Course Outline

MOC 55267 A Mastering Angular

Course Summary

Description

This training will take you from being an average Angular developer to a great one. You'll gain more insight in the workings of Angular and you'll explore more advanced programming techniques like RxJS and working with Redux. Advanced Forms will hold no secrets and you will acquire the assets to make your Angular application production ready. This course is constantly being updated to the latest version of Angular, currently Angular 6.

Objectives

At the completion of this course, Students will be able to:

- Work effectively with Reactive Extensions.
- Improve performance of an Angular application.
- Use ngrx to apply the redux pattern to an Angular application.
- Simplify architecture with smart and dumb components.
- Creating an efficient structure for large applications.
- Work with more advanced forms.
- Add server-side rendering with Angular Universal.

Topics

- Reactive Extensions for JavaScript
- Change Detection
- State management with Redux
- Bringing Redux to Angular with ngrx
- The ngrx Store
- ngrx Reducers
- Smart and Dumb Components
- Structuring an Application
- Forms Advanced
- Angular Universal

Audience

This course targets professional web developers that really want to master Angular. Participants of this course need to have a decent understanding of Angular and TypeScript

Prerequisites

These prerequisites for this course are:

- Experience with TypeScript.
- Experience with Angular.
- An IDE for web development like Visual Studio Code or WebStorm.

Duration

Two Days

Course Outline

MOC 55267 A Mastering Angular

Course Outline

I. ***Reactive Extensions for JavaScript***
Reactive programming is a world on its own and is not necessarily tied to Angular. However Angular uses RxJS in many of its APIs, so we must have a proper understanding of what it means to write reactive code. RxJS has a steep learning curve. But once you master it, there is no going back.
A. What are Reactive Extensions
B. Observable and Observer
C. Subjects
D. Cold versus Hot Observables
E. Making Async Calls
F. Combining Observables
G. Error Handling
H. Dealing with Backpressure
Lab 1: Search Spotify
- Baby steps
- Debounce
- Calling an async method
- Cancellation
- Error handling
- Combining streams

II. Change Detection
This module will teach you about the internals of Angular. This knowledge is primarily useful for performance tuning and debugging strange behavior.
A. Zones
B. How Change Detection Works
C. Immutables and Observables

III. ***State management with Redux***
With the increasing complexity of client-side applications, a lot of state has to be kept. It's quite a challenge to keep all state consistent. Redux introduces a pattern to manage this state in a convenient way, while keeping your UI up to date as well.
A. Why Redux?
B. Smart & dumb components
C. Major Principles
D. The Store, Actions and Reducers
E. Tools
Lab 1: Redux Todo App
A. The Store
B. Actions and Action Creators
C. Reducers
D. Reading data and dispatching events
E. Debuggin using the Redux DevTools

IV. ***Bringing Redux to Angular with ngrx***
ngrx is an implementation of Redux for Angular. It puts the theory of the previous module to practice.
A. Using RxJS within Redux
B. Efficient Slicing
C. Using Async Pipes

V. ***The ngrx Store***
This module explains the details of the store in ngrx.
A. Responsibilities
B. Normalizing Data
C. Initializing the Store

VI. ***ngrx Reducers***
This module explains how reducers work and how to make them more manageable. Also, you will learn how to deal with side-effects.
A. L
B. Useful Operators
C. Splitting Up Reducers
D. ngrx Effects
Lab 1: Spotify Reducers
- Playlist action and reducers
- The 'recent' list
- The search actions
- Displaying tracks
- Combining reducers

VII. ***Smart and Dumb Components***
Should everyone be able to talk everything? No! Here we learn how to avoid chaos by splitting up our component into smart and dumb ones.
A. Characteristics of Dumb Components
B. Characteristics of Smart Components
C. Performance Impact

MOC 55267 A Mastering Angular

Course Outline

Lab 1: Spotify Dispatch and Select

- Smart and dumb components
- Selecting in the Playlist component
- Selecting in the Recent component
- Dispatching from the App component
- Search
- View Tracks

VIII. ***Structuring an Application***

Applications grow larger, code bases become unmanageable. Unless you structure them right. This module will show you how.

A. Domain, Routing, Core and Shared Modules
B. Exporting and Providing
C. Clean Imports
D. Creating Libraries

Lab 1: Structuring the Weather App

- Creating modules
- Creating Feature Components

IX. ***Set up routing***

Lab 1:. With Preboot

- Services
- Clean imports
- Using shared module
- Using a library

X. ***Forms Advanced***

This module explains how to work with dynamic forms. These are forms that change as the user provides information. Using nested forms allows for more manageable chunks of code.

A. Lessons
B. Dynamically Adding Elements
C. FormArray
D. Nested Forms

Lab 1: FormArray and Nesting

- Working with FormArray
- Nesting Forms

XI. ***Angular Universal***

Angular takes a lot of measures to make your app high-performant. But you can take it a bit further, especially when it comes to load times. In this Module you will learn about Ahead-Of-Time compilation and hot loading using Angular Universal. Here you'll learn the difference between a good and a great application and how to please the elders of the internet.

A. Server-side rendering with Angular Universal
B. Hot-Loading