

Web Application Programming Interface (API)

Tahaluf Training Center 2021



Web Application Programming Interface (API)

Application Programming Interfaces (APIs) are accessible in many programming languages to help a developers to make complex useful code without any problem. They theoretical more perplexing code away from you, giving some simpler syntax to use in its place.

Students will learn how to:

- Create, compile and run web API programs using Visual Studio.
- Write and understand Web API constructs, syntax and semantics.
- Understand several concepts such as services and repositories.
- Being able to test the program using Postman.
- Connecting Database to the Web API.



Chapter 01

- 1 Overview of ASP.NET Core
- 2 Why we choose ASP.NET Core?
- 3 Differences between .NET Core and .Net Framework
- 4 Overview of ASP.NET Web API



Overview of ASP.NET Core

ASP.NET Core is a superior, cross-Platform, Support a cloud, open source system for building modern, Internet associated applications.



Overview of ASP.NET Core

With ASP.NET Core, you can:

- Build Internet of Things (IoT) applications, web applications and administrations, and portable application backends.
- Utilize a most loved advancement apparatuses on many working framework like macOS, Windows, and Linux.
- Deploy to the cloud.
- Run on **.NET Core**.



Chapter 01

- 1 Overview of ASP.NET Core
- 2 Why we choose ASP.NET Core?
- 3 Differences between .NET Core and .Net Framework
- 4 Overview of ASP.NET Web API



Why we choose ASP.NET Core?

Many developers use ASP.NET 4.x to make diverse web applications. ASP.NET Core is an rechanges and update of ASP.NET 4.x, including compositional changes and plan that outcome in a less fatty, more secluded structure.



Why we choose ASP.NET Core?

ASP.NET Core provides benefits, including:

- A unified story for building web APIs and web UI.
- Architected for testability.
- Razor Pages makes coding page-centered situations simpler and more useful.
- Blazor allows you to utilize C# in the program close by JavaScript. Offer client side and server side application rationale all composed with .NET.



Why we choose ASP.NET Core?

- Ability to develop and run on Windows, macOS, and Linux.
- Open-source and community-focused.
- Integration of modern, client-side frameworks and development workflows.
- Support for hosting Remote Procedure Call (RPC) services using gRPC.
- A cloud-ready, environment-based configuration system.



Why we choose ASP.NET Core?

- Built-in dependency injection.
- A lightweight, high-performance, and modular HTTP request pipeline.
- Ability to host on the following: Kestrel, IIS, HTTP.sys, Nginx, Apache, Docker, Side-by-side versioning.
- Tooling that simplifies modern web development.



Chapter 01

- 1 Overview of ASP.NET Core
- 2 Why we choose ASP.NET Core?
- 3 Differences between .NET Core and .Net Framework
- 4 Overview of ASP.NET Web API



Differences between .NET Core and .Net Framework

.NET Framework is the first implementation of .NET on Windows platform only.

.NET Core is the latest implementation of .NET which runs on Linux, Windows, and macOS.



Differences between .NET Core and .Net Framework

Source code is public but Microsoft doesn't accept third party contributions for it **.NET Framework**.

Open-source and Microsoft accepts third party contribution to **.NET Core**.



Differences between .NET Core and .Net Framework

.NET Framework has a very rich desktop top development framework for windows application which include Windows Forms and WPF.

.NET Core supports desktop frameworks like Windows Forms and WPF from version 3.0.



Differences between .NET Core and .Net Framework

A huge third-party packages library is also available for **.NET Framework**.

.NET Core also has support for a large number of third party packages as well but still, it doesn't compete with .NET Framework in this area.



Differences between .NET Core and .Net Framework

.NET Framework doesn't support the in-app deployment model.

.NET Core does support in-app deployment model.



Chapter 01

- 1 Overview of ASP.NET Core
- 2 Why we choose ASP.NET Core?
- 3 Differences between .NET Core and .Net Framework
- 4 Overview of ASP.NET Web API



Overview of ASP.NET Web API

The **ASP.NET Web API** is an extensible framework for building HTTP based services that can be accessed in different applications on different platforms such as web, windows, mobile etc.



Overview of ASP.NET Web API

ASP.NET Web API Characteristics

- ASP.NET Web API is an ideal platform for building RESTful services.
- ASP.NET Web API is built on top of ASP.NET and supports ASP.NET request/response pipeline.
- ASP.NET Web API maps HTTP verbs to method names.



Overview of ASP.NET Web API

- ASP.NET Web API supports different formats of response data. Built-in support for JSON, XML, BSON format.
- ASP.NET Web API can be hosted in IIS, Self-hosted or other web server that supports .NET 4.0+.



References

- [1]. <https://www.educative.io/blog/crud-operations>
- [2]. <https://www.codecademy.com/articles/what-is-crud>
- [3]. <https://www.restapitutorial.com/lessons/httpmethods.html>
- [4]. https://www.w3schools.com/sql/sql_stored_procedures.asp
- [5]. <https://www.pragimtech.com/blog/blazor/rest-api-repository-pattern/#:~:text=sql%20server%20database-,What%20is%20Repository%20Pattern,is%20in%20the%20respective%20repository>
- [6]. <https://code-maze.com/filtering-aspnet-core-webapi/>
- [7]. <https://www.infoworld.com/article/3562271/how-to-use-data-transfer-objects-in-aspnet-core-31.html>

