

Starting .NETt core

From zero to hero in 3-days

FA-0507 · 3-days

Software Development · DevOps, Cloud & Infrastructure · Cybersecurity · Data & Analytics

Introduction

Welcome to the 3-Day .NET Core Training! This course is designed to provide you with a comprehensive understanding of .NET Core, covering both front-end and back-end development. Whether you're new to .NET Core or looking to enhance your existing skills, this training will guide you through essential concepts, hands-on coding exercises, and best practices. By the end of this course, you'll be equipped to build robust, scalable, and maintainable applications using .NET Core.

Learning outcomes

By the end of this course, participants will be able to:

- Understand the fundamentals of .NET Core and its ecosystem.
- Develop web applications using ASP.NET Core.
- Implement RESTful services with ASP.NET Core Web API.
- Integrate front-end technologies like React or Angular with .NET Core.
- Utilize Entity Framework Core for data access.
- Implement authentication and authorization in .NET Core applications.
- Deploy .NET Core applications to various hosting environments.
- Debug, test, and optimize .NET Core applications for performance.

Prerequisites

Participants should have:

- Strong understanding of programming concepts.
- Experience with C# programming language.

- Basic knowledge of HTML, CSS, and JavaScript.
- Experience with any web development framework..

Training outline

1. 1. Introduction to .NET Core

Overview of .NET Core

- History and evolution
- .NET Core vs. .NET Framework

Setting Up the Development Environment

- Installing Visual Studio Code/Visual Studio
- Installing .NET Core SDK
- Overview of CLI tools

Creating Your First .NET Core Application

- Hello World application
- Project structure and files

2. 2. ASP.NET Core Fundamentals

Understanding ASP.NET Core

- Middleware pipeline
- Request handling

MVC Pattern in ASP.NET Core

- Controllers, Views, and Models
- Razor syntax and views

Routing in ASP.NET Core

- Defining routes
- Attribute routing

3. 3. Building RESTful Services with ASP.NET Core Web API

Introduction to RESTful Services

- Principles of REST
- HTTP methods (GET, POST, PUT, DELETE)

Creating a Web API Project

- Project setup
- Creating controllers and actions

Data Transfer Objects (DTOs)

- Defining and using DTOs

Versioning and Documentation

- API versioning strategies
- Using Swagger for documentation

4. 4. Front-End Integration

Introduction to Front-End Frameworks

- Overview of React and Angular

Setting Up a React or Angular Project

- Project setup and configuration
- Creating components

Consuming .NET Core APIs in Front-End

- Making HTTP requests
- Handling responses and errors

State Management

- Managing state in React/Angular applications

5. 5. Data Access with Entity Framework Core

Introduction to Entity Framework Core

- Overview and advantages
- Setting up EF Core in .NET Core project

Creating and Configuring the Database Context

- Defining the DbContext class
- Connection strings and configurations

Creating and Managing Entities

- Defining entity classes
- Migrations and updating the database

Querying Data

- LINQ queries
- Asynchronous operations

6. 6. Authentication and Authorization

Introduction to Authentication and Authorization

- Differences and importance

Implementing Authentication in ASP.NET Core

- Using Identity framework
- JWT Authentication

Role-Based and Policy-Based Authorization

- Defining roles and policies
- Applying authorization in controllers and actions

7. 7. Deployment and Hosting

Overview of Deployment Options

- On-premises vs. cloud hosting

Deploying to Azure

- Setting up an Azure account
- Deploying a .NET Core application to Azure App Service

Dockerizing .NET Core Applications

- Introduction to Docker
- Creating and deploying Docker containers

This outline provides a structured approach to learning .NET Core, ensuring a balance between theoretical knowledge and practical application. By the end of the course, you'll have a solid foundation in .NET Core development, ready to tackle real-world projects with confidence.