

Angular 5

- **Table of Content**

- Introduction to Angular (Day 1)
- Architecture Overview (Day 1)
- TypeScript (Day 1)
- Setting up Development Environment (Day 1)
- Components & Templates (Day 1)
- Data Binding (Day 2)
- Directives (Day 2)
- Pipes (Day 2)
- Services & Dependency Injection (Day 2)
- Forms & Validation (Day 3)
- Building Single Page Apps using Routing (Day 3)
- Reactive Programming using RxJS Observables (Day 3)
- Server Communication (Day 4)
- Angular Modularity (ngModules) (Day 4)
- Unit Testing Angular Apps (Day 4)
- Introduction to webpack (Day 5)
- State Management using NgRx (Day 5)
- Progressive Web Apps (Day 5)

- **Detailed Content**

- **Introducing Angular**
 - ✓ What is Angular?
 - ✓ Advantages of Angular
 - ✓ Where does Angular fit within a modern web app?
 - ✓ Traditional web app – Request & response
 - ✓ Angular app – Request & response
- **Architecture overview**
 - ✓ Introduction to key building blocks of Angular
- **TypeScript**
 - ✓ What is TypeScript?
 - ✓ Why TypeScript?
- **Setting up Development Environment**
 - ✓ Introduction to Angular CLI
 - ✓ Setting up Angular
 - ✓ Creating an app using Angular CLI
 - ✓ Setting up Bootstrap for styling
 - ✓ How an Angular app gets loaded and started?
- **Components & Templates**
 - ✓ What is a Component? What are its benefits?
 - ✓ The Root component
 - ✓ What are Decorators?
 - ✓ Understanding the component decorator
 - ✓ Creating and using components
 - ✓ Component templates
 - ✓ Component styles
 - ✓ Lifecycle Hooks
- **Data Binding**
 - ✓ What is Data Binding?
 - ✓ Interpolation
 - ✓ Property binding
 - ✓ Event binding
 - ✓ Passing and using event data
 - ✓ Two-way data binding
 - ✓ Component interaction
 - Parent to child interaction
 - Child to parent interaction

- Directives
 - ✓ Understanding Directives
 - ✓ Types of Directives
 - Attribute Directives
 - Structural Directives
 - ✓ Creating custom directives
- Pipes
 - ✓ Introduction to Pipes
 - ✓ Using pipes
 - ✓ Parameterizing pipes
 - ✓ Chaining multiple pipes
 - ✓ Creating custom pipes
- Services & Dependency Injection
 - ✓ Need for a Service
 - ✓ Creating a service
 - ✓ Understanding Dependency Injection (DI) and its benefits
 - ✓ Using a service within a component
 - ✓ Using a service within another service
 - ✓ Cross component interaction using a service
- Forms & Validation
 - ✓ Template-driven forms vs Reactive forms
 - ✓ Building a form
 - ✓ Registering form controls
 - ✓ Submitting the form
 - ✓ Understanding form state
 - ✓ Adding form validation
 - ✓ Outputting Validation Error messages
 - ✓ Using two-way binding
 - ✓ Grouping form controls
 - ✓ Resetting forms
- Building Single Page Apps using Routing
 - ✓ Need for a Router
 - ✓ Setting up and loading routes
 - ✓ Navigating with router links
 - ✓ Styling active links
 - ✓ Navigating programmatically
 - ✓ Passing parameters to routes
 - ✓ Fetching route parameters

- ✓ Passing query parameters
- ✓ Retrieving query parameters
- ✓ Setting up nested routes

- Reactive Programming using RxJS Observables
 - ✓ Introduction to Reactive Extensions (RxJS)
 - ✓ Observables
 - ✓ Creating Observables
 - ✓ Using Observable operators
 - ✓ Transforming Observables
 - ✓ Cancelling Subscriptions

- Server Communication
 - ✓ Introduction to Angular HttpClient service
 - ✓ Sending requests to server
 - ✓ Getting data from the server
 - ✓ Sending data to the server
 - ✓ Interceptors
 - ✓ Handling Http errors

- Angular Modularity (ngModules)
 - ✓ Why modules?
 - ✓ Introduction to Angular Modules
 - ✓ @NgModule decorator
 - ✓ Routing module
 - ✓ Feature module
 - ✓ Sharing module

- Unit Testing Angular Apps
 - ✓ Fundamentals of Unit Testing
 - ✓ Set Up and Tear Down
 - ✓ Spies
 - ✓ Code Coverage
 - ✓ Angular Testing Utilities - TestBed
 - ✓ Working with Components
 - ✓ Testing Property & Event Bindings
 - ✓ Handling Component Dependencies
 - ✓ Testing Async Operations

- Introduction to webpack
 - ✓ Why build tools?
 - ✓ Modularization in modern Web applications
 - ✓ Challenges with modularization

- ✓ Core purpose of webpack
- ✓ Core concepts
- ✓ webpack in action

- State Management using NgRx
 - ✓ Introduction to Redux
 - ✓ Why Redux?
 - ✓ Core concepts
 - ✓ The Store
 - ✓ Reducers
 - ✓ Actions & Action Creators

- Progressive Web Apps
 - ✓ What are PWAs?
 - ✓ App Shell
 - ✓ Service Worker
 - ✓ Web App Manifest
 - ✓ Web Push Notifications
 - ✓ Converting an Angular app into a PWA