

# The Road to Angular

**Overview:** Angular is a JavaScript Framework that makes it easy to build applications with the web and allows you to create a reactive Single-Page-Application (SPAs). It gives you full control over markup, supports **TDD**-friendly development and uses the latest web standards. If you want to get employed as a web developer at a company that utilizes Frontend technologies, you need to master Angular.

**Course Objectives:** This course will allow learners to Develop modern, complex, responsive and scalable web applications with Angular. Fully understand the architecture behind an Angular application and how to use it. Use their gained, deep understanding of the Angular fundamentals to quickly establish themselves as frontend developers. Create single-page applications with one of the most modern JavaScript frameworks out there.

**Target Audience:** These Syllabus is designed for beginners and professionals who want to learn Angular step by step with hand on practical.

**Pre-requisite:** NO Angular 1 or Angular 2 knowledge is required! Basic HTML and CSS knowledge helps but isn't a must-have. Prior JavaScript and TypeScript knowledge also helps but isn't necessary to benefit from this course.

**Course Outline:** Don't lose any time, gain an edge and start developing now!

## **Class 1: Basic JavaScript**

- What is JavaScript?
- Difference between Java and JavaScript.
- Introduction - JavaScript built in method, data types and variables.
- Conditionals - if, else if, else, switch, and ternary syntax to control the flow of a program in JavaScript.
- Function - JavaScript functions, passing data to functions, the return keyword, and concise body syntax.
- Arrays - you will learn about arrays, a data structure in JavaScript used to store lists of data.
- Loops - for and while loops to execute blocks of code multiple times.
- Iterators - how to use iterator methods to simplify the process of looping over arrays.
- Objects - JavaScript ES6 object syntax to model real-world items.
- Classes - how to create classes and use inheritance to minimize redundancy in your code.
- Practicing JavaScript

## **Class 2: Basic Typescript**

- What is Typescript
- JavaScript vs. Typescript
- Static type of Typescript

- A short example of Typescript code.
- Prepare environment of Typescript
- Class and Object and function
- Inheritance
- Practicing Typescript

### **Class 3: Angular - Introduction**

- Course details
- What is angular?
- Angular Versioning from AngularJs to Angular 7
- Project setup and First angular app
- Project structure
- Editing the first app
- A Basic Project Setup using Bootstrap for Styling
- Practicing Angular apps

### **Class 4: Components and Databinding**

- Understand and create a component using CLI
- Understanding the Role of AppModule and Component Declaration
- Working with Component Styles
- Fully Understanding the Component Selector
- Parent and Child component
- Practicing Component

### **Class 5: Components and Databinding**

- What is Databinding?
- String Interpolation
- Property Binding
- Property Binding vs String Interpolation
- Event Binding
- Bindable Properties and Events
- Passing and Using Data with Event Binding
- Important: FormsModule is required for Two-Way-Binding!
- Two-Way-Databinding
- Combining all Forms of Databinding
- Practicing Databinding

### **Class 6: Directives**

- Understanding Directives
- Using ngIf to Output Data Conditionally

- Enhancing ngIf with an Else Condition
- Styling Elements Dynamically with ngStyle
- Applying CSS Classes Dynamically with ngClass
- Outputting Lists with ngFor
- Practicing Directives

### **Class 7: Services and Dependency Injection**

- Why would we need Services?
- Creating a simple Service
- Injecting this Service into Components
- Creating a Data Service
- Injecting Services into Services
- Using Services for Cross-Component Communication
- Practice Service

### **Class 8: Routing**

- Why do we need a Router?
- Understanding the Example Project
- Setting up and Loading Routes
- Navigating with Router Links
- Understanding Navigation Paths
- Navigating Programmatically
- Passing Parameters to Routes
- Fetching Route Parameters
- Fetching Route Parameters Reactively
- An Important Note about Route Observables
- Setting up Child (Nested) Routes Using Query Parameters - Practice
- An Introduction to Guards
- Protecting Routes with canActivate
- Controlling Navigation with canDeactivate
- Understanding Location Strategies and Session Storage

