

## Comprehensive Angular 11

### Course Summary

#### Description

Angular allows developers to easily build dynamic, responsive single-page web applications that dynamically rewrite portions of the current page rather than having to generate a new page in response to every request.

Comprehensive Angular training teaches attendees how to build applications using ES6, TypeScript, and modern front-end tools, including npm and Webpack. Students also gain an understanding of application architecture and design best practices in Angular, as well as learn how to authenticate, unit test, and manage application state in an Angular application.

Note: This course is appropriate for all versions, Angular 2 through Angular 11.

#### Objectives

By the end of this course, students will be able to:

- Understand how single-page web application architectures are different than traditional web application architectures
- Use new JavaScript (ES6) language features including Classes, Modules, and Arrow Functions
- Use new TypeScript language features including Types, Decorators, Interfaces, and Generics
- Learn Angular coding and architecture best practices including project layout and using container and presentation components
- Understand and use Angular model-driven forms, observables, dependency injection, and routing
- Communicate with a backend server using Angular's HttpClient to load and save data
- Configure the router and navigate between components
- Leverage IVY, the new Angular compiler, for faster and smaller builds
- Unit test all parts of an application including Components, Services, and Pipes
- Understand RxJS and Observables and where they can be used
- Implement Authentication and Authorization in an Angular Application
- Optimize Angular Performance by changing Change Detection Strategies
- Setup new projects from scratch using the Angular CLI
- Scaffold modules, components, services, models, routes, and unit tests in accordance with best practices using the Angular CLI
- Build and deploy an application to production using the Angular CLI
- Understand & Preview Ivy: The Next-Generation Compilation & Rendering Pipeline
- Write End-to-End Tests (optional; taught only if this applies to your group)
- Upgrade an existing application from AngularJS to Angular 11 (optional; taught only if this applies to your group)

## Comprehensive Angular 11

### Course Summary (cont.)

---

#### Topics

- Introduction
- TypeScript and ECMAScript 6 (ES6) Fundamentals
- Angular Overview
- Components
- Angular Modules (NgModule)
- Project Set-Up (Using the Angular CLI)
- Data Binding
- Directives
- Pipes
- Advanced Components
- Services & Dependency Injection
- Dependency Injection
- Model-driven Forms (Reactive Forms)
- Communicating with the Server using the HttpClient Service
- Router
- Deploying an Angular Application to Production
- Ivy: Next-Generation Compilation & Rendering Pipeline
- Upgrading to the latest version of Angular from earlier versions
- Unit Testing
- RxJS and Observables
- Security
- Change Detection
- Advanced Routing
- Advanced Dependency Injection
  - Providers
- Pipes
- Conclusion

#### Prerequisite

Students should have programming experience with an object-oriented language. In addition, some experience with JavaScript is helpful, but the new language features of JavaScript and TypeScript are covered/reviewed in class.

#### Duration

Five Days

## Comprehensive Angular 11

---

### Course Outline

- I. Introduction**
- II. TypeScript and ECMAScript 6 (ES6) Fundamentals**
  - A. TypeScript Installation, Configuration & Compilation
  - B. Type Annotations
  - C. Classes
  - D. Scoping using let, var, and const Keywords
  - E. Arrow Functions
  - F. ES Modules
  - G. Decorators
  - H. Template Literals
  - I. Spread Syntax and Rest Parameters
  - J. Destructuring
- III. Angular Overview**
  - A. Benefits of Building using Angular
  - B. Understanding Angular Versions
  - C. Single-page Web Application Architectures vs. Traditional Server-side Web Application Architectures
  - D. Angular Style Guide
  - E. Angular Architecture
  - F. Angular Compared to Other JavaScript Libraries and Frameworks (React, VueJS, etc...)
  - G. Your First Angular Application
- IV. Components**
  - A. Understanding Components
  - B. Component Properties & Methods
  - C. Templates: Inline, Multi-line, and External with Component-relative Paths
- V. Angular Modules (NgModule)**
  - A. Angular Modules vs. ES Modules
  - B. Organizing your code into Feature Modules
- VI. Project Set-Up (Using the Angular CLI)**
  - A. Angular CLI Features
  - B. Creating a New Project (CLI Prompts)
  - C. Generating Code
  - D. Customizing the Angular CLI
- VII. Data Binding**
  - A. Interpolation
  - B. Property binding
  - C. Event binding
  - D. Two-way data binding
- VIII. Directives**
  - A. Structural: ngFor, ngIf, ngSwitch
  - B. Attribute: ngClass, ngStyle
- IX. Pipes**
  - A. Built-in Pipes: Using, Passing Parameters, Chaining
- X. Advanced Components**
  - A. Component Communication using @Input, @Output
  - B. Component Architecture
  - C. Component Styles
  - D. Component Lifecycle Hooks
  - E. Evaluating UI Component Frameworks & Libraries
- XI. Services & Dependency Injection**
  - A. Using a service to access data
  - B. Using a service to encapsulate business logic
  - C. Understanding the scope of services
- XII. Dependency Injection**
  - A. Understanding Dependency Injection
  - B. Angular's Dependency Injection System
  - C. Registering
  - D. Injecting
- XIII. Model-driven Forms (Reactive Forms)**
  - A. Importing the ReactiveFormsModule
  - B. FormControl, FormGroup, and AbstractControl
  - C. Binding DOM Elements to FormGroups and FormControls
  - D. Validation Rules, Messages, and Styles
  - E. Refactoring Reactive Forms for Reuse

