Angular 21

This 3-day accelerated Angular 21 program delivers focused coverage of essential Angular development topics, combining core framework concepts, practical frontend patterns, and selected hands-on exercises to help participants build solid working knowledge in a short time.

Overview

This 3-day accelerated Angular 21 program is designed for developers who want a focused and practical introduction to modern Angular development. The course covers TypeScript foundations, standalone architecture, components, forms, services, signals, RxJS, API integration, routing, and core application design patterns, while incorporating selected hands-on exercises and guided demonstrations throughout the delivery. Participants will build solid working knowledge of Angular 21 and gain a clear understanding of how its essential features are applied in real-world frontend application development.

Introduction

Angular 21 is a modern frontend framework designed for building scalable, component-driven web applications with strong support for standalone APIs, fine-grained reactivity, signals, and an improved developer experience. This program introduces participants to the essential Angular 21 development model, covering how applications are structured, how components communicate, how forms and services are implemented, and how modern routing and API interaction patterns are applied in practice. Along the way, participants also gain exposure to important contemporary Angular capabilities such as signals-based state handling, modern rendering control, and practical architectural patterns that support maintainable and responsive application development.

Topics Covered

- Introduction to TypeScript
- Introduction to Angular
- Introduction to Single Page Applications
- Building with Components
- Data Binding and Event Handling
- Attribute Directives
- Modern Control Flow and Structural Rendering
- Template Driven Forms

- Reactive Forms
- Services and Dependency Injection
- Signals and Reactive Programming
- RxJS and Observables
- HTTP Client and API Integration
- Pipes and Data Formatting
- Angular Routing and Navigation
- Angular Application Architecture

Prerequisites

Before attending this course, students should have general programming experience and knowledge of HTML, CSS, and JavaScript.

Audience:

Frontend developers, web developers, software engineers, and teams modernizing Angular applications.

Duration: 3 Days

Course Outline

Day 1

1. Introduction to TypeScript
   - 1.1. Overview
   - 1.2. Environment Setup
   - 1.3. Basic Syntax
   - 1.4. Types, Variables and Operators
   - 1.5. Functions and Arrow Functions
   - 1.6. Interfaces, Classes and Objects
   - 1.7. Modules (ES Modules)
   - 1.8. Summary

2. Introduction to Angular
   - 2.1. Why Angular 21
   - 2.2. Angular 21 Key Features
   - 2.3. Angular CLI Overview
   - 2.4. Installing and Using Angular 21

Day 2

Day 3

11. Signals and Reactive Programming
    - 11.1. Introduction to Angular Signals
    - 11.2. Creating Signals
    - 11.3. Computed Signals
    - 11.4. Signal-Based State Management
    - 11.5. Signals vs Observables
    - 11.6. Using Signals with Components
    - 11.7. Summary

12. RxJS and Observables
    - 12.1. Introduction to RxJS
    - 12.2. Observables Overview
    - 12.3. Creating Observables
    - 12.4. Subscribing and Unsubscribing
    - 12.5. Common RxJS Operators
    - 12.6. Async Pipe
    - 12.7. Summary

13. HTTP Client and API Integration
    - 13.1. Angular HTTP Client Overview
    - 13.2. Setting Up HTTP Client
    - 13.3. Creating API Services
    - 13.4. Making HTTP GET Requests
    - 13.5. POST, PUT and DELETE Requests
    - 13.6. Handling HTTP Responses
    - 13.7. Error Handling Strategies
    - 13.8. Consuming APIs in Components
    - 13.9. Summary

14. Pipes and Data Formatting
    - 14.1. What Are Pipes
    - 14.2. Built-in Pipes
    - 14.3. Decimal Pipe
    - 14.4. Currency Pipe
    - 14.5. Date Pipe
    - 14.6. Creating Custom Pipes
    - 14.7. Summary

15. Angular Routing and Navigation
    - 15.1. Routing Overview
    - 15.2. Route Configuration Using Standalone APIs
    - 15.3. Router Outlet
    - 15.4. Navigation Links
    - 15.5. Programmatic Navigation
    - 15.6. Route Parameters
    - 15.7. Lazy Loading Routes
    - 15.8. Summary

16. Angular Application Architecture
    - 16.1. Feature-based folder structure
    - 16.2. Service-based state management
    - 16.3. Separation of concerns
    - 16.4. Reusable component design
    - 16.5. Summary

17. Final Project
    - 17.1. Build a compact Angular application
    - 17.2. Component-based UI structure
    - 17.3. Form handling and validation
    - 17.4. Service integration with API calls
    - 17.5. Routing and navigation flow
    - 17.6. Signals or RxJS for state updates
    - 17.7. Summary and wrap-up