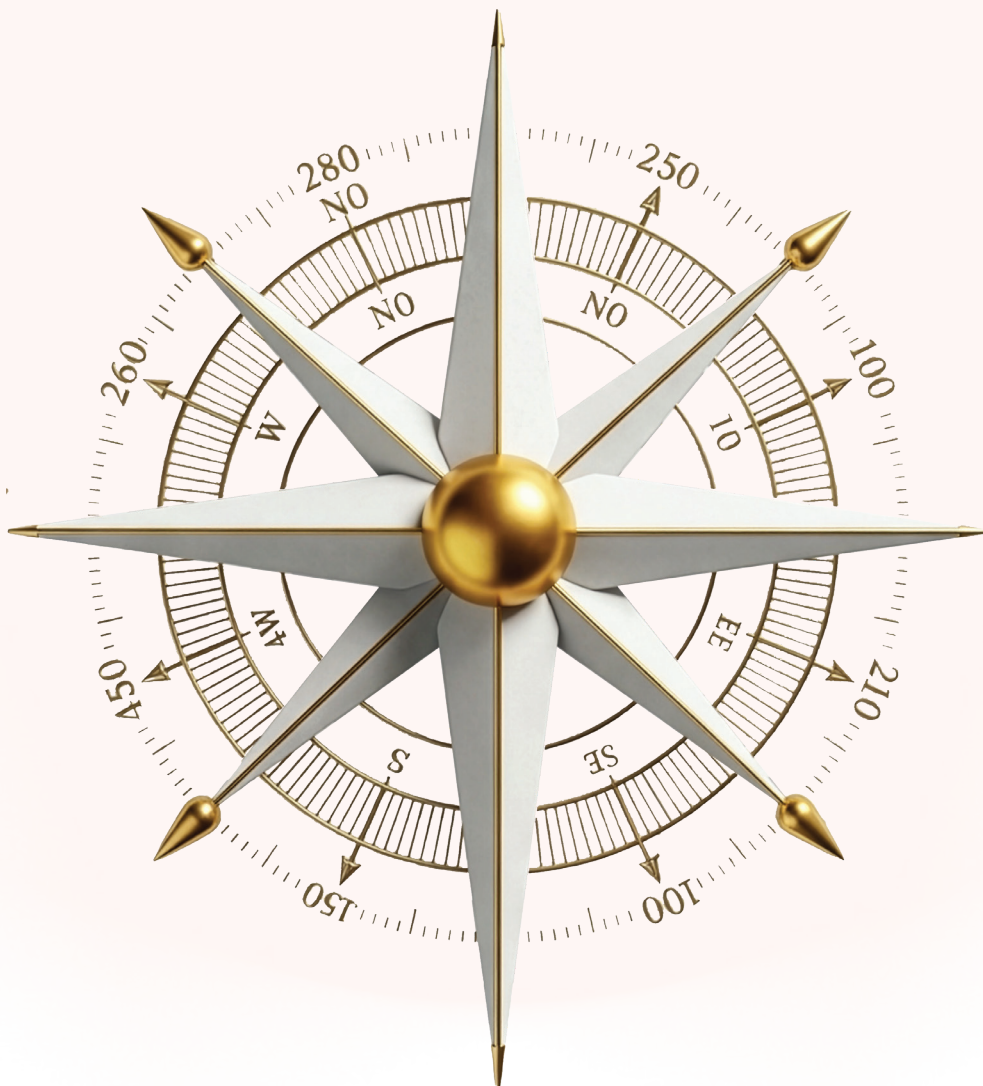




ASP.NET Core and C# Training Course: Web APIs, Entity Framework Core and xUnit Testing

1 – 5 February 2027

Paris - Right Bank business hotel



ASP.NET Core and C# Training Course: Web APIs, Entity Framework Core and xUnit Testing

Ref: 1419_23599 **Date:** 1 – 5 February 2027 **Location:** Paris - Right Bank business hotel **Fees:** 23500 SAR

Introduction:

ASP.NET Core and C# training on Web APIs, Entity Framework Core and xUnit testing is a 5-day course for C# application developers and back-end engineers, ending with a secured CRUD Web API solution with a Razor Pages client and its xUnit test suite. Business C#/.NET web applications often grow around fat controllers, manual database scripts, hard-coded secrets and untested endpoints, so releases bring data errors and broken sign-in. Nominees already write C# at work, and every day runs as a hands-on lab in Visual Studio and the dotnet CLI. CoreConcept Training Center delivers this ASP.NET Core course.

