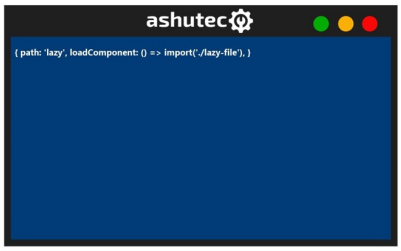

LUMIFY WORK Angular 15 Programming User Guide

APPLICATION AND WEB DEVELOPMENT
Angular 15 Programming
LENGTH **5 days**
VERSION **15**

Contents

1 WHY STUDY THIS COURSE
2 WHAT YOU'LL LEARN
3 COURSE SUBJECTS
4 WHO IS THE COURSE FOR?
5 PREREQUISITES
6 Documents / Resources
6.1 References
7 Related Posts

WHY STUDY THIS COURSE

This intensive and comprehensive Angular 15 training course provides attendees with skills they can use immediately in their work. You'll learn the fundamentals of Angular 15 development such as single-page browser applicat ions, responsive websites, and hybrid mobile applicat ions.
This course is a combinat ion of theoret ical learning and hands-on labs that includes an introduct ion to Angular, followed by TypeScript, components, direct ives, services, HTTPClient, test ing, and debugging.
Note: We can also deliver training on other versions of Angular. Please contact us to make an enquiry or register your interest.

ANGULAR AT LUMIFY WORK

WHAT YOU'LL LEARN

On successful complet ion of this course, you will have the knowledge to:

- Develop single page Angular applicat ions using Typescript
- Set up a complete Angular development environment
- Create Components, Direct ives, 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 connect ions using the WebSocket protocol
- Work with Angular Pipes to format data
- Use advanced Angular Component Router features
- Test and debug Angular applicat ions using built in tools
- Work with Angular CLI

My instructor was great being able to put scenarios into real world instances that related to my specific situation.
I was made to feel welcome from the moment I arrived and the ability to sit as a group outside the classroom to discuss our situations and our goals was extremely valuable.
I learnt a lot and felt it was important that my goals by attending this course were met.
Great job Lumify Work team.

AMANDA NICOL
IT SUPPORT SERVICES MANAGER – HEALTH WORLD LIMITED

COURSE SUBJECTS

1. **Introducing Angular**
   - What is Angular?
   - Central Features of the Angular Framework
   - Appropriate Use Cases
   - Building Blocks of an Angular Application
   - Basic Architecture of an Angular Application
   - Installing and Using Angular
   - Anatomy of an Angular Applicat ion
   - Running the Applicat ion
   - Building and Deploying the Application
   - Angular for Native Mobile Apps
2. **Introduction to TypeScript**
   - Programming Languages for Use with Angular
   - TypeScript Syntax

- Programming Editors
- The Type System – Def ining Variables
- The Type System – Def ining Arrays
- Basic Primit ive Types
- Type in Funct ions
- Type Inference
- Defining Classes
- Class Methods
- Visibility Control
- Class Constructors
- Class Constructors – Alternate Form
- Uninitialized Fields
- Interfaces
- Working with ES6 Modules
- var vs let
- Arrow Functions
- Arrow Function Compact Syntax
- Template Strings
- Generics in Class
- Generics in Funct ion

3. **Components**
   - What is a Component?
   - An Example Component
   - Creat ing a Component Using Angular CLI
   - The Component Class
   - The @Component Decorator
   - Registering a Component to Its Module
   - Component Template
   - Example: HelloComponent Template
   - Example: The HelloComponent Class
   - Using a Component
   - Run the Application
   - Component Hierarchy
   - The Applicat ion Root Component
   - The Bootstrap File
   - Component Lifecycle Hooks
   - Example Lifecycle Hooks
   - CSS Styles
4. **Component Templates**
   - Templates
   - Template Location
   - The Mustache {{ }} Syntax
   - Setting DOM Element Properties
   - Setting Element Body Text

- Event Binding
- Expression Event Handler
- Prevent Default Handling
- Attribute Directives
- Apply Styles by Changing CSS Classes
- Example: ngClass
- Applying Styles Directly
- Structural Directives
- Conditionally Execute Template
- Example: ngIf
- Looping Using ngFor
- ngFor Local Variables
- Manipulating the Collection
- Example – Deleting an Item
- Item Tracking with ngFor
- Swapping Elements with ngSwitch
- Grouping Elements
- Template Reference Variable

5. **Inter Component Communication**
   - Communication Basics
   - The Data Flow Architecture
   - Preparing the Child to Receive Data
   - Send Data from Parent
   - More About Setting Properties
   - Firing Event from a Component
   - @Output() Example – Child Component
   - @Output() Example – Parent Component
   - Full Two Way Binding
   - Setting up Two Way Data Binding in Parent
6. **Templat e Driven Forms**
   - Template Driven Forms
   - Importing Forms Module
   - Basic Approach
   - Sett ing Up a Form
   - Getting User Input
   - Omitting ngForm Attribute
   - Initialize the Form
   - Two Way Data Binding
   - Form Validation
   - Angular Validators
   - Displaying Validat ion State Using Classes
   - Additional Input Types
   - Checkboxes

- Select (Drop Down) Fields
- Rendering Options for Select (Drop Down)
- Date fields
- Radio Buttons

