

Document Generated: 12/18/2025

Learning Style: Virtual Classroom

Technology: Google

Difficulty: Beginner

Course Duration: 3 Days

Next Course Date: **February 9, 2026**

Introduction to Angular 20 Essentials (TT4165)



About this course:

Geared for experienced web developers, our Introduction to Angular Essentials 20 is a three-day, comprehensive hands-on program that explores the latest features and benefits Angular has to offer. Throughout the course you'll learn the practical "real-world" foundational and advanced Angular skills required to solve the usual challenges you might face when developing modern web applications with TypeScript.

Kicking off with an in-depth introduction to Angular CLI and TypeScript, the course propels you into the world of component-driven development, touching on vital topics like data-binding, directives, and services. As you advance through the curriculum, you'll explore SPA routing and Reactive forms.

By the end of this course, you will have built a comprehensive application that brings together all these elements: website structuring through components, dynamic interactivity via directives, and custom pipes for content refinement. You will know how to employ reactive forms for active user engagement and SPA routing for fluid navigation.

Course Objective:

The lab intensive course includes plenty of hands-on lab work designed to immerse you in the skills required to get you up and running with Angular right away. The course is rich with hands-on activities, challenge labs, knowledge checks, valuable discussions and focused projects.

Guided by our engaging, highly-experienced web development instructor, you will learn how to:

- **Build Powerful Components:** Master creating and organizing Angular components, leveraging dependency injection and lifecycle hooks for scalable apps.
- **Harness TypeScript Power:** Become proficient in TypeScript, using its advanced features like classes, interfaces, and decorators to supercharge your Angular development.
- **Create Dynamic, Reactive Forms:** Learn to build complex forms with Angular reactive approach, ensuring seamless validation and user interaction.
- **Optimize with CLI and DevTools:** Use Angular CLI for easy project setup and deployment, and harness DevTools for debugging and optimizing your applications.
- **Master SPA Navigation:** Implement smooth and efficient Single Page Application (SPA) routing with nested routes and route guards for a top-

notch user experience.

- Utilize Third-Party Libraries: Learn to effectively integrate and manage third-party libraries within Angular to extend functionality and streamline development.

Audience:

- The content is geared for experienced web developers new to Angular. In order to be successful in the course, you should have solid, current hands-on experience developing basic web applications, and be well versed in HTML5, CSS3 and JavaScript.

Prerequisite:

Attendees should have experience with the topics in the following courses, or should have recently attended these as a prerequisite:

- Introduction to HTML5, CSS3 and JavaScript
- Introduction to TypeScript: Clean Code and Strong Skills for Web Developers

Course Outline:

1. Overview of Angular Architecture

- Angular Versioning
- Data Binding
- Component Architecture
- - Show concepts: how to render the content in the browser
- Components and Dependency Injection
- Services and Events
- Common Component Lifecycles and Hooks

2. TypeScript

- Angular, ES6 and TypeScript
- Typing and Classes
- Abstract Classes and Interfaces
- Interface patterns
- Annotations
- Generics
- Optional Chaining
- Nullish coalescing

- Decorators
- Functional vs Procedural JavaScript
- Functional Approach
- Lab: TypeScript

3. Bootstrapping with Angular CLI

- Angular CLI Overview
- New Projects with CLI
- Testing and Generating with CLI

4. Angular Basics

- Configuration Files
- Working with angular.json
- Top-Level Directories
- Contents of app folder
- Angular DevTools
- ng build
- Lab: Creating Projects with Angular CLI
- Working with Angular

5. Components and Events

- Data Binding and Components
- Event Binding
- Custom Events
- Parent/Child Events
- EventEmitter/emit()
- New Input/Output function
- Lab: Nested Components

6. Standalone Components

- Project Setup and Configuration
- Folder Structure
- Components
- Services
- Routing
- Versus non-standalone components– overview
- Versus modules – overview

7. Third Party Libraries

- Angular and the Need for Libraries
- Options for adding Libraries
- Working with npm install
- Bootstrap.css
- Tailwind CSS
- Lab: Third-Party Libraries

8. Dynamic Views - Directives

- View Encapsulation
- Structural/Attribute Directives
- Conditional Styling
- @if/@else
- @for
- @defer
- Lab: Working with Directives and control flow blocks

9. Pipes

- Overview of Pipes
- Built-in Pipes
- Formatting and Conversions
- Parameterizing Pipes
- Pure vs. Impure Pipes
- Customizing Pipes
- Lab: Built-in and Custom Pipes

10. Signals and State

- What are Signals?
- How to read the value of a signal?
- How to modify the value of a signal?
- The update signal API
- Read-only signals
- The computed() Signal API
- How do we subscribe to a signal?
- Using effects
- Lab: Signals
- Angular Forms

11. Forms and the Forms API focus on Reactive Forms

- Template Forms Overview
- User Input/Interactions
- Model-Driven Reactive Forms
- FormBuilder API
- Reactive Form Validation
- Forms new control state change events
- Lab: Reactive Forms
- Single Page Applications and Routes

12. SPA - Routing

- Overview of the SPA Concept
- Routing for Page Display
- Working with ActivatedRoute
- Location Strategies

- Nested Routes
- Route redirects as functions
- Lab: Routing

Bonus Content / Time Permitting

13. Services (Dependency Injection)

- Angulars DI Framework
- Components and Injectables
- Tree-shakeable providers
- Services Architecture
- Service Interfaces and DI
- Constructor Injection
- Inject function
- Scenarios for use
- Helper Services
- Fascade Services
- Lab: Services

14. Modules

- Overview of Modules
- Feature Modules
- Feature Module Routing
- Shared Modules
- Lab: Modules

15. Standalone Components vs Modules

- Review of Standalone components
- A non-standalone component
- Using standard directives in standalone components
- practical benefits of standalone components
- Using standalone components in NgModule-based components
- Using feature NgModule-based components in standalone components
- Lazy loading with standalone components
- 3 steps to migrating to standalone components
- Lab: Using Standalone components alongside modules