



Angular: Comprehensive Angular 8 Programming

(WA2890)

Course ID #: 1121-008-08-W

Hours: 35

Course Content

Course Description:

The Comprehensive Angular 8 Programming Training course offered by Web Age Solutions covers all the essential topics found in the introductory course as well as additional topics typically encountered while developing real-world applications.

The Comprehensive Angular 8 course is designed to get students up and running with basic Angular 8 development and provide the knowledge needed for more challenging tasks.

The Angular 8 framework supports the creation of single-page browser applications as well as responsive web sites and hybrid mobile applications. This Angular 8 course covers all the basics including: Typescript, components, directives, services, pipes, form development, HttpClient and observables. In addition it covers advanced usage of HttpClient, observables and routing. Added to that are topics on consuming WebSockets data, Testing and Debugging of Angular applications.

At Course Completion:

After competing this course, student will be able to:

- Develop single page Angular applications using Typescript
- Set up a complete Angular development environment
- Create Components, Directives, Services, Pipes, Forms and Custom Validators
- Handle advanced network data retrieval tasks using Observables
- Consume data from REST web services using the Angular HTTP Client
- Handle push-data connections using the WebSockets protocol
- Work with Angular Pipes to format data
- Use advanced Angular Component Router features
- Test and debug Angular applications using built in tools
- Work with Angular CLI

Prerequisites:

Web development experience using HTML, CSS and JavaScript is required to get the most out of this Angular course. Knowledge of the browser DOM is also useful. Prior Angular experience, with AngularJS or the current version of Angular, is not required

Topics:

Chapter 1. Introducing Angular

- What is Angular?
- Central Features of the Angular Framework
- Why Angular?
- Building Blocks of an Angular Application
- Basic Architecture of an Angular Application
- Installing and Using Angular
- A Basic Angular Application
- Anatomy of a Basic Application
- The Main Component File
- The Application Module File



Angular: Comprehensive Angular 8 Programming (WA2890)

Course ID #: 1121-008-08-W

Hours: 35

- The index.html File
- The Bootstrap File
- Running the Application
- Building the Application

Chapter 2. Development Setup of Angular

- Angular Development Lifecycle
- Angular is Modular
- What is Node.js?
- Installing Node.js and NPM
- Node Package Manager (NPM)
- Package Descriptor File - package.json
- Semantic Version Numbering
- Package Version Numbering Syntax
- Updating Packages
- Uninstalling Packages
- Installing Angular Packages
- Angular CLI
- Creating a New Project
- Angular
- Development
- Dependencies
- TypeScript Typings
- Testing Tools
- Development Web Server
- Configuring the Web Server
- The Build System
- Configuring the Build

Chapter 3. Introduction to TypeScript and ES6

- Programming Languages for Use with Angular
- TypeScript Syntax
- Programming Editors
- The Type System – Defining Variables
- The Type System – Defining Arrays
- Type in Functions
- Type Inference
- The Type System – Classes & Objects
- Class Constructors
- Class Constructors – Alternate Form
- Interfaces

- Working with ES6 Modules
- Visibility Control
- var, let and const - defined
- var, let and const - usage
- Arrow Functions
- Arrow Function Compact Syntax
- Arrow Function and Caller Context
- Template Strings
- Template Strings – Variables and Expressions
- Template Strings – Multiline
- Generics - Class
- Generics - Methods
- Generics - Restricting Types
- Generics - Restricting Types: Example
- TypeScript Transpilation

Chapter 4. Components in Angular

- What is a Component?
- An Example Component
- The Component Class
- Adding a Component to Its Module
- Creating a Component Using Angular CLI
- Developing a Simple Login Component
- Component Template
- Login Component: Add HTML
- The HTML Component Template
- The templateUrl property
- Login Component: Add CSS Styling
- Login Component: Hook Up Input Fields and Button
- Login Component: Fields & Button in the Component Class
- Component Decorator Properties
- Component Hierarchy
- The Application Root Component
- Using a Regular Component
- The Build System
- Component Lifecycle Hooks
- Using a Lifecycle Hook:
- OnInit



Angular: Comprehensive Angular 8 Programming (WA2890)

Course ID #: 1121-008-08-W

Hours: 35

Chapter 5. Data and Event Binding

- Binding Syntax
- One-Way Output Binding
- Binding Displayed Output Values
- Setting Component Properties
- More About Setting Properties
- Setting DOM Element Properties
- Event Binding
- Binding Events Examples
- Firing Event from a Component
- @Output() Example - Child Component
- @Output() Example - Parent Component
- Two-Way Binding of Input Fields
- Input Binding Examples
- Two Way Binding in a Component
- Use Two Way Binding
- Breaking Down ngModel

