Course Outline

Advanced Angular 11

Course Summary

Description

Take your skills to the next level with Advanced Angular training. Students gain an understanding of application architecture and design best practices in Angular, as well as learn how to authenticate, unit test, and manage application state in an Angular application.

Note: This course is appropriate for all versions of Angular since version 2 and is current through Angular 11.

Objectives

After taking this course, students will be able to:

- 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)
- Upgrade an existing application from AngularJS to Angular 11 (optional; taught only if this applies to your group)

Topics

- Introduction
- Unit Testing
- RxJS and Observables
- Security
- Change Detection
- Advanced Angular CLI
- Advanced Routing
- Advanced Routing
- Advanced Dependency Injection
- Pipes
- Conclusion

Prerequisites

Students should have programming experience with an object-oriented language. In addition, some experience with JavaScript is helpful, but the new language features of JavaScript and TypeScript are covered/reviewed in class.

Duration

Two Days

Course Outline

Advanced Angular 11

Course Outline

I. *Introduction*

II. *Unit Testing*
- A. Tools: Jasmine, Karma
- B. Jasmine Syntax: describe, it, beforeEach, afterEach, matchers
- C. Setup and your First Test
- D. Testing Terminology: Mock, Stub, Spy, Fakes
- E. Angular Testing Terminology: TestBed, ComponentFixture, debugElement, async, fakeAsync, tick, inject
- F. Simple Component Test
- G. Detecting Component Changes
- H. Testing a Component with properties (inputs) and events (outputs)
- I. Testing a Component that uses the Router
- J. Testing a Component that depends on a Service
- K. Testing a Service and Mocking its HTTP requests
- L. Testing a Pipe

III. *RxJS and Observables*
- A. What is an Observable?
- B. Creating Observables
- C. What is an Observer?
- D. Observer Example
- E. Operators: map, switchMap, debounceTime, distinctUntilChanged
- F. Practical Application of using RxJS
- G. Subject
- H. Subject Example
- I. EventEmitter or Observable

IV. *Security*
- A. Best Practices
- B. Preventing Cross-site Scripting (XSS)
- C. Trusting values with the DOMSanitizer
- D. HTTP Attacks (CSRF and CSSI)
- E. Authentication using JSON Web Tokens (JWT)
- F. Authorization: Router Guards

V. *Change Detection*
- A. Understanding Zone.js and Change Detection
- B. Change Detection Strategies Default and OnPush

A. *Advanced Angular CLI*
- B. Customizing a build using Builder APIs in the CLI
- C. Generating web workers

VI. *Advanced Routing*
- A. Lazy-loading Angular Modules (using Dynamic Imports)
- B. Nested or Child Routes

VII. *Advanced Routing*
- A. Lazy-loading Angular Modules
- B. Nested or Child Routes

VIII. *Advanced Dependency Injection*
- A. Providers
- B. Hierarchical Injection

IX. *Pipes*
- A. Creating a custom Pipe using PipeTransform
- B. Understanding Pure and Impure Pipes

X. *Conclusion*

Advanced Angular 11

Course Outline (cont.)

Choose any two optional topics. If desired, the course can be customized to include more than two of these topics if other topics are scaled back or removed.

- npm QuickStart
  - Installing Dependencies
  - Understanding package.json and package-lock.json
  - Using npm as a Build Tool
- Managing Shared Application State using ngrx and Redux
  - Benefits Overview
  - Three Principles of Redux: Single Source of Truth, State is Read-Only, Pure Functions
  - Examples of Pure Functions
  - Reducers
  - Simple ngrx Example
  - Time-traveling with Redux Devtools
  - Full ngrx Example Application
- Upgrade Strategies from AngularJS
  - High-level Approaches
  - Concept Mapping AngularJS to Angular
  - UpgradeAdapter
  - What can be Upgraded or Downgraded
  - What cannot be Upgraded or Downgraded
  - UpgradeAdapter and Dependency Injection
- End-to-End Testing
  - What is Protractor?
  - Why Protractor?
  - Using Locators
  - Page Objects
  - Debugging E2E Tests