

Angular 20 Masterclass

Comprehensive Hands-on Curriculum (Standalone, Signals, SSR, Forms & Architecture)

Website: focushat.com

Phone: +91 8112353636

Email: hello@focushat.com

Course Key Highlights: All concepts of Angular 20 are covered in detail including Standalone-based Components, Signals, Server-Side Rendering (SSR), Directives, Reactive/Template Forms, RxJS, REST Web Services, and complete real-world project deployment.

Chapter 1: Introducing Course

- Course Overview & Objectives
- Where to find the Course Source Code

Chapter 2: Introducing Angular

- What is Angular? & Why Angular?
- Prerequisite Knowledge Check
- Building Blocks & Basic Architecture of Angular
- Evolution: AngularJS vs. Angular 2 through Angular 20
- Angular Semantic Versioning & Release Schedule
- Angular Preview Releases & Support Policy

Chapter 3: Development Setup of Angular

- Angular Environment Setup & VS Code Editor
- Understanding Node.js & NPM Role in Angular
- Angular CLI Installation & Commands
- Creating, Building, and Running First Application
- Editing & Inspecting First Angular App

Chapter 4: Project Structure

- Detailed Project Structure Walkthrough
- package.json & package-lock.json Syntax
- Managing, Updating, and Uninstalling Packages
- Angular Workflows: Module-based vs. Standalone

Chapter 15: Services and Dependency Injection

- Creating & Injecting Services in Components
- Dependency Injection (DI) Architecture: Injector & Provider
- @Injectable & @Inject Decorators
- providedIn & Tree-Shakeable Providers
- Component-level vs Module-level Service Scope
- Nested Services (Injecting Service into Service)
- Single vs Multiple Service Instances

Chapter 16: Router

- Component Router Architecture & Terminology
- Router Outlet, RouterLink (Client-Side vs Server-Side)
- Routing Modules: forRoot() and forChild()
- Route Parameters: Snapshot vs Observable Ways
- Redirects, Wildcard Routes, Child / Nested Routes
- Route Guards: CanActivate, CanActivateChild, CanDeactivate, Resolve, CanLoad
- ActivatedRouteSnapshot & RouterStateSnapshot
- Named / Auxiliary Router Outlets
- Programmatic Navigation: Navigate, NavigateByUrl, Query Parameters

Chapter 17: Angular Modules

- Root Module & @NgModule Metadata Properties
- Feature Modules, Eager Loading vs Lazy Loading
- Preloading Strategies & Custom Delay Strategies
- Importing Core Modules (BrowserModule, CommonModule)

Chapter 18: Session Management

- HTML Web Storage: localStorage & sessionStorage
- CRUD Operations, Storing JSON Objects, Checking Support & Iteration

Chapter 5: Components

- What is a Component & Standalone Component?
- Comparing Module-based vs Standalone Components
- Component Class, Application Root Component
- CLI Creation & Deep Dive into @Component Decorators:
 - template, templateUrl, styles, styleUrls
 - preserveWhitespaces, viewProvider, changeDetection
 - moduleId, interpolation, encapsulation
 - entryComponents, animations, selector
 - Input, Output, Providers (useClass, useExisting, useValue, useFactory)
 - exportAs, queries, host, jit properties
- Component Lifecycle Hooks & Constructor vs ngOnInit
- Dynamically Loading Components in Angular

Chapter 6: Introduction to TypeScript and ES6

- JavaScript vs. TypeScript, Transpilation & Features
- Advantages and Disadvantages over JavaScript
- Type Annotations, Variable Declarations (var, let, const)
- Data Types: Primitive, Any, Void, Null, Undefined, Arrays, Tuples, Enums, Classes, Interfaces
- Decision Making (if, else if, switch) & Loops (for, while, do-while)
- Access Modifiers (public, private, protected)
- Functions: Named, Anonymous, Optional/Default Parameters, Arrow Functions, Rest Parameters
- Generics, Decorators & TypeScript Type System

Chapter 7: Directives

- Overview: Component, Structural & Attribute Directives
- Custom Structural Directives (TemplateRef, ViewContainerRef, Component/Host/Embedded Views)
- Custom Attribute Directives (ElementRef, Renderer)
- Property Binding with Custom Directives

Chapter 8: Structural Directives

- Modern Control Flow Syntax (@if, @for, @switch) vs Traditional (NgIf, NgFor, NgSwitch)
- @if with HTML, Null Check, Enum, @else, @elseif, Async Pipe
- @for with @empty, index/even/odd, first/last, Async Pipe, trackBy
- @switch with Enums, Classes, and HTML Elements
- Detailed ng-template, NgIf-Then-Else expanding patterns

