

## Advanced Angular 5

### Course Summary

#### Description

Advanced Angular 5 teaches students the skills and best practices they need to design, build, test, and deploy applications that provide rich end-user experiences similar to a desktop application while still offering the ease of deployment of a web application.

#### Objectives

After taking this course, students will be able to:

- Unit test all the parts of an Angular application including Modules, Components, Services, and Pipes
- Upgrade an existing application from AngularJS to Angular 5 over time by running both frameworks in the same project
- Start a new Angular project and scaffold modules, components, services, models, routes, and unit tests following best practices using the Angular CLI
- Build and deploy an Angular application including combining and minifying JavaScript and HTML files, Tree-shaking unused code, and doing Ahead-of-Time compilation to reduce the size of the Angular framework.
- Write End-to-End Tests if your application with Protractor which uses Selenium Web Driver
- Using Redux and NgRx to maintain the state in your application

#### Topics

- Unit Testing
- Security
- Advanced Components
- Advanced Routing
- Advanced Dependency Injection
- Attribute Directives
- Pipes
- Creating, Building, and Deploying an Angular Application
- Template-driven Forms
- Upgrade Strategies from AngularJS
- Redux
- End-to-End Testing
- npm QuickStart
- Webpack Guide
- Conclusion

#### Audience

This course is designed for those wanting to learn how to build an application from scratch using Angular 5.

#### Prerequisites

Before taking this course, attendees must have substantial prior experience developing with JavaScript and have taken the Introduction to Angular 5 course (PT20331).

#### Duration

Two days

## Advanced Angular 5

### 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, Fake
  - 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. Security**
  - A. How to Prevent Cross-site Scripting (XSS)
  - B. Trusting values with the DOMSanitizer
  - C. HTTP Attacks
  - D. Security Audits of Angular Applications
- III. Advanced Components**
  - A. Component Styles
    - 1. using Metadata properties: Styles and StyleUrls
    - 2. Encapsulation Strategies
  - B. Change Detection Strategies
  - C. Component Lifecycle Hooks
- IV. Advanced Routing**
  - A. Lazy-loading Angular Modules
  - B. Location Strategies
  - C. Nested or Child Routes
  - D. Route Guards
- V. Advanced Dependency Injection**
  - A. Providers
  - B. Using the @Optional and @Host Decorators
- VI. Attribute Directives**
  - A. Creating a custom Attribute Directive using ElementRef, Render
- VII. Pipes**
  - A. Built-in Pipes: Using, Passing Parameters, Chaining
  - B. Creating a custom Pipe using PipeTransform
  - C. Understanding Pure and Impure Pipes
- VIII. Creating, Building, and Deploying an Angular Application**
  - A. Manually
  - B. Using the Angular CLI
    - 1. Overview
    - 2. Features
    - 3. Installation
    - 4. Generating a New Project
    - 5. Generating Code
    - 6. Builds
    - 7. Customizing Builds
    - 8. Angular Material Setup
    - 9. Eject
- IX. Template-driven Forms**
  - A. NgSubmit Directive
  - B. FormsModule
  - C. NgForm, NgModel, and NgModelGroup Directives
  - D. Validation Directives
    - 1. Displaying validation messages
    - 2. Styling validation messages
- X. Upgrade Strategies from AngularJS**
  - A. Preparing your AngularJS Project
    - 1. Integrating a Module Loader
    - 2. Start using TypeScript
    - 3. Use Components instead of Controllers
  - B. Angular 5 and AngularJS together
    - 1. Understanding the Upgrade Module
    - 2. Angular (Angular 5) Components in AngularJS Code
    - 3. AngularJS Directives in Angular Code
    - 4. Injecting AngularJS Services into Angular
    - 5. Injecting Angular Services into AngularJS
    - 6. Upgrade from AngularJS Router to Angular Router

## Advanced Angular 5

### Course Outline (cont'd)

- |   |  |
|---|--|
| <b>XI.     Redux</b> <ul style="list-style-type: none"><li>A.    Redux Basics</li><li>B.    Debugging and Time Traveling with Redux DevTools</li></ul>  | <b>XIII.    npm QuickStart</b> <ul style="list-style-type: none"><li>A.    Installing Dependencies Locally</li><li>B.    Using npm as a Build Tool</li></ul>   |
| <b>XII.    End-to-End Testing</b> <ul style="list-style-type: none"><li>A.    What is Protractor?</li><li>B.    Why Protractor?</li><li>C.    Using Locators</li><li>D.    Page Objects</li><li>E.    Debugging E2E Tests</li></ul> | <b>XIV.    Webpack Guide</b> <ul style="list-style-type: none"><li>A.    Installation</li><li>B.    Building/Bundling<ul style="list-style-type: none"><li>1.    JavaScript</li><li>2.    CSS</li><li>3.    HTML</li><li>4.    Images</li></ul></li><li>C.    Development Builds</li><li>D.    Production Builds</li></ul> |
|   | <b>XV.    Conclusion</b>   |