

Introduction to Angular 21

Duration: 3 Days

Prerequisites: Basic HTML, CSS, JavaScript, and TypeScript knowledge.

Course Overview: This course introduces modern Angular development using Angular 21, covering standalone components, dependency injection, routing, forms, HTTP communication, and best practices.

Learning Objectives

- Understand Angular architecture and SPA development.
- Build applications using standalone components.
- Implement data binding and component communication.
- Consume REST APIs using HttpClient.
- Create template-driven and reactive forms.
- Implement routing and route guards.
- Apply Angular best practices.

Web Development Fundamentals

- Introduction to Web Development
- Static Websites vs Dynamic Websites
- Frontend, Backend, and Full-Stack Development
- Introduction to Single Page Applications (SPA)
- What is a Web Server?
 - IIS
 - Nginx
 - Apache HTTP Server
 - Caddy
- Understanding HTTP vs HTTPS
- Understanding Request-Response Cycle

Angular Fundamentals

Introduction

- Introduction to TypeScript
- Introduction to Single Page Application
- Setting up Development Environment

- Angular Initial Application
- Angular Directory Structure
- Angular Building Blocks

Working with Components

- Angular Components and the @Component Decorator
- Standalone Components
- Creating Components using Angular CLI
- Component Decorator Metadata

Architecture

- MVC Architecture
- MVVM Architecture

Data Binding

- Interpolation
- Property Binding
- Event Binding
- Two-way Data Binding [(ngModel)]

Structural Directives

- @if
- @else
- @switch
- @case
- @default
- @for

Component Communication

- @Input()
- @Output()

Advanced Angular

Dependency Injection

- Dependency Injection
- @inject()
- Working with Services

Working with APIs

- Introduction to REST APIs
- Working with JSON Data

- Introduction to HttpClient
- Sending HTTP Requests using HttpClient
 - GET Request
 - POST Request
 - PUT Request
 - DELETE Request
- CRUD Operations using HttpClient and HttpHeaders
- HTTP Error Handling with RxJS catchError()

Angular Forms

- Template Driven Forms
- Template Driven Form Validation
- Reactive Forms
- Reactive Form Validation
- Accessing Form Values
- Resetting Form Values

Routing

- Introduction to Routing
- Implementing Routing
- Implementing Child Routes
- Introduction to Route Guards
- CanActivate Guard
- CanDeactivate Guard

Zoneless Angular

- What is Zone.js?
- Why Zoneless?
- Configuring Zoneless
- Change Detection without Zone.js
- Best Practices

Micro Frontends

- Introduction
- Iframes
- Native Federation

Angular Best Practices

- Overview of Overview of Feature Module and Shared Module
- Lazy Loading Modules
- Deferred Loading