

Specialized - Angular 7

Code:	ANGL7
Length:	5 days
URL:	View Online

After understanding how HTML5 and JavaScript work together you soon realize that to create truly capable web applications, the code becomes complex and expensive to maintain. That's where Angular comes in. This framework allows us to drastically reduce the amount of JavaScript code written, making our web apps easier to understand at all levels. Angular lets developers think of their pages as sets of components and makes our web apps truly abstract.

- Angular is the framework of choice for organizations who are creating the best and newest web applications. This interactive course will get your developers ready to begin using Angular to keep you at the cutting edge of technology.

Skills Gained

Upon completion of this course, the student should be able to:

- Create fully-featured dynamic single-page web apps with Angular that allow user inputs and responds to events
- Explain how TypeScript allows you to write future JavaScript today
- Understand and use Angular components, one-way and two-way binding, property binding, expressions
- Employ Angular Component Router
- Consume RESTful Ajax services through promises and observables
- Exchange data with network servers using the Angular HTTP Client
- Create and work with Angular Forms and form validation
- Employ best practices for Angular application style and organization

Prerequisites

Understanding of HTML5 and a very strong grasp of object-oriented JavaScript.

Course Details

Schedule

- Day 1 - Intro to Angular, The Angular CLI, The Big Picture of Angular
- Day 2 - TypeScript Intro, Components Intro, Built-in Directives
- Day 3 - Routing Intro, Event Binding, Forms
- Day 4 - Composition with Components, Ajax and REST, Observables
- Day 5 - Services, Pipes (time permitting), Modules (Time permitting)

Outline

Course Overview

Intro to Angular

- What Angular does for you
- How it goes today
- The heart of Angular
- It's opinionated!
- The infrastructure

The Angular CLI

- Why the CLI?
- Installing
- Scaffolding an app
- Scaffolding components
- How to develop with the CLI
- Using the built-in node server
- Going to production

The Big Picture

- Components are the heart
- Properties vs attributes
- How it all fits together
- Modules
- Separation of logic from view
- The four kinds of bindings
- How Angular runs
- Why SPAs?

TypeScript

- JavaScript versions (es2015, 16, 17 to the present)
- Why use TypeScript?
- How TypeScript works behind the scenes
- TypeScript features needed for Angular
- Modules, arrow functions, and classes
- Public/private members
- Static typing with TypeScript
- Constructor shorthand
- Decorators

Components Introduction

- To create a component
- Interpolation - binding data from the class to the view
- Template references
- Expressions deep dive
- The 3 methods of styling a component
- Stylesheets
- styleUrls vs. styles

Built-in Directives

- The 5 different types of directives
- Attribute directives
- ngStyle and ngClass
- The structural directives
- *ngIf
- *ngSwitchCase
- *ngFor

Angular Routing

- How Angular handles SPAs
- The steps to routing
- Setting the root of your SPA
- The router-outlet
- How to design and code your routes
- Handline a direct URL
- Pushing the user to a route via JavaScript
- Creating well-designed links
- Reading route parameters
- Reading queryStrings

Event Binding

- All the events Angular supports
- How to wire up the events
- Handling the event object

Forms and two-way binding

- Two-way binding with ngModel
- Avoiding the common mistakes
- Validation with status tokens
- Error tokens
- Automatic CSS class settings

Composition with Components

- The case for components
- The steps to compose
- Passing data from host to inner
- The trick to passing data back up
- Emitting custom events
- Two-way binding between components

Ajax in Angular

- Why Angular is the perfect place to handle Ajax and RESTful calls
- HttpClient and HttpClientModule
- How to handle the callback with Promises and async/await
- What's in the response object
- How to unwrap it

Observables

- What is an observable?
- How they're different from Promises
- How to create and process an observable
- Observables are lazy - why
- Pipeable (lettable) operators
- Using them with HTTP and RESTful services

Services in Angular (Time Permitting)

- Dispelling the wrong beliefs about services - What they really are
- Best practices and patterns
- How to add services
- Best placement in your project
- Writing and using custom services

Angular Pipes (Time Permitting)

- Pipes are like Unix pipes
- All the built-in Angular pipes
- How to create custom pipes

Modules (Time permitting)

- Why modules?
- Shared modules
- Creating modules
- What goes in the decorators?

- Providing services
- Specifying Inline Styles (JavaScript)

Schedule (as of 3)

Date	Location
------	----------

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy

Get Your Free Copy Now

ExitCertified® Corporation and iMVP® are registered trademarks of ExitCertified ULC and ExitCertified Corporation and Tech Data Corporation, respectively
Copyright ©2021 Tech Data Corporation and ExitCertified ULC & ExitCertified Corporation.
All Rights Reserved.

Generated 9