

Updated on 15/10/2024

Sign up

WebApi .NET Core training

2 days (14 hours)

Presentation

WebAPI .NET Core training gives you the skills you need to create and maintain robust, scalable and secure APIs, based on modern development best practices.

This course highlights the importance of documentation and SDKs, as well as the management of essential aspects such as asynchronicity, idempotency and API versioning.

Web APIs are at the heart of many modern applications, enabling interaction between various systems and services. With [.NET Core](#), you'll learn how to develop high-performance APIs while benefiting from a cross-platform, modular, high-performance platform.

Our training will enable you to master the design, implementation, securing and maintenance of WebAPIs in .NET Core, while taking inspiration from reference API models, recognized for its documentation, clear API and best practices.

This course is aimed at any developer wishing to improve their skills in creating professional APIs, while respecting modern industry standards.

Objectives

- Developing robust, high-performance APIs with .NET Core
- Understand and implement essential concepts such as rate limiting, versioning and asynchronism
- Integrate Swagger for interactive, up-to-date API documentation
- Implement the REPR pattern for optimized, maintainable APIs
- Apply best practices for managing transactions, idempotence, and securing APIs

Target audience

- Web and application developers
- Architects

Prerequisites

- Good knowledge of object-oriented programming
- Development experience with the .NET Framework
- Understanding of basic HTTP concepts and RESTful APIs

Technical requirements

- Visual Studio Code or Visual Studio installed
- Access to a source code repository, for example via GitHub

WebApi .NET Core Training Program

Introduction to WebAPI and .NET Core

- Understanding the architecture and foundations of .NET Core
- WebAPIs: role, use cases and benefits
- Creating your first WebAPI with ASP.NET Core
- Development environment configuration (Visual Studio, CLI, etc.)
- WebAPI project structure in .NET Core (controllers, routing, templates)
- Differences between .NET Framework and .NET Core for WebAPIs

HTTP request management and routing

- Handling HTTP verbs: GET, POST, PUT, DELETE, PATCH
- Implementation of conventional and attribute-based routing.
- Route customization and parameter management (URI, Query Strings)
- Data models and serialization (JSON, XML)
- HTTP response management (status, headers, bodies)
- Setting up input validation with Data Annotations

Documentation and Swagger

- Introduction to Swagger: purpose and use in API documentation
- Implementing Swagger in an ASP.NET Core project
- Advanced Swagger configuration (groups, versions, filters)
- Automatic documentation generation with Swagger
- Add XML comments to improve documentation
- Overview of third-party tools for interactive API documentation (e.g. Postman)

Advanced API management: Versioning, Rate Limiting and Security

- WebAPI versioning implementation (query string, headers, route)
- Breaking changes management with good versioning.
- Rate limiting to protect API resources
- Implementation of authentication and authorization policies (JWT, OAuth2)
- Securing WebAPIs with HTTPS and CORS
- Idempotency and transaction management in WebAPIs

Performance enhancement and advanced practices

- Implementing the REPR pattern for your EndPoints
- Using caching to improve API performance
- Managing asynchronous operations with async/await in ASP.NET Core
- Atomicity and resilience in APIs: techniques for guaranteeing robust transactions
- Use of SDKs and libraries to simplify API interactions (inspired by Stripe)
- Optimize API responses with data pagination, sorting and filtering

Testing, deployment and maintenance

- Setting up unit tests for WebAPIs (XUnit, Moq)
- Integration tests to verify interaction with external databases and services
- Use Postman and similar tools to test WebAPIs manually
- Strategies for deploying WebAPIs in the cloud (Azure, AWS, Docker)
- Follow-up and monitoring of WebAPIs in production (Logging, Health Checks)
- Log management and auditing for traceability and debugging

Companies concerned

This training course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.