Building web apps with Angular and TypeScript

CODICE	MOC55266
DURATA	3 gg
PREZZO	1.050,00 €
EXAM	

DESCRIZIONE

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.

OBIETTIVI RAGGIUNTI

- 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.

TARGET

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.

PREREQUISITI

Before attending this course, students must have:

- 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.

CONTENUTI

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 : 7Building 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.

Additional Reading

None