Chapter 19: RxJS

- Observables, Subscriptions & In-Memory Web API
- Observable with Async Pipe, ngFor, ngIf, and map operator
- Observables vs Promises
- RxJS Subjects, BehaviorSubject, and ReplaySubject

Chapter 20: HTTP Client

- HttpClientModule, Service Setup & HTTP vs HttpClient
- GET, POST, PUT, DELETE Request Patterns
- HttpParams, HttpHeaders, responseType, reportProgress, withCredentials
- Consuming JSON and Text Data via Subscriptions & Async Pipes
- Error Handling & RxJS retry() Operator

Chapter 21 & 22: Template Driven Forms

- Template-Driven vs Reactive Forms Comparison
- FormsModule, NgForm, NgModel Integration
- Form Validation States: ng-touched, ng-untouched, ng-pristine, ng-valid, ng-invalid
- Angular Validators, Displaying State, Disabling Submissions
- Handling Checkboxes, Select Dropdowns, Date, and Radio Buttons

Chapter 23: Model Driven (Reactive) Forms

- Reactive Forms Architecture: FormControl, FormGroup, FormBuilder
- formControlName Binding, Value Access, reset(), setValue(), patchValue()
- valueChanges & statusChanges Observables
- FormArray & Nested FormArrays / FormGroups
- Reactive Forms with Angular Material & Custom Validators

Chapter 24: Angular Material, Bootstrap & Utilities

- Angular Material: Features, Installation, Theming & Modules
- Bootstrap & jQuery Integration in Angular
- Cookie Management using ngx-cookie-service

Chapter 25: Single Page Applications (SPA)

- SPA Architecture vs Traditional Web Applications
- SPA Advantages, Challenges & Implementation via Angular Router

Chapter 9: Attribute Directives

- ngStyle and Alternative Syntaxes
- ngClass with Strings, Arrays, and Objects
- Combining NgClass with NgFor, NgSwitch, and NgIf

Chapter 10: Data and Event Binding

- Data Binding Classification & Syntax
- One-Way Data Binding: Interpolation, Attribute, Style, Class, Property Binding
- Property Binding vs Interpolation
- Event Binding & Two-Way Data Binding Mechanics

Chapter 11: Signals (Angular 20 Core)

- What are Signals? Creating, Reading, and Updating Signals
- Computed Signals & Effects
- Signals vs RxJS Observables
- Signals in Templates & Components
- Best Practices, Use Cases & Limitations

Chapter 12: SSR (Server-Side Rendering)

- What is Angular SSR & Angular Universal
- Architecture: How SSR Works in Angular
- Setting Up SSR & Server Entry Points
- Client vs Server Code Execution & Hydration
- Pre-rendering, TransferState API, SSR Deployment
- SEO Best Practices & SSR Challenges

Chapter 13: Pipes and Data Formatting

- What are Pipes & ngNonBindable Usage
- Built-in Pipes: Lowercase, Uppercase, Titlecase, Slice, Json, Decimal, Percent, Currency, Date
- Parameterized Pipes & Pipe Chaining
- Async Pipe Deep Dive
- Creating Custom Pipes & Chaining Custom Pipes
- Pure vs Impure Pipes with Practical Examples

Chapter 14: Decorators

- Class & Property Decorators: @NgModule, @Component, @Injectable, @Directive, @Pipe
- Input/Output Decorators & EventEmitter (.emit())
- DOM Decorators: @HostBinding, @HostListener
- View Queries: @ContentChild, @ContentChildren, @ViewChild, @ViewChildren

Chapter 26: REST Web Services in Angular

- Understanding REST: CRUD Operations with JSON
- Angular Service Classes with HttpClient & RequestOptions
- URL Path Parameters, Query Parameters, Headers & API Security

Chapter 27: Debugging

- Debugging Practices, Development Mode & Element Inspection
- TypeScript Breakpoints & Debugging Workflows
- Troubleshooting Common Exceptions (ngModel, router-outlet, package.json, Missing Providers)

Chapter 28: Angular Style Guide

- Official Angular Style Guide & Naming Conventions
- Single Responsibility Principle & Application Structure

Chapter 29 & 30: JWT Security & 3rd-Party Packages

- JWT Authentication Workflow & Adding Tokens to Headers
- Angular HTTP Interceptors for Authentication
- CSS, SCSS, SASS Styling
- Integrating Feather Icons, Font-Awesome & Toastr Notifications

