

3.      Angular 7 Introduction of a new compiler - **Compatibility Compiler (ngcc)**

4.      Introduce a new Pipe called - **KeyValuePipe**

5.      Angular 7 now supporting to **TypeScript**2.9.

6.      Introduction of a new elements features - enable **Shadow DOM** v1 and slots

7.      Introduction of a new router features - warn if navigation triggered outside Angular zone

8.      Introduction of a new mappings for **ngfactory**  and **ngsummary** files to their module names in **AOT**summary resolver.

9.      Introduction of a new "**original**" placeholder value on extracted **XMB**

10. Introduction of a new ability to recover from malformed **URLs**

11. Introduction of a new compiler support **dot (.)** in import statements and also avoid a crash in **ngc-wrapped**

12. Update compiler to flatten **nested template fns**

Lab Exercises

Lab 1. Introduction to Angular  
Lab 2. Introduction to TypeScript  
Lab 3. Introduction to Components   
Lab 4. Create the Subscription Form Component  
Lab 5. Understanding Data Binding  
Lab 6. One Way Data Binding in a Custom Component  
Lab 7. Using Basic Angular Directives  
Lab 8. Using Structural Directives   
Lab 9. Custom Attribute Directive  
Lab 10. Template Driven Form  
Lab 11. Validation of a Template Driven Form  
Lab 12. Reactive Form  
Lab 13. Service and Dependency Injection  
Lab 14. HTTP Client Development  
Lab 15. Using Pipes   
Lab 16. Basic Single Page Application Using Router  
Lab 17. Angular Communication with REST Services  
Lab 18. HTTP Error Handling and Recovery  
Lab 19. Using Angular Bootstrap  
Lab 20. Consuming Data from Web Sockets   
Lab 21. Lazy Module Loading  
Lab 22. Advanced Routing  
Lab 23. Introduction to Unit Testing  
Lab 24. Debugging Angular Applications