Course Objectives:

- Write C# code using records, nullable reference types, pattern matching, LINQ and async methods in an object-oriented domain model
- Structure an ASP.NET Core solution with the WebApplication host, an ordered middleware pipeline and dependency injection service lifetimes
- Build server-rendered pages with MVC controllers and Razor Pages, and HTTP endpoints with controller-based and minimal Web APIs
- Persist and evolve relational data with Entity Framework Core, the Fluent API and code-first migrations, returning validated input and ProblemDetails errors
- Secure applications with ASP.NET Core Identity, JWT bearer tokens and authorisation policies, supported by configuration, logging and xUnit tests
- Deliver a secured CRUD Web API solution with a Razor Pages client, xUnit test suite and Docker or Azure App Service deployment

Target Audience:

- Application developers responsible for building and extending business web applications in C# and ASP.NET Core
 - Back-end engineers responsible for HTTP endpoints, data access and integration logic in ASP.NET Core
 - Developers responsible for moving older ASP.NET Web Forms or MVC applications onto ASP.NET Core
 - Technical leads responsible for solution structure, code review and security settings in C#/.NET teams
 - Test engineers responsible for automating checks against C# web applications and their endpoints
-

Course Outline:

Day 1: Modern C# Language, Object-Oriented Design and SDK Solution Layout

- C# Records, Nullable Reference Types and Pattern Matching Expressions
- Interfaces, Abstract Classes and SOLID Principles in Domain Classes
- LINQ Query and Method Syntax Over In-Memory Collections
- Async and Await With Task-Based Pattern and Cancellation Tokens
- .NET SDK Solution Layout With dotnet CLI and NuGet Packages

Day 2: ASP.NET Core Hosting Model, Middleware Pipeline and Dependency Injection

- Program.cs WebApplication Builder and Kestrel Web Server Setup
- Middleware Pipeline Ordering With UseRouting, UseAuthentication and UseExceptionHandler
- Built-In Dependency Injection With Transient, Scoped and Singleton Lifetimes
- MVC Controllers, Views and Tag Helpers for Server-Rendered Pages
- Razor Pages PageModel Handlers and Model Binding for Forms

Day 3: Web API Endpoints, Entity Framework Core Data Access and Validation

- Controller-Based Web APIs With Attribute Routing and Action Results
- Minimal API Endpoint Groups and OpenAPI Documentation With Swagger
- Entity Framework Core DbContext, Fluent API Mapping and Relationships
- EF Core Code-First Migrations and Model Snapshot History Table
- Data Annotations Model Validation and ProblemDetails Error Responses

Day 4: Identity, Authorisation, Configuration, Logging and Automated Testing

- ASP.NET Core Identity User Store and Cookie Authentication
- JWT Bearer Tokens and Policy-Based Authorisation Requirements
- Options Pattern, appsettings.json Environments and User Secrets Configuration
- ILogger Structured Logging and Health Check Endpoints
- xUnit Unit Tests and WebApplicationFactory Integration Tests With SQLite

Day 5: Lab Building and Deploying a Secured CRUD Web API Solution

- Case Solution Brief, Entity Model and Endpoint Contract Definition
 - Lab Implementation of EF Core Data Layer and CRUD Endpoints
 - Lab Integration of JWT Security, Validation and Razor Pages Client
 - Lab Packaging With Dockerfile and Azure App Service Deployment
 - Secured CRUD Web API Solution Completion With xUnit Test Run
-

Skills You Will Gain:

- C# Language Proficiency
- LINQ Query Writing
- Middleware Pipeline Configuration
- Razor Pages Development
- Web API Endpoint Design
- Entity Framework Core Data Modelling
- Token-Based Authentication Setup
- Integration Test Automation

Why Attend This Course:

- Hand the application owner or development lead a secured CRUD Web API solution with its Razor Pages client, xUnit test suite and deployment notes
- Decide between MVC and Razor Pages, controllers and minimal APIs, and cookie or JWT authentication for each new ASP.NET Core application
- Prevent unapplied schema changes, leaked connection strings and unauthorised endpoint calls that cause failed releases and security incidents
- Pass on the solution layout, middleware ordering checklist and WebApplicationFactory test fixture to other C# developers

Conclusion:

Back at work, the participant hands the development lead or application owner a secured CRUD Web API solution with a Razor Pages client, an xUnit and WebApplicationFactory test suite and deployment notes for Docker or Azure App Service. The lead uses it to decide how new ASP.NET Core applications are layered, how schema changes travel through migrations and which authentication scheme each application adopts. After the first release built this way, the team should review failed test runs, migration history, authorisation failures and application logs.

Frequently Asked Questions FAQ:

What should participants know before the ASP.NET Core and C# training course?

Participants should already write C# or another object-oriented language at work and understand HTTP requests, JSON and basic SQL. A laptop able to run Visual Studio or Visual Studio Code with the dotnet CLI and Docker Desktop is used in the daily labs.

How does the ASP.NET Core and C# training course differ from a Java back-end or game scripting course?

It applies C# to web applications on ASP.NET Core: middleware, Razor Pages, Web APIs, Entity Framework Core, Identity and xUnit. Java back-end courses use a different language and framework, and game scripting courses apply C# to engine components rather than HTTP applications.

Why do teams choose ASP.NET Core for web applications and Web APIs?

ASP.NET Core runs cross-platform on Windows, Linux and macOS, hosts in the Kestrel web server and combines MVC, Razor Pages and Web APIs in one programming model with built-in dependency injection, so one team builds pages and APIs with the same tools.

What do participants take back from the ASP.NET Core and C# training course?

Participants take back a secured CRUD Web API solution with a Razor Pages client, EF Core migrations, an xUnit and WebApplicationFactory test suite, a Dockerfile and deployment notes, plus a middleware ordering checklist for the next ASP.NET Core project.
