

# Angular Course Structure – Hemant (9820866165)

## a) Introduction

- Introduction
- What is Angular
- Architecture of Angular Apps
- Set Up Development Environment
- Your First Angular App
- Structure of Angular Projects
- Webpack
- Angular Version History
- Angular Basics
- 5 questions
- Course Structure
- Making a Promise
- Asking Questions

## b) TypeScript Fundamentals

- Introduction
- What is TypeScript?
- Your First TypeScript Program
- Declaring Variables
- Types
- Type Assertions
- Arrow Functions
- Interfaces
- Classes
- Objects
- Constructors
- Access Modifiers
- Access Modifiers in Constructor
- Parameters
- Properties
- Modules
- Exercise
- Solution
- A Problem with the Current Implementation

## c) Angular Fundamentals

- Introduction
- Building Blocks of Angular Apps
- Components
- Generating Components Using Angular CLI
- Templates
- Directives
- Services
- Dependency Injection
- Generating Services Using Angular CLI
- List of Authors

## d) Displaying Data and Handling Events

- Introduction
- Property Binding
- Attribute Binding
- Adding Bootstrap
- Class Binding
- Style Binding
- Event Binding
- Event Filtering
- Template Variables
- Two-way Binding
- Pipes
- Custom Pipes
- Favorite Component
- 1 question
- Title Casing

## e) Building Re-usable Components

- Introduction
- Component API
- Input Properties
- Aliasing Input Properties
- Output Properties
- Passing Event Data
- Aliasing Output Properties
- Templates
- Styles
- View Encapsulation
- ngContent
- ngContainer
- Like Component

## f) Directives

- Introduction
- ngIf
- Hidden Property
- ngSwitchCase
- ngFor
- ngFor and Change Detection
- ngFor and Trackby
- The Leading Asterisk
- ngClass
- ngStyle
- Safe Traversal Operator
- Creating Custom Directives
- Exercise: ZippyComponent

## g) Template-driven Forms

- Introduction
- Building a Bootstrap Form
- Types of Forms
- ngModel
- Adding Validation
- Specific Validation Errors
- Styling Invalid Input Fields
- Cleaner Templates
- ngForm
- ngModelGroup
- Control Classes and Directives
- Disabling the Submit Button
- Working with Check Boxes
- Working with Drop-down Lists
- Working with Radio Buttons
- Course Form

## h) Reactive Forms

- Introduction
- Building a Bootstrap Form
- Creating Controls Programmatically
- Adding Validation
- Specific Validation Errors
- Implementing Custom Validation
- Asynchronous Operations
- Asynchronous Validators
- Showing a Loader Image
- Validating the Form Input Upon Submit
- Nested FormGroups
- FormArray
- FormBuilder
- Quick Recap
- Change Password Form

## i) Consuming HTTP Services

- Introduction
- JSONPlaceholder
- Getting Data
- Creating Data
- Updating Data
- Deleting Data
- OnInit Interface
- Separation of Concerns
- Extracting a Service
- Handling Errors
- Handling Unexpected Errors
- Handling Expected Errors
- Throwing Application-specific Errors
- Handling Bad Request Errors
- Importing Observable Operators and Factory Methods
- Global Error Handling
- Extracting a Reusable Error Handling Method
- Extracting a Reusable Data Service
- The Map Operator
- Optimistic vs Pessimistic Updates
- Observables vs Promises
- GitHub Followers Page

## j) Routing and Navigation

- Introduction
- Routing in a Nutshell
- Configuring Routes
- RouterOutlet
- RouterLink
- RouterLinkActive
- Getting the Route Parameters
- Why Route Parameters Are Observables
- Routes with Multiple Parameters
- Query Parameters
- Subscribing to Multiple Observables
- The SwitchMap Operator
- Programmatic Navigation
- Blog Archives

## k) Authentication and Authorization

- Introduction
- Application Overview
- Architecture
- JSON Web Tokens
- Starter Code
- Implementing Login
- Implementing Logout
- Showing or Hiding Elements
- Showing or Hiding Elements based on the User's Role
- Getting the Current User
- CanActivate Interface
- Redirecting Users After Logging In
- Protecting Routes Based on the User's Role
- Accessing Protected API Resources
- Quick Recap

## l) Deployment

- Introduction
- Preparing for Deployment
- JIT vs AOT Compilation
- Angular Compiler in Action
- Building Applications with Angular CLI
- Environments
- Adding Custom Environments
- Linting with Angular CLI
- Linting in VSCode
- Other Deployment Options
- Deploying to GitHub Pages
- Deploying to Firebase
- Heroku
- Deploying to Heroku
- Engines
- Exercise

## m) Building Real-time Server-less Apps with Firebase

- Introduction
- What is Firebase?
- Your First Firebase Project
- Working with Firebase Databases
- Installing Firebase
- Reading Lists
- A Real-time Database
- Observables and Memory Leaks
- Unsubscribing from Subscriptions
- Async Pipe
- Reading an Object
- As Keyword
- Adding an Object
- Updating an Object
- Removing an Object
- Additional Resources

## n) Animations

- Introduction
- Examples of Animations
- Different Ways to Create Animations
- Angular Animations
- Importing the Animations Module and Polyfills
- Implementing a Fade-in Animation
- Implementing a Fade-out Animation
- States
- Transitions
- Animatable Properties
- Creating Reusable Triggers
- Build a re-usable slide animation
- 1 question
- Easings
- Keyframes
- Creating Reusable Animations with animation()
- Parameterizing Reusable Animations
- Animation Callbacks
- Querying Child Elements with query()
- Animate Child Elements with animateChild()
- Running Parallel Animations with group()
- Staggering Animations with stagger()
- Working with Custom States
- Multi-step Animations
- Separation of Concerns

## o) Angular Material 2

- Introduction
- What is Angular Material
- Installing Angular Material
- Check Boxes
- Radio Buttons
- Selects
- Inputs
- Text Areas
- Date Pickers
- Icons
- Buttons
- Chips
- Progress Spinners
- Tooltips
- Tabs
- Dialogs
- Passing Data to Dialogs
- Other Components
- Creating a Reusable Module
- Themes
- SASS
- Creating a Custom Theme
- Using Angular Material's Typography
- Customizing Typography
- An Important Note

## p) Redux

- Introduction
- What is Redux
- Building Blocks of Redux
- Pure Functions
- Installing Redux
- Working with Actions
- The Select Decorator
- Avoiding State Mutation
- Using Immutable.js
- Exercise
- Redux DevTools
- Calling Backend APIs
- Refactoring Fat Case Statements
- Dealing with Complex Domains

## q) Unit Testing

- Introduction
- What is Automated Testing?
- Types of Tests
- Unit Testing Fundamentals
- Working with Strings and Arrays
- Set Up and Tear Down
- Working with Forms
- Working with Event Emitters
- Working with Spies
- Interaction Testing
- Working with Confirmation Boxes
- Limitations of Unit Tests
- Code Coverage
- Exercises

## r) Integration Testing

- Introduction
- The Setup
- Generating the Setup Code
- Testing Property Bindings
- Testing Event Bindings
- Providing Dependencies
- Getting the Dependencies
- Providing Stubs
- Testing the Navigation
- Dealing with Route Params
- Testing RouterOutlet Components
- Shallow Component Tests
- Testing Attribute Directives
- Dealing with Asynchronous Operations

## s) Project:

- Getting Started
- Authentication and Authorization
- Product Management Module
- Product Catalog Module
- Shopping Cart Module
- Check Out Module
- Modularization & Final

Improvements