

Angular

Overview

Angular is one of the popular web frameworks for building modern web applications. Angular is built on top of Typescript lang. It's a complete framework that comes bundled with necessary tools like routing, HTTP request, error handling, template engine etc to create a Single Page application.

This training program is specifically designed to facilitate Angular framework knowledge to candidates. This program will enable the candidates to start working on an existing Angular project or start creating new Angular projects.

Table of Contents

Prerequisites

1. Basic understanding of HTML
2. Basic understanding of CSS
3. Basic understanding of JavaScript

ES6

- let and const keyword
- arrow functions
- for...of and for...in loop
- array and object destructuring
- advanced function usage
- promises
- async - await

Typescript

- type definition in Typescript
- data types
- working with classes
- interface
- type
- generic type
- decorators

Angular

- Concept of Single Page Applications
 - What is it?
 - How does it work?
 - What is the benefit of creating a SPA?
 - Structure of a SPA
- Introducing Angular
 - What is Angular?
 - What role does Angular play in a SPA?
 - Why is Angular useful when we have other technologies for creating a SPA?
- Core Angular Concepts (a brief overview)
 - Modules
 - Components
 - Services
 - Directives
 - Routing
- Angular CLI
 - What is it?
 - Why should it be used?
 - How to install it
 - Creating a new angular app (*workspace and application difference*)
 - Explain the structure of the app and how to run it
- Angular components
 - What is a component?
 - What is a component made up of?
 - How to create a component using CLI?
- Angular Modules
 - What is a feature module?
 - How to create it?
 - How to put components inside a feature module?
 - How to create a shared module?
- Angular Material
 - Installation
 - Setup, theming,
 - Using Material Components in Angular App
- Data Binding
 - What is it?
 - Types
 - One Way (interpolation and property binding)
 - Two Way (using ng-model)

- Event Binding
- Component Communication
 - ParentChild using @Input()
 - Child Parent using @Output
 - Sibling communication using an Event Hub pattern
- Angular Built in directives
 - ngIf, ngFor, ngSwitch, ngSwitchCase, ngSwitchDefault, ngClass, ngStyle
 - Creating and using custom directives
 - Show them how to create structural and attribute directives
- Pipes in Angular
 - What & Why?
 - Built in pipes date, currency, percent, json, number
- Services and DI in Angular
 - What is Dependency Injection and why is it required?
 - How does DI work in Angular
 - What and Why services?
 - Service providers and injectors
 - Injector hierarchy in Angular(root, module and component)
 - How to create and use a service?
- RxJs in Angular Observables
 - What & Why Observables?
 - What is RxJs (just overview)
 - What are Subjects and BehaviorSubjects?
- HTTP Communication in Angular
 - What is HttpClient?
 - How to register it?
 - Create Angular Service which uses HttpClient and calls a REST API - example CRUD operation
- Routing
 - What & Why routing?
 - Angular Router features
 - How to setup routing
 - Talk about route config file
 - RouterModule.forRoot() method
 - Router links and router outlet
 - Passing params to route
 - Fetching params from a route
- Unit testing
 - Karma test runner
 - Jasmine testing framework
 - writing unit test for Angular application