ANGULAR 6

Lesson 00

Angular Framework…

- Angular is a framework for building client side web applications in HTML, CSS and JavaScript / Typescript
- Angular is written in Typescript. It implements core and optional functionality as a set of Typescript libraries that you import into your apps.
- Sponsored and maintained by Google.
- Angular applications are built around a design pattern called *Model-View-Controller* (MVC)

Capgemini

Why Angular?...

- Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges
- Cross Platform
- Speed and Performance
- Productivity
- Full Development Story

Capgemini

What you will learn...

- Develop modern, complex, responsive and scalable web applications with Angular
- Use the gained, deep understanding of the Angular fundamentals to quickly establish as frontend developers
- Fully understand the architecture behind an Angular application and how to use it
- Create single-page applications with on of the most modern JavaScript frameworks out there
- Troubleshoot common Angular errors & Master the best practices
- Design and Implement Application Authentication and Authorization from scratch

Pre-requisites

- HTML5 fundamentals
- CSS3 fundamentals
- JavaScript Basics

Intended Audience

- Web application developers

ANGULAR 6 – DAY 1

ES6 & Typescript

- Var, Let and Const keyword
- Arrow functions, default arguments
- Template Strings, String methods
- Object Destructuring
- Spread and Rest operator

- Typescript Fundamentals
- Types & type assertions
- Typescript OOPS - Classes, Interfaces, Constructor, etc
- Creating custom types
- Decorators

Introduction to Angular Framework

- Introduction to Angular Framework, History & Overview
- Environment Setup
- Angular CLI, Installing Angular CLI
- NPM commands & package.json
- Bootstrapping Angular App, Components, AppModule
- Project Setup, Editor Environments
- First Angular App & Directory Structure
- Angular Fundamentals, Building Blocks
- MetaData

ANGULAR 6 – DAY 2

Essentials of Angular

- Component Basics
- Setting up the templates
- Creating Components using CLI
- Nesting Components
- Data Binding - Property & Event Binding, String Interpolation, Style binding
- Two-way data binding
- Input Properties, Output Properties, Passing Event Data
- Case Study

Templates, Styles & Directives

- Template, Styles, View Encapsulation, adding bootstrap to angular app
- Built-in Directives
- Creating Attribute Directive
- Using Renderer to build attribute directive
- Host Listener to listen to Host Events
- Using Host Binding to bind to Host Properties
- Building Structural Directives

ANGULAR 6 – DAY 3

Capgemini

Pipes, Services & Dependency Injection

- Parametrized Pipes
- Chaining Multiple Pipes
- Creating a Custom Pipe
- Creating a Filter Pipe
- Pure and Impure Pipes (or: How to "fix" the Filter Pipe)
- Understanding the "async" Pipe
- Services
- Dependency Injections
- Creating Data Service
- Understanding Hierarchical Injector
- Services for Cross Component Communication
- Injection Tokens

Capgemini

Template-Driven and Reactive Forms

- Template-Driven vs Reactive Approach
- Understanding Form State
- Built-in Validators & Using HTML5 Validation
- Grouping Form Controls
- FormGroup, FormControl, FormBuilder, FormArray
- Forms with Reactive Approach
- Predefined Validators & Custom Validators
- Async Validators
- Showing validation errors

ANGULAR 6 – DAY 4

Capgemini

Components Deep Dive / Routing

- Component Life Cycle Hooks
- Reusable components in angular using <ng-content>
- ng-content and @ContentChild
- Inter-component communication (using Input, Output and Services)

- Navigating with Router links
- Understanding Navigation Paths
- Navigating Programmatically
- Passing Parameters to Routes
- Passing Query Parameters and Fragments
- Setting up Child (Nested) Routes
- Passing static data on routes
- Map() operator and switchMap() operator
- Redirecting and wildcard routes
- Outsourcing Route Configuration (create custom module)

Http Requests / Observables

- HTTP Requests
- Sending GET Requests
- Sending a PUT Request
- Transform Responses with Observable Operators (map())
- Using the Returned Data
- Catching Http Errors
- Pipe(), map(), catchError()
- Interceptors

- Basics of Observables & Promises
- Analysing a Built-in Angular Observable
- Building & Using a Custom Observable
- Understanding Observable Operators
- Using Subjects to pass and listen to data

ANGULAR 6 – DAY 5

Capgemini

Deployment / Authentication

- Deployment Preparations
- JIT vs AOT Compilation
- Setup development env and backend env
- Environments configuration
- Linting
- Deployment

- How Authentication works in SPA
- JSON Web Tokens
- Signup, Login and logout application
- Router Protection, Route Guards
- CanActivate interface
- Checking and using Authentication Status

ANGULAR 6 – DAY 6

Animation, Material, Redux

- Angular Animation
- Material Design Bootstrap with Angular
- New features in angular cli
  - *ng update*
  - *ng add*
- Angular Elements
- CLI workspaces
- RxJs 6 Support

- Redux introduction, Installing redux
- Building Blocks of Redux
- Working with Actions, Reducers

ANGULAR 6 – DAY 7

Modules, Security, Optimization

- Modules, Feature Modules, Routes with Feature module, Shared Modules
- Lazy loading of modules, cross module communication
- Protecting Lazy Loaded Routes
- Understanding Core Module
- Create Basic Core Module
- How Module and Services work together
- Optimization and performance issues
- Basic Security contexts

References

- https://angular.io/docs
- https://angular.io/guide/quickstart
- https://angular.io/tutorial