ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Introduction to 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 2 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 three day introduction 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:

- Understand the Angular architecture
- Use npm as a Build Tool
- Work with TypeScript and ES6/ES2015
- Develop Angular Components
- Use directives and work with data binding
- Work with Services and Dependency Injection
- Create and validate forms
- Use HTTP with Observables and Promises
- Create a single-page application using Routing
- Format data using Pipes
- Setup a project

Topics

- Introduction
- npm QuickStart
- TypeScript and ES6/ES2015 Introduction
- Components
- Data Binding
- Directives
- Service and Dependency Injection
- Advanced Components
- Forms
- Form Validation
- Data Architectures
- HTTP
- Routing
- Pipes
- Project Setup

Audience

This course is designed for experienced web developers.

Prerequisites

Before taking this course, students should have prior experience developing with JavaScript.

Duration

Three days

Introduction to Angular

Course Outline

Introduction to Angular

Course Outline (cont'd)

H. Application Wide Dependency Injection
I. @Injectable Classes
J. Multiple Service Instances
K. @Optional and @Host Decorators
L. Providers
   1. Syntax
   2. Alternative/Aliased
   3. Class, Value, and Factory

VIII. Advanced Components
A. Lifecycle Hooks
B. Composing Your User Interface
C. Component Communication
   1. @Input
   2. @Output and EventEmitters
D. Component Styles
   1. Metadata: styles and styleUrls
   2. View Encapsulation
   3. Style Scoping with Special Selectors

IX. Forms
A. Benefits of Angular Forms
B. New Forms API
C. Form Strategies
   1. Template-driven
   2. Model-driven
   3. Pros and Cons
D. Form Directives: Template-driven
   1. ngForm
   2. ngModel
   3. ngModelGroup
   4. ngSubmit
E. Getting Data from Form Controls
   1. Local Template Reference Variables
F. Binding to HTML Form Elements
   1. Select
   2. Checkbox

X. Form Validation
A. Validation Directives
B. Tracking Change State of Form Controls
C. CSS Classes
D. Validation Messages
E. Validation Styles

XI. Data Architectures
A. Model-View-Controller (MVC): Traditional Web Applications
B. Model-View-Whatever (MVW/MV*): Angular
C. New Architectures
   1. Observables and Reactive Programming
   2. Flux
D. Flexible Data Architecture in Angular

XII. HTTP
A. Setup
   1. Providing the HTTP Services
   2. Enable RxJS Operators
B. Http in Services using Promises
   1. Fetching Data
   2. Handling the Response
   3. Handling Errors
C. Http in Components using Promises
   1. Fetching Data
   2. Handling the Response
   3. Handling Errors
D. Observables and Reactive Programming
   1. The Reactive Extensions for JavaScript (RxJS)
   2. Big Ideas About Streams
E. Http in Services using Observables
F. Http in Components using Observables
G. Async Pipe
H. In-Memory Web API
I. Http Put
J. Http Delete
K. Cross-origin HTTP Requests

XIII. Routing
A. Component Router
B. Router Terminology
C. Router Setup
D. Location Strategies
   1. PathLocationStategy
   2. HashLocationStrategy
E. Router Directives
   1. routerLink
   2. routerLinkActive
   3. routerOutlet
F. Navigating
   1. Creating a Link
   2. Navigating with Code (Router)
   3. Route parameters
   4. Query parameters
   5. Retrieving Parameters
      a) ActivatedRoute
      b) Synchronous
      c) Asynchronous
G. Web Server Configuration

Introduction to Angular

Course Outline (cont'd)

XIV. **Pipes**
- A. What are Pipes?
- B. Using Pipes
  1. In Templates
  2. In Code
- C. Built-in Pipes
  1. Date, Number, Decimal, Currency, UpperCase and LowerCase
- D. Pipe Syntax
  1. Passing parameters to a pipe
- E. Chaining pipes
- F. Changes from Angular (deprecated pipes)

XV. **Project Setup**
- A. npm Dependencies
- B. TypeScript configuration
- C. SystemJS/Webpack configuration
- D. App Component
- E. Create an App Module
- F. Entry Point (main.ts) and Bootstrapping
- G. index.html
  1. scripts to include
- H. Build and run the app
- I. Other
  1. SASS
  2. Bootstrap, SemanticUI, Material Design
- J. Using a seed or boilerplate