

MOC 55266 Building Web Apps with Angular and TypeScript

Course Summary

Description

In recent years a lot of effort has gone into making HTML and JavaScript a better place for web apps rather than just web content. With Angular, you can exploit these new and modern concepts to take it to the next level. By using a componentized approach, Angular is better equipped than ever to build performant data-driven web-apps. While Angular takes care of data binding, navigation and server communication; TypeScript allows you to use the most advanced features JavaScript has to offer on any browser. Features like strong typing make your application more maintainable, better structured and flexible. This course is constantly being updated to the latest version of Angular, currently Angular 6. Enjoy this beautiful synergy between Google's Angular and Microsoft's TypeScript.

Objectives

After completion of this course, you will be able to:

- Set up and write application with TypeScript.
- Create and manage Angular applications.
- Use data binding to update your screen.
- Split up complex interfaces into components.
- Write their own directives and pipes.
- Create complex forms with validation.
- Communicate with a REST backend.
- Write an Single Page Application with client-side routing

Topics

- Introduction to Angular
- Strongly Typed JavaScript with TypeScript
- Core Concepts
- Data binding
- Components
- Attribute Directives
- Structural Directives
- Dependency Injection and Providers
- Pipes
- Working with Forms
- Talking to the Server
- Building a Single Page Application

Audience

This course targets professional web developers who are looking for a kick-start into the world of Angular and TypeScript. Participants of this course need to have a good understanding of JavaScript, HTML and CSS and a notion of node.js and npm.

Prerequisites

The prerequisites for this course are:

- A good understanding of JavaScript.
- Basic understanding of HTML and CSS.
- At least a notion of node.js and npm.
- An IDE for web development like Visual Studio Code or WebStorm.

Duration

Three Days

MOC 55266 Building Web Apps with Angular and TypeScript

Course Outline

I. *Introduction to Angular*

In this module you'll see what Angular is all about and why it is so important in modern web development.

- A. Evolution in Web App Development
- B. Angular Core and Modules
- C. TypeScript, Dart, Plain Old JavaScript

II. *Strongly Typed JavaScript with TypeScript*

Let's face it: JavaScript was never designed for big applications. Many constructs to tame complex code like interfaces and strong typing are completely absent. And many of the new cool JS features are not implemented in current browsers.

TypeScript is the solution to both problems; allowing you to write modern, application-scale JavaScript.

- A. Writing Application-Scale JavaScript
- B. Type-Safe JavaScript Development with TypeScript
- C. Implementing Types, Classes and Inheritance
- D. Namespaces and Modules

Lab 1: Toy Store

- Getting started with a TypeScript project
- Making the models: Product, Category, Order
- Creating the Shopping Cart
- Using an external library
- Adding the code to the HTML page

III. *Core Concepts*

In this module you'll get acquainted with the most important building blocks for any Angular application. This is the foundation of all following chapters.

- A. Components
- B. Modules
- C. Services
- D. Tools

Lab 1: Inspecting a First Project

- Opening an running the project

- Elements of an applications
- Loading Modules
- tsconfig.json

IV. *Data binding*

Data binding allows you to forget about the HTML while writing JavaScript code. It allows you to inject data into a view without creating a strong dependency between the two. This results into more flexible, testable and maintainable code.

- A. The Importance of Binding
- B. Component to View
- C. Structural Directives
- D. Local Template Variables
- E. Value Conversion
- F. View to Component

Lab 1: TaskManager with Data Binding

- Modules
- Displaying a list of tasks
- Style
- Adding a Task

V. *Components*

In this module we'll dive a bit deeper into Components. You'll learn how to create a hierarchy of components and how to communicate between them.

- A. Using Multiple Components
- B. Input and Output
- C. ViewChild and ContentChild
- D. EventEmitter
- E. Directive Life Cycle

Lab 1: TaskManager: Using Multiple Components

- Adding some style
- TaskCreator component
- TaskCard component
- TaskList component
- App component
- Registering components

VI. *Attribute Directives*

What if you want to add custom behavior to an existing element? In this module you'll explore existing directives

MOC 55266 Building Web Apps with Angular and TypeScript

Course Outline (Cont.)

Like NgClasses and NgStyle; and learn how to build your own.

- A. What are Attribute Directives?
- B. Built-in Attribute Directives
- C. Custom Attribute Directives

VII. *Structural Directives*

With structural directives you can change the flow in your HTML. For example, how do you generate HTML dynamically based upon your data without having to write HTML in your JavaScript?

- A. What are Structural Directives?
- B. Built-in Structural Directives
- C. Templates
- D. Custom Structural Directives

VIII. *Dependency Injection and Providers*

Dependency Injection (DI) is the art of taking two strongly coupled objects and tearing them apart. This helps you write understandable, maintainable and testable code. It's not really a choice in Angular either: you have to do it. Angular has an entire mechanism based on Providers to support DI which you will explore in this module.

- A. Terminology
- B. Dependency Injection Basics
- C. Services
- D. Providers
- E. Factories
- F. Injection Tokens

Lab 1: Creating a Task Service

- Implementing the Service
- Providing the service
- Updating the AppComponent
- Replacing the mock service with a real service

IX. *Module 9: Pipes*

Pipes are a convenient way to make little changes to values in a view. This includes formatting and filtering of data. It's easy to use and to extend the possibilities of pipes.

- A. Using a Pipe
- B. Built-in Pipes
- C. Custom Pipes
- D. Pure versus Impure

Lab 1: Temperature Pipe

- Create a pipe for displaying temperatures in Kelvin, Celsius and Fahrenheit
- Use your newly created pipe

X. *Working with Forms*

Forms are essential to any app that allows you to manage data. You need to do more than just data binding. You need validation, automatic formatting, respond to data changes as so on. Angular provides two different approaches to dealing with this: Template-driven forms and Model-driven forms. This module will explore both.

- A. What's in a Form
- B. Responding to Changes
- C. FormBuilder
- D. Data Validation

Lab 1: Task Editor Form

- A. The TaskEditor component
- B. Basic Validation
- C. Multiple validators
- D. Custom validators

XI. *Talking to the Server*

This module will teach you how to retrieve and send data to your backend. We will focus on REST and use RxJS's Observables to get the job done.

- A. Sending and Receiving Data
- B. HttpClient Module
- C. HTTP Interceptors
- D. Observables versus Promises

Lab 1: Working with Observables

- Update Components not working with observables

Lab 2: Talking to the Server

- Providing the HttpClient Module
- Implementing the HTTP service

MOC 55266 Building Web Apps with Angular and TypeScript

Course Outline (cont.)

XII. *Building a Single Page Application*

Instead of hopping from one page to the next, you can design your website as a Single Page Application. This makes your website feel and perform more like an application. SPAs have many advantages, but are usually difficult to implement. This module will teach about the constructs available in Angular to build a SPA.

- A. What is a SPA
- B. Router Module
- C. Route Configuration
- D. Parent-Child Navigation
- E. Route Guards

Lab 1: 7Building a SPA: Rabbit Rescue

- Replace static HTML with components and templates
- Set up routing per feature area
- Linking the feature area with root area