Chapter 6. Attribute Directives

- What are Attribute Directives
- Apply Styles by Changing CSS Classes
- Changing Classes – Example
- Applying Styles Directly
- Applying Styles Directly - Example
- Controlling Element Visibility
- Setting Image Source Dynamically
- Setting Hyperlink Source Dynamically
- Writing a Custom Attribute Directive
- Using a Custom Attribute Directive
- Supplying Input to a Directive
- Handling Event from a Custom Directive

Chapter 7. Structural Directives

- Structural Directives
- Adding and Removing Elements Dynamically
- If-Else Syntax of ngIf
- Looping Using ngFor
- ngFor - Basic Example
- Creating Tables with ngFor
- ngFor Local Variables
- Manipulating the Collection

- Example - Deleting an Item
- Swapping Elements with ngSwitch
- ngSwitch - Basic Syntax

Chapter 8. Template Driven Forms

- Template Driven Forms
- Importing Forms Module
- A Basic Angular Form
- Binding Input Fields
- Accessing the NgForm Directive
- Binding the Form Submit Event
- The Submit Function
- Basic HTML5 Validation - "required" Attribute
- HTML5 vs. Angular Validation
- Angular Validators
- The NgModel Directive
- Controlling when validation is applied
- Displaying
- Form
- Validation State
- Displaying
- Field
- Validation State
- Displaying Validation State Using Classes
- Disabling Submit when Form is Invalid
- Submitting the Form
- Binding to Object Variables
- Binding to Object Variables - Code
- Additional Input Types
- Checkboxes
- Select (Drop Down) Fields
- Rendering Options for Select (Drop Down)
- Date fields
- Radio Buttons

Chapter 9. Reactive Forms

- Reactive Forms Overview
- The Building Blocks
- Import ReactiveFormsModule
- Construct a Form



Angular: Comprehensive Angular 8 Programming (WA2890)

Course ID #: 1121-008-08-W

Hours: 35

- Design the Template
- FormControl Constructor
- Getting Form Values
- Setting Form Values
- The Synchronous Nature
- Subscribing to Input Changes
- Validation
- Built-In Validators
- Showing Validation Error
- Custom Validator
- Using a Custom Validator
- Sub FormGroups - Component Class
- Sub FormGroups - HTML Template
- Why Use Sub FormGroups

Chapter 10. Angular Modules

- Why Angular Modules?
- But, We Already Had ES6 Module
- Angular Built-in Modules
- The Root Module
- Feature Modules
- Create Feature Module Using CLI
- The Module Class
- @NgModule Properties
- Using One Module From Another
- Importing BrowserModule or CommonModule
- Lazy-Loaded Modules
- How to Organize Modules?
- Third Party Modules

Chapter 11. Services and Dependency Injection

- What is a Service?
- Creating a Basic Service
- What is Dependency Injection?
- What Dependency Injection Looks Like
- Injectors
- Injector Hierarchy
- Register a Service with a Module Injector
- Registering a Service with the Root Injector
- Registering a Service with a Component's

Injector

- Where to Register a Service?
- Dependency Injection in Other Artifacts
- Providing an Alternate Implementation
- Dependency Injection and @Host
- Dependency Injection and @Optional

Chapter 12. HTTP Client

- The Angular HTTP Client
- Using The HTTP Client - Overview
- Importing HttpClientModule
- Simple Example
- Service Using HttpClient
- ES6 Import Statements
- Making a GET Request
- What does an Observable Object do?
- Using the Service in a Component
- The PeopleService Client Component
- Error Handling
- Making a POST Request
- Making a PUT Request
- Making a DELETE Request

Chapter 13. Pipes and Data Formatting

- What are Pipes?
- Built-In Pipes
- Using Pipes in HTML Template
- Chaining Pipes
- Using Pipes in Code
- Internationalized Pipes (i18n)
- Loading Locale Data
- Decimal Pipe
- Currency Pipe
- Custom Pipes
- Custom Pipe Example
- Using Custom Pipes
- Using a Pipe with ngFor
- A Filter Pipe
- A Sort Pipe
- Pipe Category: Pure and Impure
- Pure Pipe Example
- Impure Pipe Example



Angular: Comprehensive Angular 8 Programming (WA2890)

Course ID #: 1121-008-08-W

Hours: 35

Chapter 14. Introduction to Single Page Applications

- What is a Single Page Application (SPA)
- Traditional Web Application
- SPA Workflow
- Single Page Application Advantages
- HTML5 History API
- SPA Challenges
- Implementing SPA's Using Angular

