ASP.NET Core

1. **What is ASP.NET Core?**

* ASP.NET Core is a **cross platform**, **high-performance**, and open-source framework for building modern**, cloud-based, internet-connected applications**.
* ASP.NET core is a redesign of ASP.NET 4.x.

1. **Benefits and Features?**

* **Cross-Platform**
  + ASP.NET Core applications can be developed and run across different platforms like **Windows, mac-OS, Linux**.
  + ASP.NET Core application can be hosted on- **IIS, Apache, Docker, Self-host in your own process**.
* **One Unified Programming model for MVC and Web API**

Both the MVC Controller class and the ASP.NET Web API Controller class inherit from the same Controller base class and returns **IActionResult.**

|  |
| --- |
|  |

* **Dependency Injection**
* **Testability**

With build-in supports for dependency injection and unified programming model for creating both web-application and webAPI, unit testing of ASP.NET core application is easy.

* **Open-Source**
* **Modular**
  + ASP.NET Core provides modularity with **MiddleWare** Components.
  + **Both the request and response pipelines are composed using the middleware components.**
  + Rich set of built-in middleware components.
  + Custom Middleware components can also be created.

1. **Main Methods in ASP.NET Core**

ASP.NET Core application initially starts as a console application and main method with in the program class is the entry point with in this application. So in the run-time when we execute this application it looks for the main method and this is where ASP.NET core application kicks off.

**This main method configures ASP.NET Core and starts it and at that point it becomes an ASP.NET Core web application.**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Threading.Tasks;  using Microsoft.AspNetCore;  using Microsoft.AspNetCore.Hosting;  using Microsoft.Extensions.Configuration;  using Microsoft.Extensions.Logging;  namespace EmployeeManagement  {  public class Program  {  public static void Main(string[] args)  {  CreateWebHostBuilder(args).Build().Run();  }  //CreateDefaultBuilder is the static method that create the webhost  // with certain preconfigure default  // UseStartup is the extension method which will call the startup class  public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>  WebHost.CreateDefaultBuilder(args)  .UseStartup<Startup>();  }  } |

1. **An ASP.NET Core application can be hosted:-**

* InProgress or
* OutOfProcess

1. **Some of the tasks that CreateDefaultBuilder() performs**

* Setting up the webserver.
* Loading the host and application configuration from various configuration sources.
* Configuring logging.

1. **ASP.NET Core InProcess Hosting**

* To Configure InProcess Hosting

|  |
| --- |
| <AspNetCoreHostingModel>InProcess</AspNetCoreHostingModel> |

* CreateDefaultBuilder() method calls UseIIS() method and host the app inside of the IIS worker process (w3wp.exe or iisexpress.exe).
* InProcess hosting delivers significantly higher request throughput that OutOfProcess hosting.
* To get the process name executing the app.

|  |
| --- |
| System.Diagnostics.Process.GetCurrentProcess().ProcessName |

**With Out of process hosting**

* 2 Web Servers- Internal and External Webserver
* The internal web server is Kestrel.
* The external web server can be IIS, Nginx or Apache

**Whereas In Process Hosting there is only one web server i.e. IIS. So with In Process hosting we don’t have the penalty of proxy request between Internal and External Webserver.**

**For a Performance standpoint, InProcess hosting is better that OutOfProcess hosting.**

|  |
| --- |
|  |

1. **What is Kestrel?**

* Cross-Platform webserver for ASP.NET Core.
* Kestrel can be used by itself as an edge server.
* The process used to host the app is dotnet.exe.

**We can use Dotnet CLI to create, run, build etc for Dotnet Project.**

1. **ASP.NET Core Out Of Process Hosting**

**With Out of process hosting**

* 2 Web Servers- Internal and External Webserver
* The internal web server is Kestrel.
* The external web server can be IIS, Nginx or Apache (it is not mandatory to use)

|  |
| --- |
|  |

* To enable OutOfProcess hosting in the application just remove the below line or change InProcess OutOfProcess in the project file.

|  |
| --- |
| <AspNetCoreHostingModel>InProcess</AspNetCoreHostingModel> |

|  |
| --- |
|  |

1. **Difference between InProcess and OutOfProcess Hosting**

|  |
| --- |
|  |

**So if we use In Process Hosting Kestrel Server is not used.**