

ASP.NET

Anatomy of .NET

.NET Base Classes

Microsoft Intermediate Language

CLR

Client Server model

IIS Web Server

Namespaces

ASP.NET: How the ASP.NET works

- Basics of ASP.NET

- Creating and Deploying the ASP.NET applications

- Concept of Code behind

- Use of Web Applications using ASP.NET

- Difference between Windows and Web Applications.

- ASP.NET directives.

Anatomy Of .Net

.NET and .NET Framework

[.NET is a developer platform](#) made up of [tools](#), [programming languages](#), and libraries for building many different types of applications.

There are various implementations of .NET. Each implementation allows .NET code to execute in different places—Linux, macOS, Windows, iOS, Android, and many more.

1. **.NET Framework** is the original implementation of .NET. It supports running websites, services, desktop apps, and more on Windows.
2. **.NET** is a cross-platform implementation for running websites, services, and console apps on Windows, Linux, and macOS. [.NET is open source](#) on GitHub. .NET was previously called .NET Core.
3. [Xamarin/Mono](#) is a .NET implementation for running apps on all the major mobile operating systems, including iOS and Android.

.NET Framework

.NET is a framework to develop software applications. It is designed and developed by Microsoft and the first beta version released in 2000.

It is used to develop applications for web, Windows, phone. Moreover, it provides a broad range of functionalities and support.

This framework contains a large number of class libraries known as Framework Class Library (FCL). The software programs written in .NET are executed in the execution environment, which is called CLR (Common Language Runtime). These are the core and essential parts of the .NET framework.

This framework provides various services like memory management, networking, security, memory management, and type-safety.

Following is the .NET framework Stack that shows the modules and components of the Framework.

The .NET Framework is composed of four main components:

- 1.Common Language Runtime (CLR)
- 2.Framework Class Library (FCL),
- 3.Core Languages (WinForms, ASP.NET, and ADO.NET), and
- 4.Other Modules (WCF, WPF, WF, Card Space, LINQ, Entity Framework, Parallel LINQ, Task Parallel Library, etc.)

CLR (Common Language Runtime)

It is a program execution engine that loads and executes the program. It converts the program into native code. It acts as an interface between the framework and operating system. It does exception handling, memory management, and garbage collection. Moreover, it provides security, type-safety, interoperability, and portability. A list of CLR components are given below:

FCL (Framework Class Library)

It is a standard library that is a collection of thousands of classes and used to build an application. The BCL (Base Class Library) is the core of the FCL and provides basic functionalities.

WinForms

Windows Forms is a smart client technology for the .NET Framework, a set of managed libraries that simplify common application tasks such as reading and writing to the file system.

ASP.NET

ASP.NET is a web framework designed and developed by Microsoft. It is used to develop websites, web applications, and web services. It provides a fantastic integration of HTML, CSS, and JavaScript. It was first released in January 2002.

ADO.NET

ADO.NET is a module of .Net Framework, which is used to establish a connection between application and data sources. Data sources can be such as SQL Server and XML. ADO .NET consists of classes that can be used to connect, retrieve, insert, and delete data.

WPF (Windows Presentation Foundation)

Windows Presentation Foundation (WPF) is a graphical subsystem by Microsoft for rendering user interfaces in Windows-based applications. WPF, previously known as "Avalon", was initially released as part of .NET Framework 3.0 in 2006. WPF uses DirectX.

WCF (Windows Communication Foundation)

It is a framework for building service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another.

WF (Workflow Foundation)

Windows Workflow Foundation (WF) is a Microsoft technology that provides an API, an in-process workflow engine, and a rehostable designer to implement long-running processes as workflows within .NET applications.

LINQ (Language Integrated Query)

It is a query language, introduced in .NET 3.5 framework. It is used to make the query for data sources with C# or Visual Basics programming languages.

Entity Framework

It is an ORM based open source framework which is used to work with a database using .NET objects. It eliminates a lot of developers effort to handle the database. It is Microsoft's recommended technology to deal with the database.

Parallel LINQ

Parallel LINQ or PLINQ is a parallel implementation of LINQ to objects. It combines the simplicity and readability of LINQ and provides the power of parallel programming.

It can improve and provide fast speed to execute the LINQ query by using all available computer capabilities.

Apart from the above features and libraries, .NET includes other APIs and Model to improve and enhance the .NET framework.

.NET Base Classes

.NET Framework Class Library is the collection of classes, namespaces, interfaces and value types that are used for .NET applications.

It contains thousands of classes that supports the following functions.

- Base and user-defined data types
- Support for exceptions handling
- input/output and stream operations
- Communications with the underlying system
- Access to data
- Ability to create Windows-based GUI applications
- Ability to create web-client and server applications
- Support for creating web services

.NET Framework Base Class Library

.NET Base Class Library is the sub part of the Framework that provides library support to Common Language Runtime to work properly. It includes the System namespace and core types of the .NET framework.

.NET Common Language Runtime (CLR)

.NET CLR is a run-time environment that manages and executes the code written in any .NET programming language.

It converts code into native code which further can be executed by the CPU.

