Introduction to Angular 5

Course Summary

Description

Introduction to Angular 5 teaches students the skills and best practices they need to design, build, test, and deploy applications that provide rich end-user experiences similar to a desktop application while still offering the ease of deployment of a web application.

Objectives

After taking this course, students will be able to:
- Understand how single-page web application architectures (including Angular) are different than traditional web development architectures
- Use new JavaScript (ES6) language features including Classes, Modules, and Arrow Functions
- Use new TypeScript language features including Static Types, Interfaces, and Generics
- Build an application from scratch using Angular 5
- Understand Angular coding and architecture best practices
- Understand and use Angular Model-driven Forms, Observables, Dependency Injection, and Routing
- Communicate with a backend server using Angular's HttpClient service to load and save data.

Topics

- Introduction
- Why Angular?
- Understanding Angular Versions
- Upgrading to Angular 5 from Angular 2 or Angular 4
- Angular 5 Features
- TypeScript and ECMAScript 6 (ES6) Fundamentals
- Angular 5 Basics

- Template Syntax
- Components
- Services & Dependency Injection
- Dependency Injection
- Model-driven Forms (Reactive Forms)
- RxJS and Observables
- Communicating with the Server using the HttpClient Service
- Router

Audience

This course is designed for those wanting to learn how to build an application from scratch using Angular 5.

Prerequisites

Before taking this course, attendees must have substantial prior experience developing with JavaScript.

Duration

Three days

Introduction to Angular 5

Course Outline

I. Introduction
- A. Why Angular?
- B. Understanding Angular Versions
- C. Upgrading to Angular 5 from Angular 2 or Angular 4
- D. Angular 5 Features

II. TypeScript and ECMAScript 6 (ES6) Fundamentals
- A. Classes
- B. ES Modules
- C. Arrow Functions
- D. Template Literals
- E. Scoping using let and const Keywords
- F. Spread Syntax and Rest Parameters
- G. Destructuring
- H. Decorators (JavaScript Aspect-Oriented Programming)

III. Angular 5 Basics
- A. Components
- B. Templates
    1. Inline Templates
    2. Multi-line Templates using ES6 Template Literals
    3. External with Component-relative Paths
- C. Modules
    1. Angular Modules vs. ES Modules
- D. Models

IV. Template Syntax
- A. HTML in templates
- B. Interpolation
- C. Binding syntax
- D. Property binding
- E. Event binding
- F. Two-way data binding
- G. Attribute, class, and style bindings

- H. Built-in Directives
    1. Built-in attribute directives: NgClass, NgStyle, NgModel
    2. Built-in structural directives: NgIf (includes enhanced *ngIf syntax), NgFor
- I. Template Input Variables
- J. The NgSwitch Directives
- K. Template Reference Variables
- L. Input and output properties
- M. Template Expression Operators
- N. Pipe (|)
- O. Safe Navigation Operator (?.)

V. Components
- A. Component Lifecycle Hooks
    1. Implementing the OnInit Lifecycle Hook
- B. Component Communication
    1. Input properties
    2. Output properties: Custom Events using EventEmitters

VI. Services & Dependency Injection
- A. Using a services to access data
- B. Using a service to encapsulate business logic
- C. Understanding the scope of services

VII. Dependency Injection
- A. Understanding Dependency Injection
- B. Angular's Dependency Injection System
- C. Registering
- D. Injecting
- E. Hierarchical Injection

VIII. Model-driven Forms (Reactive Forms)
- A. Importing the ReactiveFormsModule
- B. FormControl, FormGroup, and AbstractControl

Introduction to Angular 5

Course Outline (cont'd)

C. Binding DOM Elements to FormGroups and FormControls
D. Validation Rules, Messages, and Styles
E. Refactoring ReactiveForms for Reuse
F. Custom Validators

IX. **RxJS and Observables**
A. What is an Observable?
B. Observable Operators
C. Creating Observables Using Static Operators
D. What is an Observer?
E. Observer Example
F. Subject
G. Subject Example
H. EventEmitter or Observable

X. **Communicating with the Server using the HttpClient Service**
A. Deciding between Promises or Observables (RxJS)
B. Making a HTTP GET Request
C. Sending data to the server using Http POST and PUT Requests
D. Issuing a Http DELETE Request
E. Intercepting Requests and Responses
F. WebSockets

XI. **Router**
A. Importing the RouterModule and Routes
B. Configuring Routes
C. Displaying Components using a RouterOutlet
D. Navigating declaratively with RouterLink
E. Navigating with code using the Router
F. Accessing parameters using ActivatedRoute
G. Organizing your code into Modules