ASP.Net Core

ASP.Net Core is a cross-platform, high-performance, open-source framework for building modern, cloud-enabled web applications and services.

Features

- Cross-platform
  Asp.Net core apps can be hosted on Windows, LINUX and Mac.

- Can be hosted on different serves
  Supports Kestrel, IIS, Nginx, Docker, Apache

- Open-source

- Cloud-enabled
  Out-of-box support for Microsoft Azure

Modules

- ASP.Net core MVC
  For creating medium to complex web applications

- ASP.Net core WebAPI
  For creating RESTful services for all types of client applications

- ASP.Net core Razor Pages
  For creating simple & page-focused web applications

- ASP.Net core Blazor
  For creating web applications with C# code both on client-side and server-side

ASP.Net Web Forms vs ASP.Net MVC vs ASP.Net Core

ASP.Net Web Forms:

- 2002
- Performance issues due to server events and view-state.
- Windows-only
- Not cloud-friendly
- Not open-source
- Event-driven development model

ASP.Net MVC:

- 2009
- Performance issues due to dependencies with ASP.Net (.net framework)
- Windows-only
- Slightly cloud-friendly
- Open source
- Model-View-Controller (MVC) pattern

ASP.Net Core:

- 2016
- Faster performance

- Cross-platform
- Open source
- Cloud-friendly
- Model-View-Controller (MVC) pattern

What is ASP.Net Core?

ASP.Net Core is a free, open-source, and cross-platform framework for building cloud based applications, such as web apps, IoT apps, and mobile backends. It is designed to run on the cloud as well as on-premises.

ASP.Net Core is not an upgraded version of ASP.Net. ASP.Net Core is completely rewriting that work with the .Net Core framework. It is much faster, configurable, modular, scalable, extensible, and has cross-platform support. It is best suitable for developing cloud-based such as web applications, mobile applications and IoT applications.

What are the features of ASP.Net Core?

Following are the core features that are provided by the ASP.Net Core

- Built-in support for Dependency Injection.
- Built-in support for logging framework and it can be extensible.
- Introduced a new, fast and cross-platform web server – Kestrel. So, a web application can run without IIS, Apache, and Nginx.
- Multiple hosting platforms are supported.
- It supports modularity, so the developer need to include the module required by the application.
- Command-line support to creating, building and running of the application.
- There is no web.config file. We can store the custom configuration into an appsettings.json file.
- It has good support for asynchronous programming

What are the advantages of ASP.Net Core over ASP.Net (.Net Framework)?

These are the following advantages of ASP.Net Core over ASP.Net:

- It offers faster performance due to its minimalistic design.
- It is cross-platform, so it can be run on Windows, Linux and Mac.
- It is open-source.
- There is no dependency on framework installation because all the required dependencies are shipped with our application to the production server.
- Multiple deployment platform available with ASP.Net Core.

What is ASP.Net Core meta package?

The ASP.Net Core shared framework (Microsoft.AspNetCore.App) contains assemblies that are developed and supported by Microsoft.

Microsoft.AspNetCore.App is installed when the .Net Core 3.0 or later SDK is installed. The shared framework is set of assemblies (.dll files) that are installed on the machine and includes a runtime component and a targeting pack.

When do you choose classic ASP.Net MVC over ASP.Net Core?

Though ASP.Net Core is a better choice in almost all the aspects, you don't have to switch to ASP.Net Core if you are maintaining a legacy ASP.Net application that you are happy with and that is no longer actively developed.

ASP.NET MVC is a better choice if you:

- Don’t need cross-platform support for your Web app.
- The existing team is already working on an existing app and extending its functionality.
- The existing developers need a learning curve to upgrade themselves to ASP.Net Core.

What is a web application framework, and what are its benefits?

Learning to build modern web application can be daunting. Most of the applications have a standard set of functionalities such as:

- Build a dynamic response that corresponds to an HTTP request.
- Allow users to login to the application and manage their data.
- Store the data in the database.
- Handle database connections and transactions.
- Route URLs to appropriate methods.
- Supporting sessions, cookies, and user authorization.
- Format output (e.g: HTML, JSON, XML)
- Improve security.

Frameworks help developers to write, maintain, and scale applications. They provide tools and libraries that simplify the above recurring tasks, eliminating a lot of unnecessary complexity.