Cognixia®

COURSE CONTENT

www.cognixia.com

COURSE NAME

Angular 17 With Electron Training

COURSE OVERVIEW

- Understand how single-page web application architectures are different than traditional web application architectures
- Use new JavaScript (ES6) language features including Classes, Modules, and Arrow Functions
- Use new TypeScript language features including Types, Decorators, Interfaces, and Generics
- Learn Angular coding and architecture best practices including project layout and using container and presentation components
- Understand and use Angular model-driven forms, observables, dependency injection, and routing
- Electron framework for building desktop Apps.
- Communicate with a backend server using Angular's HttpClient to load and save data
- Configure the router and navigate between components
- Unit test all parts of an application including Components, Services, and Pipes
- Understand RxJS and Observables and where they can be used
- Implement Authentication and Authorization in an Angular Application
- Optimize Angular Performance by changing Change Detection Strategies
- Setup new projects from scratch using the Angular CLI
- Scaffold modules, components, services, models, routes, and unit tests in accordance with best practices using the Angular CLI
- Build and deploy an application to production using the Angular CLI
- Write End-to-End Tests (optional; taught only if this applies to your group)

DURATION

5 Days

PREREQUISITES

- JavaScript and HTML

SYSTEM REQUIREMENT

- VS Code
- NodeJS ver 20
- Chrome browser
- Windows machine with min 12GB of RAM

COURSE OUTLINE

Day - 1

- Introduction
- TypeScript and ECMAScript 6 (ES6) Fundamentals
  - TypeScript Installation, Configuration & Compilation
  - Type Annotations
  - Classes
  - Scoping using let, var, and const Keywords
  - Arrow Functions
  - ES Modules
  - Decorators
  - Template Literals
  - Spread Syntax and Rest Parameters
  - Destructuring
- Angular Overview
  - Benefits of Building using Angular
  - Understanding Angular Versions
  - Single-page Web Application Architectures vs. Traditional Server-side Web Application Architectures
  - Angular Style Guide
  - Angular Architecture
  - Angular Compared to Other JavaScript Libraries and Frameworks (React, VueJS, etc.)
  - Your First Angular Application
- Components
  - Understanding Components
  - Component Properties & Methods
  - Templates: Inline, Multi-line, and External with Component-relative Paths
- Angular Modules (NgModule)
  - Angular Modules vs. ES Modules
  - Organizing your code into Feature Modules
- Project Set-Up (Using the Angular CLI)
  - Angular CLI Features
  - Creating a New Project
  - Generating Code
  - Customizing the Angular CLI

- Data Binding
  - Interpolation
  - Property binding
  - Event binding
  - Two-way data binding

Day - 2

- Directives
  - Structural: ngFor, ngIf, ngSwitch
  - Attribute: ngClass, ngStyle
- Pipes
  - Built-in Pipes: Using, Passing Parameters, Chaining
- Advanced Components
  - Component Communication using @Input, @Output
  - Component Architecture
  - Component Styles
  - Component Lifecycle Hooks
  - Evaluating UI Component Frameworks & Libraries
- Services & Dependency Injection
  - Using a service to access data
  - Using a service to encapsulate business logic
  - Understanding the scope of services

Day - 3

- Dependency Injection
  - Understanding Dependency Injection
  - Angular's Dependency Injection System
  - Registering
  - Injecting
- Model-driven Forms (Reactive Forms)
  - Importing the ReactiveFormsModule
  - FormControl, FormGroup, and AbstractControl
  - Binding DOM Elements to FormGroups and FormControls
  - Validation Rules, Messages, and Styles
  - Refactoring Reactive Forms for Reuse
  - Custom Validators
- Communicating with the Server using the HttpClient Service
  - Deciding between Promises or Observables (RxJS)
  - Making an HTTP GET Request
  - Sending data to the server using Http POST and PUT Requests

- Issuing an Http DELETE Request
- Intercepting Requests and Responses

- Router
  - Importing the RouterModule
  - Configuring Routes
  - Displaying Components using a RouterOutlet
  - Navigating declaratively with RouterLink
  - Navigating with code using the Router
  - Accessing parameters using ActivatedRoute

Day - 4

- Deploying an Angular Application to Production
  - Building the application using the Angular CLI
  - Deploying to a web server
  - Angular Material UI
- Angular Roadmap for the Future
  - Ivy Renderer
  - Angular Elements
- Unit Testing
  - Tools: Jasmine, Karma
  - Jasmine Syntax: describe, it, beforeEach, afterEach, matchers
  - Setup and your First Test
  - Testing Terminology: Mock, Stub, Spy, Fakes
  - Angular Testing Terminology: TestBed, ComponentFixture, debugElement, async, fakeAsync, tick, inject
  - Simple Component Test
  - Detecting Component Changes
  - Testing a Component with properties (inputs) and events (outputs)
  - Testing a Component that uses the Router
  - Testing a Component that depends on a Service
  - Testing a Service and Mocking its Http requests
  - Testing a Pipe
- RxJS and Observables
  - What is an Observable?
  - Creating Observables
  - What is an Observer?
  - Observer Example
  - Operators: map, switchMap, debounceTime, distinctUntilChanged
  - Practical Application of using RxJS
  - Subject

- Subject Example
- EventEmitter or Observable

- Working with NGRX
- What is Electron
  - What Electron precisely does?
  - What is an Electron app?
  - Creating a project
  - Installing & setting up Electron
  - Calling Electron APIs by example
  - Packaging your Electron ap
  - Create app windows with BrowserWindow
  - Create menus with custom items
  - Menu roles
  - USB Drivers
  - GRPCS
  - Shell module to open files and folders
  - App events
  - Main process & Renderer process
  - Creating The Interface
  - Renderer Node Integration
  - IPC Communication
  - Implementing Imagemin & Plugins
  - Send Events To Renderer
  - Creating Log Files

Day - 5

- Security
  - Best Practices
  - Preventing Cross-site Scripting (XSS)
  - Trusting values with the DOMSanitizer
  - HTTP Attacks (CSRF and CSSI)
  - Authentication using JSON Web Tokens (JWT)
  - Authorization: Router Guards
- Change Detection
  - Understanding Zone.js and Change Detection
  - Change Detection Strategies Default and OnPush
- Advanced Routing
  - Lazy-loading Angular Modules
  - Nested or Child Routes

- Advanced Dependency Injection
  - Providers
  - Hierarchical Injection
- Pipes
  - Creating a custom Pipe using PipeTransform
  - Understanding Pure and Impure Pipes
- Working With ChartJS and D3JS
- Conclusion