### **Class 9: Observables and Promise**

- Angular 6 & RxJS 6
- Building & Using a First Simple Observable
- Understanding Observable Operators
- Promise

- Promise Vs. Observables
- Practicing Observables and Promise

### **Class 10: Angular Form - Template-Driven (TD)**

- Template-Driven (TD) vs. Reactive Approach
- An Example Form
- TD: Creating the Form and Registering the Controls
- TD: Submitting and Using the Form
- TD: Understanding Form State
- TD: Adding Validation to check User Input
- Built-in Validators & Using HTML5 Validation
- TD: Using the Form State
- TD: Outputting Validation Error Messages
- TD: Set Default Values with ngModel Property Binding
- TD: Using ngModel with Two-Way-Binding
- TD: Grouping Form Controls
- TD: Setting and Patching Form Values
- TD: Using Form Data
- TD: Resetting Forms
- Practicing Template-Driven Forms

### **Class 11: Angular Form - Reactive Approach**

- Introduction to the Reactive Approach
- Reactive: Setup
- Reactive: Creating a Form in Code
- Reactive: Submitting the Form
- Reactive: Adding Validation
- Reactive: Getting Access to Controls
- Reactive: Grouping Controls
- Reactive: Arrays of Form Controls (FormArray)
- Reactive: Creating Custom Validators
- Reactive: Using Error Codes
- Reactive: Reacting to Status or Value Changes
- Reactive: Setting and Patching Values
- Practicing Reactive Forms

### **Class 12: Angular Pipes**

- Using Pipes

- Parametrizing Pipes
- Creating a Custom Pipe
- Parametrizing a Custom Pipe
- Example: Creating a Filter Pipe
- Practicing Pipes

### **Class 13: Http**

- Introduction & How Http Requests Work in SPAs
- Example App & Backend Setup
- Sending Requests (Example: POST Request)
- Adjusting Request Headers
- Sending GET Requests
- Sending a PUT Request
- Transform Responses Easily with Observable Operators (map)
- Practicing Http

### **Class 14: Authentication and Route protection in Angular app**

- How Authentication Works in Single-Page-Applications
- More about JWT
- Creating a Signup Page and Route
- Use getIdToken() instead of getToken()
- Signing Users Up
- Signin Users In
- Requiring a Token (on the Backend)
- Sending the Token
- Checking and Using Authentication Status
- Adding a Logout Button
- Route Protection and Redirection Example
- Practicing Authentication

### **Class 15: Using Angular modules and Optimizing apps**

- Understanding the App Module
- Creating a Recipe Feature Module
- Loading Components via Selectors vs Routing
- Understanding Lazy Loading
- Protecting Lazy Loaded Routes with canLoad
- How Modules and Services Work Together

- Understanding the Core Module
- Creating a Basic Core Module
- Restructuring Services to use the Child Injector
- Preloading Lazy Loaded Routes

#### **Class 16: Deployment in Angular app**

- Deployment Preparations and Important Steps
- Example: deploy to IIS
- Practicing Deployment

#### **Class 17: Unit testing and Animation**

- Why Unit Tests?
- Analyzing the Testing Setup (as created by the CLI)
- Running Tests (with the CLI)
- Adding a Component and some fitting Tests
- Testing Dependencies: Components and Services

#### **Class 18, 19 and 20 - Project work and Written test**

##### **Specifically, you will learn:**

- Which architecture Angular uses
- How to use TypeScript to write Angular applications
- All about directives and components, including the creation of custom directives/components
- How databinding works
- All about routing and handling navigation
- What Pipes are and how to use them
- How to access the Web (e.g. RESTful servers)
- What dependency injection is and how to use it
- How to use Modules in Angular
- How to optimize your (bigger) Angular Application
- We will build a major project in this course
- And much more!