

Advanced Angular

Course Summary

Description

Angular is a powerful client-side JavaScript framework from Google that supports simple, maintainable, responsive and modular applications. It uses modern web platform capabilities including ES6 to deliver app-like experiences with zero-step installation. Applications are architected by combining modular, reusable UI web components. Angular facilitates productivity with automatic data binding via a simple and powerful template syntax as well as rich tooling support in numerous IDEs (including autocomplete, navigation and refactoring). The ability to extend HTML to include custom tags with behavior for application building is a powerful idea and among the many reasons that Angular is so widely used. Angular has become a platform that allows for one code base across web apps, native mobile apps and desktop apps.

Angular training teaches developers how to use the newest version of Angular to facilitate development of app-like experiences with zero-step installation. This is a two day course.

Prior to the class, students need to contact the sales representative to obtain the necessary files for this course.

Objectives

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

- Unit Testing
- Angular Migration Strategies
- Angular CLI
- End-to-end Testing with Protractor
- Advanced Custom Directives & Components
- Model-driven Forms
- Custom Pipes

Topics

- Advanced Directives and Components
- Unit Testing
- Custom Pipes
- Model-Driven Forms
- Angular Migration Strategies
- Preparation
- Angular CLI
- End-to-End Testing with Protractor

Audience

This course is designed for experienced web developers.

Prerequisites

Before taking this course, students should have the following skills:

- Prior experience developing with JavaScript.
- Experience with Angular equivalent to Introduction to Angular 2 class.

Duration

Two days

Advanced Angular

Course Outline

I. Advanced Directives and Components

- A. Access and update the DOM using ElementRef and Renderer
- B. Respond to User Events using HostListener
- C. Set properties on the host element using HostBinding
- D. Content Projection/Transclusion
 - 1. @ViewChild(ren), @ContentChild(ren)
 - 2. ng-content

II. Unit Testing

- A. Tools: Jasmine & Karma
- B. Mocks, Stubs, Fakes, and Spies
- C. Angular Testing
- D. TestBed, ComponentFixture, and Debug Element
- E. async, fakeAsync, tick, and inject
- F. Your First JavaScript Test
- G. Testing a Simple Component
- H. Detecting Changes
- I. Using External Templates
- J. Components with Inputs and Outputs
- K. Component with Router
- L. Component with Service
- M. Testing a Service in Isolation
- N. Mocking HTTP Calls
- O. Testing Pipes

III. Custom Pipes

- A. Custom Pipe Example
- B. Using a Custom Pipe
 - 1. In Templates
 - 2. In Code
- C. Pure and Impure Pipes

IV. Model-Driven Forms

- A. Setup/Bootstrap
- B. Model
- C. Form Component
- D. Metadata
- E. Template

V. Angular 2 Migration Strategies

- A. Preparation
 - 1. Following The Angular Style Guide
 - 2. Using a Module Loader
 - 3. Migrating to TypeScript
 - 4. Using Component Directives
- B. Upgrading with The Upgrade Adapter
 - 1. How The Upgrade Adapter Works
 - 2. Bootstrapping Hybrid Angular Applications
 - 3. Using Angular Components from Angular Code
 - 4. Using Angular Component Directives from Angular Code
 - 5. Projecting Angular 1 Content into Angular 2 Components
 - 6. Transcluding Angular 2 Content into Angular 1 Component Directives
 - 7. Making Angular Dependencies Injectable to Angular
 - 8. Making Angular Dependencies Injectable to Angular

VI. Angular CLI

- A. Installation
- B. Usage
- C. Generating a New Project
- D. Generating Components, Directives, Pipes and Services
- E. Generating a Route
- F. Creating a Build
- G. Build Targets and Environment Files
- H. Bundling

VII. End-to-End Testing with Protractor

- A. Setup
- B. Demo