Angular 8 Developer Training Course

Master ES6,TypeScript Angular 8

Duration 5 days

Prerequisite:

- A minimum of 1 Year Experience in Programming
- Must have worked in JavaScript
- Must know Html,css
- Must have worked on Web Applications

Target Audience:

Beginner to Expert level

Software Requirement and Hardware Requirement

Hardware Requirement:

- 8 GB RAM Minimum
- Windows 7 or later

Internet Connection

- Admin rights for installing node modules

Note : if you face problem during npm install phase, then you have to set admin rights and also proxy must be enabled.

Proxy setup:

- Open an command prompt or terminal session and run the following commands to configure npm to work with your web proxy. The commands use proxy.company.com as the address and 8080 as the port.

```
npm config set proxy http://proxy.company.com:8080
```

IDE

1. Visual Studio Code

Runtime

- Node 1.8.x

NPM Modules

- Can be installed during Session Only.

Web Browsers - Any one Browser - Recommended Chrome

- Chrome

Tools -

- Chrome Plugin -POST Man

Day 01

ES 6 & TypeScript

- Project Setup
- What is Type Script
- Why TypeScript
- ES 6 Standards Overview
- Let, const – Block Scope
- Template String literals
- Variables Introduction
- Type systems
- Number,string,Boolean,void
- Object, Array
- Classes, instance variables, methods
- Arrow functions
- Encapsulation && modifiers private,public,protected
- Interfaces

- Generics
- ES6 Modules
- Decorators

Angular 8

Getting Started

- What is Angular 8
- Angular and Single Page Web Application Development
- Why Angular
- Project IVY
- Angular CLI, Angular Universal

Project Setup

- Understanding node , npm
- IDE, Visual Studio Code
- Installing Angular Plugins in IDE
- Chrome Browser latest

Angular CLI

- npm install -g @angular/cli
- Create a new project
- ng new my-app
- Serve the application
- Test "First Program"

Angular Architecture

- Architecture Overview
- Modules
- Components
- Meta Data
- Data binding
- Directives
- Services
- Dependency Injection

Writing first app

- Components
- Imports
- Template
- Angular Modules
- Bootstrapping
- Component Tree

Day 02

Template and Data binding

- Displaying Data
- Template Syntax
- Component Interaction
- Types of binding
- Interpolation
- Template expressions
- Two way data binding
- Property Binding
- Event Binding
- class Binding
- Style Binding

Modules ,DI

- Angular Modularity
- Dependency Injection in Angular
- The root App Module
- NG Module
- Shared Modules
- Dependency Injection
- Built in Modules Overview
- FormsModule ,HttpClient Modules, Animations

Boot Strapping in main.ts

- Compile just-in-time (JIT)

- AOT- Compile ahead-of-time
- Angular Modularity
- The root App Module
- Boot Strap file
- Compile Just in Time

Services and Reactive Programming with RxJS

- What is Service
- Service is Just Plain Type Script class
- @Injectable
- Types of Services
- Syn Services
- Async Services
- Custom Services
- Built in Services
- Streams & Reactive Programming
- Observables & RxJS
- RxJS & Angular

Day 03

Pipes

- Pipes and Data filtering
- Built-in Pipes
- Async Pipe
- Custom Pipes
- Chaining Pipes

Forms

- Overview
- Model Driven Forms
- Model Driven Form Validation
- Submitting & Resetting
- Reactive Model Form
- Template Driven Forms

Dependency Injection & Providers

- Overview

- Injectors
- Provider
- Tokens
- Configuring Dependency Injection in Angular

HTTP Client

- Overview
- HTTPClient Modules
- Core HTTPClient API
- Get
- Post
- Delete
- HTTP Example with Promises
- HTTP Example with Observables
- Observer able Operators
- Subscribe
- Tab
- Catch
- Map
- filter

Day 04

State Management (ngRx)(Flux)

- State Management Strategies
- Unidirectional data flow
- Redux as a state management solution
- Store Vs Services
- Application Store
- Reducers
- Actions
- Dispatching Actions
- Action Creators
- Using ngRx
- Handling Errors

Routing

- Overview

- Route Configuration
- Navigation
- Parameterized Routes
- Nested Routes
- Router Guards
- Routing Strategies
- Layz loading

Directives

- Built in Directives
- Custom Directives
- Host Listener & Host Binding
- Inputs & Configuration

Day 05

HTML5 Storage with Angular

- Introduction to localStorage
- Introduction to sessionStorage
- Security with HTML5 storage
- Cookie fundamentals
- HTML5 Storage for Application Cache

Unit Testing Angular

- Test Driven Development
- Introduction to Jasmine
- Introduction to Karma
- Jasmine Test Suites, Specs, Expectations
- Jasmine Matchers
- Jasmine Custom Matchers
- Setup Injection, Environment for Angular testing
- Testing Pipes
- Testing a Components
- Testing a Service
- Testing with HTTP
- Mocking HTTP, Services
- Testing Directives

Angular AoT Compilation

- AoT Compilation
- AoT Module Bootstrapping

Setup Release Environments

- Release Build
- Bundle Creation
- Source map generation with ts-config