# Building web apps with Angular and TypeScript (MS-55266)

**Modality: Virtual Classroom**

**Duration: 3 Days**

## About this Course:

This intermediate-level 3 Days Training Program is specifically designed for IT Professionals & Experienced Web Developers who want to learn the fundamental of building Web Applications with JavaScript & HTML. This course provides a comprehensive overview of configuration Angular & TypeScript and helps professionals make the best use of its core features and functionalities. The primary objective of this course is to help candidates learn the skills required to build data-driven web apps using Angular. The Key Concepts of Navigation, Server Communication, and Data Bindings in Angular are also elaborated in this course.

In addition to this, professionals also get to learn about advanced TypeScript Features & Functionalities. Candidates get to learn about maintaining and structuring web applications and learns the art of designing flexible web apps. Through this course, professionals will learn how to utilize the strong synergy between Microsoft TypeScript and Google Angular to build innovative web applications.

## Course Objectives:

The core objective of this course is to help professionals develop a better understanding and sound knowledge of the following key concepts:

- Fundamentals and Essentials of Angular
- Setting up a TypeScript
- Angular Applications Development & Management
- Implementation of Data Binding
- Splitting up Complex and Large Interfaces into Small Components
- Creating Customized Directives & Pipes
- Complex Forms Development with Validation
- Fundamentals and Essentials of REST API
- Client-Side Routing & Single Page Applications

## Audience:

This course is specifically tailored for the following group of professionals and interested candidates:

- Seasoned Web Developers
- Candidates interested in Learning TypeScript and Angular
- Candidates striving to pursue Professional Career as a Web Developer

## Prerequisites:

Professionals planning to enroll in the Building web apps with Angular and TypeScript (MS-55266) course must comply with the following prerequisites:

- Fundamental Knowledge of JavaScript and HTML
- Know-how of CSS and Familiarity with npm and node.js

# Course Outline:

### Module 1: Introduction to Angular

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

### Lessons

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

After completing this module, students will be able to:

- Have an understanding of modern web UI technologies and the role they play in modern development.

### Module 2: 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.

### Lessons

- Writing Application-Scale JavaScript
- Type-Safe JavaScript Development with TypeScript
- Implementing Types, Classes and Inheritance
- Namespaces and Modules

### Lab : Toy Store

- Getting started with a TypeScript project
- Making the models: Product, Category, Order…
- Creating the ShoppingCart
- Using an external library

- Adding the code to the HTML page

After completing this module, students will be able to:

- Have an understanding of the benefits of TypeScript
- Use core features of TypeScript
- Set up a new TypeScript project
- Compile and run TypeScript project

## Module 3: 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.

### Lessons

- Components
- Modules
- Services
- Tools

### Lab : Inspecting a First Project

- Opening an running the project
- Elements of an applications
- Loading Modules
- tsconfig.json

After completing this module, students will be able to:

- Understand the goal of Components and Services.
- Group Components and Services into Modules.

Work with Angular Tools.

## Module 4: 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.

### Lessons

- The Importance of Binding

- Component to View
- Structural Directives
- Local Template Variables
- Value Conversion
- View to Component

## Lab : TaskManager with Data Binding

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

After completing this module, students will be able to:

- Create and update a view using data binding.
- Handle user interactions in a component.
- Refer to elements with local template variables.
- Use pipes for value conversion.

## Module 5: 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.

## Lessons

- Using Multiple Components
- Input and Output
- ViewChild and ContentChild
- EventEmitter
- Directive Life Cycle

## Lab : TaskManager: Using Multiple Components

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

After completing this module, students will be able to:

- Split up complex components into multiple components
- Set up parent-child communication

- Work with various life cycle hooks

## Module 6: Attribute Directives

What if you want to add custom behavior to an existing element? In this module you'll explore existing directives like NgClasses and NgStyle; and learn how to build your own.

### Lessons

- What are Attribute Directives?
- Built-in Attribute Directives
- Custom Attribute Directives

After completing this module, students will be able to:

- Understand the role of attribute directives.
- Use built-in attribute directives.

Create your own attribute directives.

## Module 7: 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?

### Lessons

- What are Structural Directives?
- Built-in Structural Directives
- Templates
- Custom Structural Directives

After completing this module, students will be able to:

- Understand the role of structural directives.
- Use built-in structural directives.
- Create your own structural directives.

## Module 8: 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.

**Lessons**

- Terminology
- Dependency Injection Basics
- Services
- Providers
- Factories
- Injection Tokens

**Lab : Creating a Task Service**

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

After completing this module, students will be able to:

- Create their own services.
- Work with the various providers in Angular.
- Inject services into components and other services.

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.

**Lessons**

- Using a Pipe
- Built-in Pipes
- Custom Pipes
- Pure versus Impure

**Lab : Temperature Pipe**

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

After completing this module, students will be able to:

- Use built-in pipes.
- Create their own pipes.

- Understand the importance of pure pipes.

## Module 10: 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 a so on. Angular provides two different approaches to dealing with this: Template-driven forms and Model-driven forms. This module will explore both.

### Lessons

- What's in a Form
- Responding to Changes
- FormBuilder
- Data Validation

### Lab : Task Editor Form

- The TaskEditor component
- Basic Validation
- Multiple validators
- Custom validators

After completing this module, students will be able to:

- Choose between template-driven and model-driven forms.
- Create a form and submit the information.
- Add validation and visual feeback.

## Module 11: 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.

### Lessons

- Sending and Receiving Data
- HTTPClient Module
- HTTP Interceptors
- Observables versus Promises

### Lab : Working with Observables

- Update Components wot work with observables

## Lab : Talking to the Server

- Providing the HttpClient Module
- Implementing the HTTP service

After completing this module, students will be able to:

- Work with HttpClient to make rest calls.
- Process the result with observables.

## Module 12: 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.

### Lessons

- What is a SPA
- Router Module
- Route Configuration
- Parent-Child Navigation
- Route Guards

## Lab : Building a SPA: Rabbit Rescue

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

After completing this module, students will be able to:

- Create a single page application.
- Create feature areas.
- Set up routing per feature.
- Intercept navigation with guards.