Chapter 15. The Angular Component Router

- The Component Router
- View Navigation
- The Angular Router API
- Creating a Router Enabled Application
- Hosting the Routed Component
- Navigation Using Links and Buttons
- Programmatic Navigation
- Passing Route Parameters
- Navigating with Route Parameters
- Obtaining the Route Parameter Values
- Retrieving the Route Parameter Synchronously
- Retrieving a Route Parameter Asynchronously
- Query Parameters
- Supplying Query Parameters
- Retrieving Query Parameters Asynchronously
- Problems with Manual URL entry and Bookmarking

Chapter 16. Advanced HTTP Client

- Request Options
- Returning an HttpResponse Object
- Setting Request Headers
- Creating New Observables
- Creating a Simple Observable
- The Observable.create() Method
- Observable Operators
- More About map
- Piping Operators

- The flatMap() Operator
- The tap() Operator
- The zip() Operator
- Caching HTTP Response
- Making Sequential HTTP Calls
- Making Parallel Calls
- Customizing Error Object with catchError()
- Error in Pipeline
- Error Recovery

Chapter 17. Consuming WebSockets Data in Angular

- Web Sockets Overview
- Web Sockets Use Cases
- Web Socket URLs
- Web Sockets Servers
- Web Socket Client
- The socket.io-client library
- Using socket.io-client in JavaScript
- Setting up socket.io-client in Angular Projects
- Using socket.io-client in an Angular service
- Angular websocket.service Notes:
- The Angular Web Socket Client Sample App
- Angular websocket.component.ts
- The Full websocket.component.ts code
- Implementation Modifications

Chapter 18. Advanced Routing

- Routing Overview
- Routing Enabled Project
- Routing Enabled Feature Module
- Using the Feature Module
- Lazy Loading the Feature Module
- Creating Links for the Feature Module Components
- More About Lazy Loading
- routerLinkActive binding
- Default Route



Angular: Comprehensive Angular 8 Programming (WA2890)

Course ID #: 1121-008-08-W

Hours: 35

- Wildcard Route Path
- redirectTo
- Child Routes
- Defining Child Routes
- <router-outlet> for Child Routes
- Links for Child Routes
- Navigation Guards
- Creating Guard Implementations
- Using Guards in a Route
- Route Animations

Chapter 19. Introduction to Testing Angular Applications

- Unit Testing Angular Artifacts
- Testing Tools
- Testing Setup
- Typical Testing Steps
- Test Results
- Jasmine Test Suites
- Jasmine Specs (Unit Tests)
- Expectations (Assertions)
- Matchers
- Examples of Using Matchers
- Using the not Property
- Setup and Teardown in Unit Test Suites
- Example of beforeEach and afterEach Functions
- Angular Test Module
- Example Angular Test Module
- Testing a Service
- Injecting a Service Instance
- Test a Synchronous Method
- Test an Asynchronous Method
- Using Mock HTTP Client
- Supplying Canned Response
- Testing a Component
- Component Test Module
- Creating a Component Instance
- The ComponentFixture Class
- Basic Component Tests
- The DebugElement Class
- Simulating User Interaction

Chapter 20. Debugging

- Overview of Angular Debugging
- Viewing TypeScript Code in Debugger
- Using the debugger Keyword
- Inspecting Components
- Saving ng.probe Component References
- Modifying Values using Component References
- Debug Logging
- What is Augury?
- Installing Augury
- Opening Augury
- Augury - Component Tree
- Augury - Router Tree.
- Augury - NgModules Tab
- Common Exceptions
- Common Exceptions: 'No such file: package.json'
- Common Exceptions: 'Cant bind to ngModel'
- Common Exceptions: 'router-outlet not a known element'
- Common Exceptions: 'No provider for Router!'

Chapter 21. Introduction to Angular Material

- Material Design (MD)
- Angular Material (AM)
- Adding Angular Material to a Project
- Angular Default Component
- Angular Material Schematics
- Angular Material Modules
- Adding Angular Material Modules to a Project
- material.module.ts File
- Adding material.module.ts to a Project
- Angular Material Themes
- Using Themes
- Configuring Animations
- Enabling Gesture Support
- Angular Material Icons
- Angular Material Components
- Navigation



Angular: Comprehensive Angular 8 Programming (WA2890)

Course ID #: 1121-008-08-W

Hours: 35

- Layout
- Layout Examples
- Form Components
- Example Controls