

Angular 14: Components, Reactive Forms & Routing

Learn to build Angular 14 applications using components, templates, forms, routing and services.

13

MODULES

205

TOPICS

COURSE OVERVIEW

This course covers the fundamentals of Angular development, starting with TypeScript basics and progressing to building components, templates and inter-component communication. It examines template-driven and reactive forms, services with dependency injection, pipes for data formatting, and use of the HTTP client for GET, POST, PUT and DELETE requests. It then addresses the Angular component router, including navigation, route parameters, child routes and navigation guards, along with advanced HTTP client techniques using observables. The course concludes with unit testing Angular applications using Jasmine and the Angular testing tools.

WHO THIS COURSE IS FOR

Angular Developers

Front-End Developers

Web Developers

Software Engineers

UI Developers

LEARNING OBJECTIVES

- Explain the core features and architecture of Angular applications
- Apply TypeScript syntax including classes, interfaces and generics
- Build components and structure component hierarchies
- Implement data binding and directives within component templates
- Configure inter-component communication using inputs and outputs
- Develop template-driven and reactive forms with validation
- Implement services and dependency injection
- Configure routing, navigation guards and lazy-loaded feature modules

13 Modules, 205 Topics

01

Introducing Angular

TOPICS

- 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

02

Introduction to TypeScript

TOPICS

- TypeScript Syntax
- Programming Editors
- The Type System – Defining Variables
- The Type System – Defining Arrays
- Type in Functions
- Type Inference
- Defining Classes
- Class Methods
- Class Constructors
- Class Constructors – Alternate Form
- Interfaces
- Working with ES6 Modules
- Visibility Control
- var vs let
- Arrow Functions
- Arrow Function Compact Syntax
- Arrow Function and Caller Context
- Template Strings
- Generics in Class
- Generics in Function

03

Components

TOPICS

- 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
- Using a Component
- Run the Application
- Component Hierarchy
- The Application Root Component
- The Bootstrap File

04

Component Templates

TOPICS

- Templates
- Template Location
- The Mustache {{ }} Syntax
- Setting DOM Element Properties
- 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
- Template Reference Variable

05

Inter Component Communication

TOPICS

- 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

06

Template Driven Forms

TOPICS

- 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

07

Reactive Forms

TOPICS

- Reactive Forms Overview
- The Building Blocks
- Import ReactiveFormsModule
- Construct a Form
- Design the Template
- FormControl Constructor
- Getting Form Values
- Setting Form Values
- The Synchronous Nature
- Subscribing to Input Changes
- Validation
- Built-In Validators
- Showing Validation Error
- Custom Validator
- Using a Custom Validator

08

Services and Dependency Injection

TOPICS

What is a Service? · Creating a Basic Service · The Service Class · What is Dependency Injection? · Injecting a Service Instance · Injectors

09

Pipes and Data Formatting

TOPICS

- What are Pipes?
- Built-In Pipes
- Using Pipes in HTML Template
- Chaining Pipes
- The number Pipe
- Currency Pipe
- Create a Custom Pipe
- Custom Pipe Example
- Using Custom Pipes
- Using a Pipe with ngFor
- A Filter Pipe

HTTP Client

TOPICS

- The Angular HTTP Client
- Using The HTTP Client - Overview
- Importing HttpClientModule
- Simple Example
- Service Using HttpClient
- ES6 Import Statements
- Making a GET Request
- What does an Observable Object do?
- Using the Service in a Component
- Error Handling
- Customizing Error Object with .catch()
- Making a POST Request
- Making a PUT Request
- Making a DELETE Request

The Angular Component Router

TOPICS

- 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 a Route Parameter
- Routing Enabled Feature Module
- Using the Feature Module
- Lazy Loading the Feature Module
- Creating Links for the Feature Module
- Components
- More About Lazy Loading
- Preloading Modules
- routerLinkActive binding
- Default Route
- Wildcard Route Path
- redirectTo
- Child Routes
- Defining Child Routes
- for Child Routes
- Links for Child Routes
- Navigation Guards
- Creating Guard Implementations
- Using Guards in a Route

Advanced HTTP Client

TOPICS

- Request Options
- Returning an `HttpResponse` Object
- Setting Request Headers
- Creating New Observables
- Creating a Simple Observable
- The `Observable.create()` Method
- Observable Operators
- More About `map`
- Piping Operators
- The `flatMap()` Operator
- The `tap()` Operator
- The `zip()` Operator
- Caching HTTP Response
- Making Sequential HTTP Calls
- Making Parallel Calls
- Customizing Error Object with `catchError()`

Unit Testing Angular Applications

TOPICS

- Unit Testing Angular Artifacts
- Testing Tools
- Typical Testing Steps
- Test Results
- Jasmine Test Suites
- Jasmine Specs (Unit Tests)
- Expectations (Assertions)
- Matchers
- Examples of Using Matchers
- Using the `not` Property
- Setup and Teardown in Unit Test Suites
- Example of `before Each` and `after Each` Functions
- Angular Test Module
- Example Angular Test Module
- Testing a Service
- Injecting a Service Instance
- Test a Synchronous Method
- Test an Asynchronous Method
- Using Mock HTTP Client
- Supplying Canned Response
- Testing a Component
- Component Test Module
- Creating a Component Instance
- The Component Fixture Class
- Basic Component Tests
- The Debug Element Class

WHAT IS INCLUDED

Everything You Need to Succeed

- Koenig course material for every module
- Delivery by certified instructors and subject matter experts
- Hands-on exercises throughout the course
- Course completion certificate from Koenig

ABOUT KOENIG

Who We Are

Established in 1993, Koenig Solutions is a globally recognised training provider with 35+ technology partnerships, 5,000+ training programmes and a network of certified instructors and subject matter experts, supporting individuals and enterprise teams worldwide.

FLEXIBLE DELIVERY

Learn the Way That Suits You

Classroom Training

Live Online Classes (ILO)

Fly-Me-A-Trainer (FMAT)

Flexi (Self-Paced)

1-on-1 Training

Ready to Get Started?

Talk to a learning advisor about schedules, group training and customised delivery.

Visit koenig-solutions.com

✉ info@koenig-solutions.com