Course Outline

Introduction to Angular 11

Course Summary

Description

Angular allows developers to easily build dynamic, responsive single-page web applications that dynamically rewrite portions of the current page rather than having to generate a new page in response to every request.

The Introduction to Angular training teaches attendees how to build applications using ES6, TypeScript, and modern front-end tools, including npm and Webpack.

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:

- 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
- Understand & Preview Ivy: The Next-Generation Compilation & Rendering Pipeline

Topics

- Introduction
- TypeScript and ECMAScript 6 (ES6) Fundamentals
- Angular Overview
- Components
- Angular Modules (NgModule)
- Project Set-Up (Using the Angular CLI)
- Data Binding
- Directives
- Pipes
- Advanced Components
- Services & Dependency Injection
- Dependency Injection
- Model-driven Forms (Reactive Forms)
- Communicating with the Server using the HttpClient
- Router
- Deploying an Angular Application to Production
- Ivy: Next-Generation Compilation & Rendering Pipeline
- Upgrading to the latest version of Angular from earlier versions
- Conclusion

Introduction to Angular 11

Course Summary (cont.)

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

Three Days

Course Outline

Introduction to Angular 11

Course Outline

I. *Introduction*

II. *TypeScript and ECMAScript 6 (ES6) Fundamentals*

A. TypeScript Installation, Configuration & Compilation
B. Type Annotations
C. Classes
D. Scoping using let, var, and const Keywords
E. Arrow Functions
F. ES Modules
G. Decorators
H. Template Literals
I. Spread Syntax and Rest Parameters
J. Destructuring

III. *Angular Overview*

A. Angular Compared to Benefits of Building using Angular
B. Understanding Angular Versions
C. Single-page Web Application Architectures vs. Traditional Server-side Web Application Architectures
D. Angular Style Guide
E. Angular Architecture
F. Other JavaScript Libraries and Frameworks (React, VueJS, etc…)
G. Your First Angular Application

IV. *Components*

A. Understanding Components
B. Component Properties & Methods
C. Templates: Inline, Multi-line, and External with Component-relative Paths

V. *Angular Modules (NgModule)*

A. Angular Modules vs. ES Modules
B. Organizing your code into Feature Modules

VI. *Project Set-Up (Using the Angular CLI)*

A. Angular CLI Features
B. Creating a New Project (with new CLI Prompts)
C. Generating Code
D. Customizing the Angular CLI

VII. *Data Binding*

A. Interpolation
B. Property binding
C. Event binding
D. Two-way data binding

VIII.*Directives*

A. Structural: ngFor, ngIf, ngSwitch
B. Attribute: ngClass, ngStyle

IX. *Pipes*

A. Built-in Pipes: Using, Passing Parameters, Chaining

X. *Advanced Components*

A. Component Communication using @Input, @Output
B. Component Architecture
C. Component Styles
D. Component Lifecycle Hooks
E. Evaluating UI Component Frameworks & Libraries

XI. *Services & Dependency Injection*

A. Using a service to access data
B. Using a service to encapsulate business logic
C. Understanding the scope of services

XII. *Dependency Injection*

A. Understanding Dependency Injection
B. Angular's Dependency Injection System
C. Registering
D. Injecting

XIII.*Model-driven Forms (Reactive Forms)*

A. Importing the ReactiveFormsModule
B. FormControl, FormGroup, and AbstractControl
C. Binding DOM Elements to FormGroups and FormControls
D. Validation Rules, Messages, and Styles
E. Refactoring Reactive Forms for Reuse

Introduction to Angular 11

Course Outline (cont.)

F. Custom Validators

XIV. *Communicating with the Server using the HttpClient* Service

A. Deciding between Promises or Observables (RxJS)
B. Making an HTTP GET Request
C. Sending data to the server using Http POST and PUT Requests
D. Issuing an HTTP DELETE Request
E. Intercepting Requests and Responses

XV. *Router*

A. Importing the RouterModule
B. Configuring Routes
C. Displaying Components using a RouterOutlet
D. Navigating declaratively with RouterLink
E. Navigating with code using the Router
F. Accessing parameters using ActivatedRoute

XVI. *Deploying an Angular Application to Production*

A. Building an application using the Angular CLI
B. Differential loading: creating a modern build (ES2015) and a legacy build (ES5)
C. Deploying to a web server

XVII. *Ivy: Next-Generation Compilation & Rendering Pipeline*

A. Understanding Ivy
B. Status: Is Ivy Ready? (opt-in preview)
C. Previewing Ivy in a new Project
D. Previewing Ivy in an existing Project

XVIII. *Upgrading to the latest version of Angular from earlier versions*

A. 2.x and above
B. Update Guide
C. Deprecation Guide
D. Looking for AngularJS to Angular upgrades? See Advanced and Comprehensive Angular courses.

XIX. *Conclusion*

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically