

ASP.NET Core 2.2 REST API Development

Foundations, Creating REST APIs, Controllers, Domain, OpenAPI (Swagger), Testing, Architecture

ASP.NET Core 2.2 is the modern approach to developing web UI and REST API applications using .NET Core 2.2. This exciting technology richly supports OpenAPI-based REST development, test-driven development, domain models, custom URL routing schemes, flexible viewing and separation of concerns.

This course covers in-depth the foundations of ASP.NET Core 2.2 and how to use it to create REST APIs. Important: this course does not cover ASP.NET MVC user interface development (e.g. use of Razor).

ASP.NET Core 2.2 comes with a range of features to build REST APIs: both the HTTPS interface and what is behind it (e.g. domain model, database access via EFCore)+[enhanced OpenAPI](#) support.

Modern REST API developers use C#, .NET Core and ASP.NET Core 2.2 to build server-side implementations, then use OpenAPI (formerly called Swagger) to describe an API and tools such as [OpenAPI-Generator](#) to auto-generate client-side stubs (e.g. for Angular 7 UI in the browser).

Contents of One-Day Training Course	
<p>Target Audience REST API developers wishing to learn about the latest approach to creating REST APIs using .NET Core 2.2 technologies.</p> <p>Prerequisites Attendees are expected to be experienced C# developers with some understanding of the HTTP protocol.</p> <p>Attendance at our <i>.NET Core 2.2 CLR Programming</i> course or equivalent experience.</p> <p>No previous experience of ASP.NET Core required, as this course explores it from the foundations upwards.</p>	<p>Overview of ASP.NET Core 2.2 “ASP.NET Core is an open-source and cross-platform framework for building modern cloud based internet connected applications, such as web apps, IoT apps and mobile backends.” [link] Goals for ASP.NET Core Advantages and challenges Web UI and Web API unified</p> <p>Review Of REST API Pipeline Developer setup required (Visual Studio) What components are needed to deliver REST solution via ASP.NET Core Quick Tour of ASP.NET Core Focus in on server-side implementation HTTP request pipeline New features of ASP.NET Core 2.2 What’s coming in ASP.NET Core 3.0</p> <p>First REST API App Walkthrough of a simple app introducing all the major features Solution layout Project template in Visual Studio 2017 ASP.NET Core 2.2 can run on both .NET Core and .NET Framework, but ASP.NET Core 3.0 will only run on .NET Core</p> <p>Important Concepts Attributes (Produces, Route, ..) Use of async HTTP methods (POST, GET, ..) Accessing parameter information Conventions Analyzers</p> <p>Controllers And Actions Plays coordination role ControllerBase ApiController ActionResult Different results for different needs Parameterizing actions Formatting results</p> <p>Domain & Data Models Domain-driven design Tour of DDD concepts (e.g. repository) In ASP.NET Core context, model is a domain model (not a data model) Domain model define repository, which data model implements (e.g. EF Core 2.2)</p> <p>Architecture Designing ASP.NET Core solutions Common design problems</p> <p>OpenAPI (Swagger) Use of add-on OpenAPI capabilities with .NET Core 2.2 to create portable representation of API and documentation</p> <p>Security Web application security Using ASP.NET Core application services Various threats</p> <p>Testing Approaches to testing web apps Unit testing and Mocking Debugging tools (e.g. Fiddler, Postman)</p> <p>Project A demonstration of how to use ASP.NET Core for more substantial workloads</p>