7. **Reactive Forms**
   - Reactive Forms Overview
   - The Building Blocks
   - Import Reactive FormsModule
   - Construct a Form
   - Design the Template
   - Getting Input Values
   - Initializing the Input Fields
   - Setting Form Values
   - Subscribing to Input Changes
   - Validation
   - Built-In Validators
   - Showing Validation Error
   - Custom Validator
   - Using a Custom Validator
   - Supplying Configuration to Custom Validator
   - FormArray – Dynamically Add Inputs
   - FormArray – The Component Class
   - FormArray – The Template
   - FormArray – Values
   - Sub FormGroups – Component Class
   - Sub FormGroups – HTML Template
   - Why Use Sub FormGroups
8. **Services and Dependency Injection**
   - What is a Service?
   - Creat ing a Basic Service
   - The Service Class
   - What is Dependency Injection?
   - Injecting a Service Instance
   - Injectors
   - Injector Hierarchy
   - Registering a Service with the Root Injector
   - Registering a Service with a Component ‘s Injector
   - Register a Service with a Feature Module Injector
   - Where to Register a Service?
   - Dependency Injection in Other Artifacts
   - Providing an Alternate Implementation
   - Dependency Injection and @Host
   - Dependency Injection and @Optional
9. **HTTP Client**

- The Angular HTTP Client
- Using The HTTP Client – Overview
- Import ing HttpClientModule
- Service Using HttpClient
- Making a GET Request
- What does an Observable Object do?
- Using the Service in a Component
- The PeopleService Client Component
- Error Handling
- Customizing the Error Object
- Making a POST Request
- Making a PUT Request
- Making a DELETE Request

10. **Pipes and Data Formatting**
    - What are Pipes?
    - Built-In Pipes
    - Using Pipes in HTML Template
    - Chaining Pipes
    - Internat ionalized Pipes (i18n)
    - Loading Locale Data
    - The date Pipe
    - The number Pipe
    - Currency Pipe
    - Create a Custom Pipe
    - Custom Pipe Example
    - Using Custom Pipes
    - Using a Pipe with ngFor
    - A Filter Pipe
    - Pipe Category: Pure and Impure
    - Pure Pipe Example
    - Impure Pipe Example
11. **Introduction to Single Page Applications**
    - What is a Single Page Applicat ion (SPA)
    - Traditional Web Application
    - SPA Workflow
    - Single Page Application Advantages
    - HTML5 History API
    - SPA Challenges
    - Implementing SPA's Using Angular
12. **The Angular Component Router**
    - The Component Router
    - View Navigation
    - The Angular Router API

- Creating a Router Enabled Application
- Hosting the Routed Components
- 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

13. **Advanced HTTP Client**
    - Request Options
    - Returning an HttpResponse Object
    - Setting Request Headers
    - Creating New Observables
    - Creating a Simple Observable
    - The Observable Constructor Method
    - Observable Operators
    - The map and filter Operators
    - The flatMap() Operator
    - The tap() Operator
    - The zip() Combinator
    - Caching HTTP Response
    - Making Sequential HTTP Calls
    - Making Parallel Calls
    - Customizing Error Object with catchError()
    - Error in Pipeline
    - Error Recovery
14. **Angular Modules**
    - Why Angular Modules?
    - Anatomy of a Module Class
    - @NgModule Properties
    - Feature Modules
    - Example Module Structure
    - Create a Domain Module
    - Create a Routed/Routing Module Pair
    - Create a Service Module
    - Creating Common Modules
    - Using One Module From Another
15. **Advanced Routing**
    - Routing Enabled Feature Module

- Using the Feature Module
- Lazy Loading the Feature Module
- Creating Links for the Feature Module Components
- More About Lazy Loading
- Preloading Modules
- routerLinkActive binding
- Default Route
- Wildcard Route Path
- redirectTo
- Child Routes
- Defining Child Routes for Child Routes
- Links for Child Routes
- Navigation Guards
- Creating Guard Implementations
- Using Guards in a Route

16. **Unit Testing Angular Applications**
    - Unit Testing Angular Artifacts
    - Testing Tools
    - Typical Testing Steps
    - Test Results
    - Jasmine Test Suites
    - Jasmine Specs (Unit Tests)
    - Expectations (Assert ions)
    - 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
    - Test ing a Component
    - Component Test Module
    - Creating a Component Instance
    - The ComponentFixture Class
    - Basic Component Tests
    - The DebugElement Class
    - Simulating User Interaction

17. **Debugging**
    - Overview of Angular Debugging
    - Viewing TypeScript Code in Debugger
    - Using the debugger Keyword
    - Debug Logging
    - What is Angular DevTools?
    - Using Angular DevTools
    - Angular DevTools – Component Structure
    - Angular DevTools – Change Detect ion Execut ion
    - Catching Syntax Errors

WHO IS THE COURSE FOR?

This course is aimed at anyone who needs to learn the fundamentals of Angular 15 development and apply it straight away to creating web applications.
We can also deliver and customise this training course for larger groups – saving your organisation time, money and resources. For more information, please contact us via email on ph.training@lumifywork.com

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
- No prior experience of Angular or AngularJS is required

The supply of this cours e by Lumify Work is governed by the booking terms and conditions .
Please read the terms and conditions carefully before enrolling in this cours e, as enrolment in the course is conditional on acceptance of thes e terms and conditions .

https://www.lumifywork.com/en-ph/courses/angular-15-programming/

ph.training@lumifywork.com
lumifywork.com
facebook.com/LumifyWorkPh
linkedin.com/company/lumify-work-ph
twitter.com/LumifyWorkPH
youtube.com/@lumifywork

Documents / Resources

	LUMIFY WORK Angular 15 Programming [pdf] User Guide Angular 15 Programming, Programming

References

- Lumify Work | Lumify Work AU
- ywork.com
- Lumify Work | Lumify Work AU
- Lumify Work | Lumify Work AU
- Angular 15 Programming | Lumify Work PH
- User Manual

Manuals+, Privacy Policy

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.