

Advanced Angular 6

Course Summary

Description

Build applications with the user experience of a desktop application and the ease of deployment of a web application using Angular. Start from scratch by learning the JavaScript package manager (npm), the language (TypeScript), and the tools. Use those foundational skills to implement an Angular application using components, templates, services, routing and tests. Along the way, understand best practices as well as the architecture and design of Angular applications.

Objectives

At the end of this course, students will be able to:

- 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
- 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)
- Upgrade an existing application from AngularJS to Angular 6 (optional; taught only if this applies to your group)

Topics

- Unit Testing
- RxJS and Observables
- Security
- Change Detection Strategies
- Advanced Routing
- Advanced Dependency Injection
- Pipes
- Angular CLI
- Deploying an Angular Application to Production
- npm QuickStart
- Managing Shared Application State using ngrx and Redux
- Upgrade Strategies from AngularJS
- End-to-End Testing
- Template-driven Forms
- Conclusion

Advanced Angular 6 Course Summary (cont'd)

Audience

This course is designed for those wanting to learn how to build an application from scratch using Angular 6.

Prerequisites

All Angular training students must have substantial prior experience developing with JavaScript. If attendees will not have prior JavaScript experience, we would be delighted to precede this class with a one- or two-day intensive JavaScript primer.

Duration

Two days

Advanced Angular 6

Course Outline

I. 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 using a Spy
- K. Testing a Component that depends on a Service using a Fake
- L. Testing a Service and Mocking its Http requests
- M. Testing a Pipe

II. 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

III. 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

IV. Change Detection Strategies

- A. Default
- B. OnPush

V. Advanced Routing

- A. Lazy-loading Angular Modules
- B. Location Strategies
- C. Nested or Child Routes

VI. Advanced Dependency Injection

- A. Providers
- B. Using the @Optional and @Host Decorators

VII. Pipes

- A. Built-in Pipes: Using, Passing Parameters, Chaining
- B. Creating a custom Pipe using PipeTransform
- C. Understanding Pure and Impure Pipes

VIII. Angular CLI

- A. Creating a New Project
- B. Generating Modules, Components, Models, and Services
- C. Customizing the CLI to include CSS Preprocessors (SASS) and other UI libraries: Bootstrap, Material, or Clarity

IX. Deploying an Angular Application to Production

- A. Manually
- B. Using the Angular CLI

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

X. npm QuickStart

- A. Installing Dependencies
- B. Understanding package.json and package-lock.json
- C. Using npm as a Build Tool

Advanced Angular 6 Course Outline (cont'd)

XI. Managing Shared Application State using ngrx and Redux

- A. Benefits Overview
- B. Three Principles of Redux: Single Source of Truth, State is Read-Only, Pure Functions
- C. Examples of Pure Functions
- D. Reducers
- E. Simple ngrx Example
- F. Time-traveling with Redux Devtools
- G. Full ngrx Example Application

XII. Upgrade Strategies from AngularJS

- A. High-level Approaches
- B. Concept Mapping AngularJS to Angular
- C. UpgradeAdapter
- D. What can be Upgraded or Downgraded
- E. What cannot be Upgraded or Downgraded
- F. UpgradeAdapter and Dependency Injection

XIII. End-to-End Testing

- A. What is Protractor?
- B. Why Protractor?
- C. Using Locators
- D. Page Objects
- E. Debugging E2E Tests