Introduction to Angular Programming (3 Days)

This Introduction to Angular training class covers all the essential topics and is designed to get students up and running with Angular development in the shortest amount of time. Developers, application designers, and anyone tasked with supporting Angular applications will gain valuable knowledge required for success in their job roles.

COURSE BENEFITS

- Develop single page Angular applications using Typescript.
- Set up a complete Angular development environment.
- Create Components.
- Work with Component Templates.
- Build Inter Component Communication.
- Create Template Driven Forms.
- Develop Reactive Forms.
- Services and Dependency Injection.
- Work with the HTTP Client.
- Work with Pipes and Data Formatting.
- Navigate between multiple view screens using the Angular component router.

CLASS PREREQUISITES

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

- Web development experience using HTML, CSS, and JavaScript. Prior Angular experience, with AngularJS or the current version of Angular, is not required.

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

- Knowledge of the browser DOM.

Course Outline

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

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

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

Component Templates

- Templates
- Template Location
- The Mustache {{ }} Syntax
- Setting DOM Element Properties
- Setting Element Body Text
- Event Binding
- Expression Event Handler
- Prevent Default Handling
- Attribute Directives
- Apply Styles by Changing CSS Classes
- Example: ngClass
- Applying Styles Directly
- Structural Directives
- Conditionally Execute Template
- Example: ngIf
- Looping Using ngFor
- ngFor Local Variables
- Manipulating the Collection
- Example - Deleting an Item
- Item Tracking with ngFor
- Swapping Elements with ngSwitch
- Grouping Elements
- Template Reference Variable

Inter Component Communication

- Communication Basics
- The Data Flow Architecture
- Preparing the Child to Receive Data
- Send Data from Parent
- More About Setting Properties

Firing Event from a Component
@Output() Example - Child Component
@Output() Example - Parent Component
Full Two Way Binding
Setting up Two Way Data Binding in Parent

Template Driven Forms

Template Driven Forms
Importing Forms Module
Basic Approach
Setting Up a Form
Getting User Input
Omitting ngForm Attribute
Initialize the Form
Two Way Data Binding
Form Validation
Angular Validators
Displaying Validation State Using Classes
Additional Input Types
Checkboxes
Select (Drop Down) Fields
Rendering Options for Select (Drop Down)
Date fields
Radio Buttons

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

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

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

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

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

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