## Comprehensive Angular 11

---

### Course Outline (cont.)

- F. Custom Validators
- XIV. *Communicating with the Server using the HttpClient Service***
  - A. Deciding between Promises or Observables (RxJS)
  - B. Making an HTTP GET Request
  - C. Sending data to the server using Http POST and PUT Requests
  - D. Issuing an HTTP DELETE Request
  - E. Intercepting Requests and Responses
- XV. *Router***
  - A. Importing the RouterModule
  - 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
- XVI. *Deploying an Angular Application to Production***
  - A. Building an application using the Angular CLI
  - B. Differential loading: creating a modern build (ES2015) and a legacy build (ES5)
  - C. Deploying to a web server
- XVII. *Ivy: Next-Generation Compilation & Rendering Pipeline***
  - A. Understanding Ivy
  - B. Ivy Goals
  - C. Ivy Benefits & Current Status
  - D. Ivy Bundle Sizes
- XVIII. *Upgrading to the latest version of Angular from earlier versions***
  - A. 2.x and above
  - B. Update Guide
  - C. Deprecation Guide
  - D. Patching Dependencies: npm audit fix
- E. Looking for AngularJS to Angular upgrades? See optional topics below.
- XIX. *Unit Testing***
  - A. Tools: Jasmine, Karma
  - B. Jasmine Syntax: describe, it, beforeEach, afterEach, matchers
  - C. Setup and your First Test
  - D. Testing Terminology: Mock, Stub, Spy, Fakes
  - E. Angular Testing Terminology: TestBed, ComponentFixture, debugElement, async, fakeAsync, tick, inject
  - F. Simple Component Test
  - G. Detecting Component Changes
  - H. Testing a Component with properties (inputs) and events (outputs)
  - I. Testing a Component that uses the Router
  - J. Testing a Component that depends on a Service
  - K. Testing a Service and Mocking its HTTP requests
  - L. Testing a Pipe
- XX. *RxJS and Observables***
  - A. What is an Observable?
  - B. Creating Observables
  - C. What is an Observer?
  - D. Observer Example
  - E. Operators: map, switchMap, debounceTime, distinctUntilChanged
  - F. Practical Application of using RxJS
  - G. Subject
  - H. Subject Example
  - I. EventEmitter or Observable
- XXI. *Security***
  - A. Best Practices
  - B. Preventing Cross-site Scripting (XSS)
  - C. Trusting values with the DOMSanitizer
  - D. HTTP Attacks (CSRF and CSSI)

## Comprehensive Angular 11

---

### Course Outline (cont.)

- E. Authentication using JSON Web Tokens (JWT)
- F. Authorization: Router Guards
- XXII. *Change Detection***
  - A. Understanding Zone.js and Change Detection
  - B. Change Detection Strategies Default and OnPush
- XXIII. *Advanced Routing***
  - A. Lazy-loading Angular Modules (using Dynamic Imports)
  - B. Nested or Child Routes
- XXIV. *Advanced Dependency Injection***
  - A. *Providers*
  - B. Hierarchical Injection
- XXV. *Pipes***
  - A. Creating a custom Pipe using PipeTransform
  - B. Understanding Pure and Impure Pipes
- XXVI. *Conclusion***
  - Choose any two optional topics. If desired, the course can be customized to include more than two of these topics if other topics are scaled back or removed.
    - npm QuickStart
      - Installing Dependencies
      - Understanding package.json and package-lock.json
      - Using npm as a Build Tool
    - Managing Shared Application State using ngrx and Redux
      - Benefits Overview
      - Three Principles of Redux: Single Source of Truth, State is Read-Only, Pure Functions
      - Examples of Pure Functions
      - Reducers
      - Simple ngrx Example
      - Time-traveling with Redux Devtools
    - Upgrade Strategies from AngularJS
      - High-level Approaches
      - Concept Mapping AngularJS to Angular
      - UpgradeAdapter
      - What can be Upgraded or Downgraded
      - What cannot be Upgraded or Downgraded
      - UpgradeAdapter and Dependency Injection
    - End-to-End Testing
      - What is Protractor?
      - Why Protractor?
      - Using Locators
      - Page Objects
      - Debugging E2E Tests