.NET CLR Functions

Following are the functions of the CLR.

It converts the program into native code.

Handles Exceptions

Provides type-safety

Memory management

Provides security

Improved performance

Language independent

Platform independent

Garbage collection

Provides language features such as inheritance, interfaces, and overloading for object-oriented programmings.

.NET CLR Structure:-

Following is the component structure of Common Language Runtime.

Base Class Library Support

It is a class library that provides support of classes to the .NET application.

Thread Support

It manages the parallel execution of the multi-threaded application.

COM Marshaler

It provides communication between the COM objects and the application.

Type Checker

It checks types used in the application and verifies that they match to the standards provided by the CLR.

Code Manager

It manages code at execution run-time.

Garbage Collector

It releases the unused memory and allocates it to a new application.

Exception Handler

It handles the exception at runtime to avoid application failure.

ClassLoader

It is used to load all classes at run time.

Microsoft Intermediate Language

The Microsoft Intermediate Language (MSIL), also known as the Common Intermediate Language (CIL) is a set of instructions that are platform independent and are generated by the language-specific compiler from the source code. The MSIL is platform independent and consequently, it can be executed on any of the Common Language Infrastructure supported environments such as the Windows *.NET* runtime.

The MSIL is converted into a particular computer environment specific machine code by the [JIT compiler](#). This is done before the MSIL can be executed. Also, the MSIL is converted into the machine code on a requirement basis i.e. the JIT compiler compiles the MSIL as required rather than the whole of it.

Client-Server Model

The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources. Examples of Client-Server Model are Email, World Wide Web, etc.

1. How the Client-Server Model works ?

In this article we are going to take a dive into the **Client-Server** model and have a look at how the **Internet** works via, web browsers. This article will help us in having a solid foundation of the WEB and help in working with WEB technologies with ease.

- **Client:** When we talk the word **Client**, it mean to talk of a person or an organization using a particular service. Similarly in the digital world a **Client** is a computer (**Host**) i.e. capable of receiving information or using a particular service from the service providers (**Servers**).
- **Servers:** Similarly, when we talk the word **Servers**, It mean a person or medium that serves something. Similarly in this digital world a **Server** is a remote computer which provides information (data) or access to particular services.

So, its basically the **Client** requesting something and the **Server** serving it as long as its present in the database.

Advantages of Client-Server model:

- Centralized system with all data in a single place.
- Cost efficient requires less maintenance cost and Data recovery is possible.

- The capacity of the Client and Servers can be changed separately.

Disadvantages of Client-Server model:

- Clients are prone to viruses, Trojans and worms if present in the Server or uploaded into the Server.
- Server are prone to Denial of Service (DOS) attacks.
- Data packets may be spoofed or modified during transmission.
- Phishing or capturing login credentials or other useful information of the user are common and MITM(Man in the Middle) attacks are common.

Internet Information Services Web Server

The term "IIS" stands for Internet Information Services, which is a general-purpose webserver that runs on the Windows operating system. The IIS accepts and responds to the client's computer requests and enables them to share and deliver information across the [LAN](#) (or Local Area Network) such as a corporate intranet and the WAN (or Wide Area Network) the internet.

It hosts the application, websites, and other standard services needed by users and allows developers to make websites, applications and virtual directories to share with their users. A web server provides the users with information in several different forms, such as File exchanges as a download, uploads, Images files, HTML pages, and text documents. The webserver are commonly used as a portal for sophisticated and highly interactive websites, applications that tie middleware and back-end applications together to make enterprise-grade-systems.

For example, [AWS](#) enables media services such as Netflix to provide real-time streaming content. Amazon web services also enable public cloud administration all through the web servers. Generally, the IIS is also compared with the Apache, which is also a kind of web server that is freely available for everyone.

We can simply say that both work the same except that the Apache web server can be used almost on any operating system such as Windows, [Linux](#), and Mac, while the IIS is only available for Windows. However, the IIS integrates with Microsoft's other products, such as the [.NET Framework](#), the [ASP](#) scripting language. The IIS also has its own helpdesk to manage and solve issues while, on the other hand, the Apache web server's supports almost come from the user community. Additionally, the IIS has the security features, which makes it a more secure and efficient option than the Apache.

Namespaces

Namespaces are used to organize the classes. It helps to control the scope of methods and classes in larger [.Net](#) programming projects. In simpler words you can say that it provides a way to keep one set of names (like class names) different from other sets of names.

The biggest advantage of using namespace is that the class names which are declared in one namespace will not clash with the same class names declared in another namespace. It is also referred to as **named group of classes** having common features. The members of a namespace can be **namespaces**, [interfaces](#), [structures](#), and [delegates](#).

Defining a Namespace

To define a namespace in C#, we will use the **namespace** keyword followed by the name of the namespace and curly braces containing the body of the namespace as follows:

Syntax:

```
namespace name_of_namespace {  
  
    // Namespace (Nested Namespaces)  
    // Classes  
    // Interfaces  
    // Structures  
    // Delegates  
  
}
```

ASP.NET (How the Asp.Net Works?)

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.

ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.

ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.