

Web API Development with ASP.NET Core 6 Eğitimi

Eğitim Hakkında

Web API Development with ASP.NET Core 6 Eğitimi; tüm beceri düzeyleri içindir ve C# ve ASP.NET çerçevesinde bilgi edinmiş ve becerilerini REST ilkelerini kullanarak yüksek düzeyde ölçeklenebilir WEB API'lerini öğrenmek için kullanmak isteyen yeni başlayanlar ve orta düzey geliştiriciler için uygundur. Web API Development with ASP.NET Version 6 (.NET 6 olarak da bilinir) kullanacağız ve RESTful WEB API oluşturacağız .

Eğitim esnasında API'ler ve uç noktalar oluşturma konusunda çok fazla deneyim kazanacaksınız ve sonunda kendi ASP.NET Core API'lerinizi oluşturmak için yeterli pratik bilgiye sahip olacaksınız.

REST ilkelerinin neler olduğunu öğrenerek başlayacağız, ASP.NET Core'un yeni bir projenin parçası olarak oluşturduğu dosyaları öğreneceğiz. Ardından domain ve domain modellerimizi oluşturup inceleyeceğiz.

Ardından Entity framework core yükleyeceğiz ve EF Core geçişlerini kullanarak SQL Server Veritabanımızı oluşturacağız.

Ardından API'miz için denetleyiciler oluşturacağız ve bunları Swagger UI kullanarak test edeceğiz. Ayrıca; Domain and Data models kavramlarını anlayıp uygulayacağız ve ASP.NET CORE Web API'mizi oluşturmak için clean coding techniques kullanacağız .

Neler Öğreneceksiniz

- Understand APIs
- REST Principles
- Creating New ASP.NET Core API Project
- Understand .NET 6 and New API Project In .NET 6
- Learn And Install Entity Framework Core (EF Core) In ASPNET Core API Project
- Understand Domain Models and Repository Pattern
- Use Swagger To Test ASPNET Core WEB API project
- Understand Domain models and DTOs
- Learn How To Map Models To One Another Using AutoMapper
- Learn Asynchronous Programming (Async Await)
- Validate ASPNET Core WEB API and Learn and Use Fluent Validations To Validate the API
- Understand and Add Authentication and Authorization To ASPNET Core API and Create

- Read and Write Users To Test this Via Postman
- Incorporate Authentication Into SwaggerUI

Eğitim İçeriği

Introduction

- Evolution of .NET and .NET Core
- .NET SDKs and Runtimes
- Visual Studio and Visual Studio Code

.NET 6.0 SDK

- Installation
- Version Management
- Command-Line Interface (CLI)

What's New in C#

- Record Types
- Init Only Setters
- Nullable Reference Types
- Global Using Directives
- File-Scoped Namespace Declarations
- Top-Level Statements

ASP.NET Core Application Architecture

- NuGet Packages
- Application Startup
- Hosting Environments
- Middleware and the Request Pipeline
- Services and Dependency Injection

Application Configuration

- Configuration Providers and Sources
- Configuration API
- Options Pattern
- HTTPS and HTTP/2

Request Routing

- RESTful Services
- Endpoint Routing
- Route Templates
- Route Constraints
- Route Template Precedence
- Attribute-Based Routing

Models

- Persistence Ignorance
- Dependency Inversion
- Asynchronous Data Access
- Object-Relational Mapping
- Entity Framework Core
- Dapper ORM

Controllers

- Responsibilities
- Requirements and Conventions
- Dependencies
- Action Results
- ApiController Attribute

Web APIs

- API Controllers
- OpenAPI / Swagger
- Testing APIs
- Content Negotiation
- CRUD Operations
- Patch Requests
- Cross-Origin Resource Sharing (CORS)

Data Validation

- Introduction
- Data Annotations
- Model Binding

Error Handling

- Best Practices
- HTTP Error Status Codes
- Developer Exception Page

Logging

- Configuration
- ILogger
- Serilog and Seq

Testing

- Unit Testing
- xUnit
- Testing Controllers
- Integration Testing

Security

- Authentication
- ASP.NET Identity
- Authorization
- Web API Authentication
- JSON Web Tokens (JWT)
- OAuth 2.0 and OpenID Connect
- Secrets Management

Remote Procedure Calls (gRPC)

- Introduction
- Protobuf
- Server
- Client
- Limitations

Deployment

- dotnet publish
- Kestrel
- IIS
- Docker