**ASP.NET WEB API**

**What is ASP.NET Web API?**

The term API stands for “Application Programming Interface”. ASP.NET Web API is a framework, provided by Microsoft, which makes it easy to build Web APIs, i.e. HTTP based services. The ASP.NET Web API is an ideal platform for building Restful services on top of the .NET Framework. These Web API services can be consumed by a variety of clients such as

1. Browsers
2. Mobile applications
3. Desktop applications
4. IOTs, etc.

The most important thing to keep in mind is that we can develop both Restful and Non-Restful Web Services using the ASP.NET Web API framework. But mostly this framework is used to create RESTful services. In short, this framework does not provide any specific architectural style for creating the services. In this article series, we are going to discuss creating RESTful services from scratch using the Web API framework.

**What is Rest?**

REST stands for **(Representational State Transfer)** is an architectural style for building web services that allows communication between client and server over HTTP. It treats each service as a resource, which can be accessed and manipulated using standard HTTP methods like GET, POST, PUT, PATCH, and DELETE.

**Types of APIs in General**

In the context of software development, APIs are broadly categorized as:

| **Type** | **Description** |
| --- | --- |
| Open APIs (Public) | Available to external developers and users (e.g., Google Maps API). |
| Internal APIs (Private) | Used only within an organization. |
| Partner APIs | Shared with specific partners and require permission or contracts. |
| Composite APIs | Combine multiple services or data sources into a single API call. |

**HTTP Methods**

GET: This method is used to retrieve the data from the Database.

POST: This method is used to make a new entry in the Database.

PUT: This method is used to update all the properties of the current resource in the Database.

DELETE: This method is used to delete the data from the Database.

**What are the advantages of using ASP.NET Web API?**

Using [ASP.NET Web API](https://dotnettutorials.net/course/asp-net-web-api/) has a number of advantages, but the core advantages are as follows:

1. It supports all the HTTP features. That means you can use all the built-in HTTP Heapers such as Content-Type, Accept, Authorization, etc. and HTTP Status codes such as 500, 200, 404, etc., and HTTP verbs such as GET, POST, PUT, PATCH, and DELETE to perform CRUD operations
2. It supports Attribute Routing which is good for SEO as well as user-friendly URLs.
3. It supports content negotiation i.e. as per the client request, the server sends the response in that format (if possible). The Response is generated in JSON or XML format using [MediaTypeFormatter](https://dotnettutorials.net/lesson/media-type-formatters-web-api/).
4. It has the ability to be hosted in IIS as well as self-host outside of IIS
5. Supports Model Binding and Validation.

**How Web API Works in C#**

When you create a Web API in C# (especially using ASP.NET Core), you generally:

1. Create a **Model** – Represents the data structure.
2. Create a **Controller** – Contains methods that handle HTTP requests.
3. Define **Routes** – To map HTTP requests to the controller methods.
4. Optionally connect to a **Database** using Entity Framework.

**Tools and Technologies**

* **Visual Studio / VS Code** – IDEs for developing C# Web APIs
* **Postman** – For testing APIs
* **Swagger (OpenAPI)** – For documenting and trying API endpoints
* **Entity Framework Core** – For database integration
* **.NET CLI** – Command-line tool to build and run apps

**Conclusion**

Web API in C# is an essential technology for building modern web and mobile applications. It allows seamless communication between clients and servers using RESTful principles. With tools like ASP.NET Core, building secure, efficient, and scalable APIs in C# is both straightforward and powerful.