BRIEF OVERVIEW OF WEB API

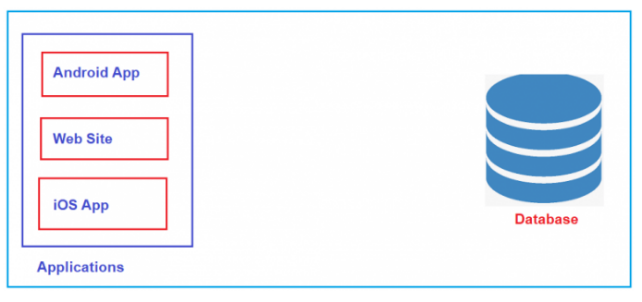
A **Web API in ASP.NET** is a framework for building HTTP services that can be accessed by various clients like browsers, mobile devices, or desktop applications. It is widely used to create RESTful services and is built on top of the **ASP.NET Core** or the older **ASP.NET Web API** (non-Core) platform.

**Key Features of ASP.NET Web API**

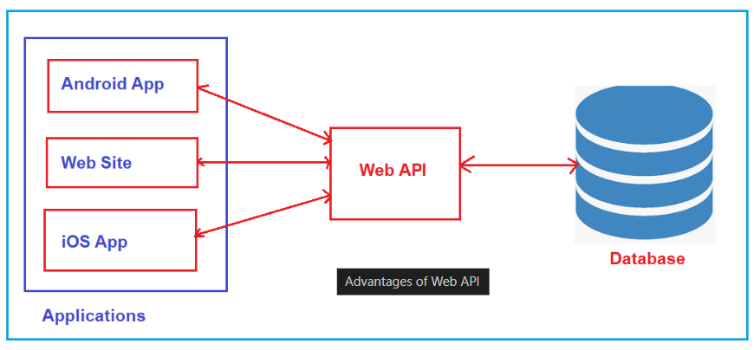
* **RESTful Services**  
  Supports HTTP methods like GET, POST, PUT, DELETE for CRUD operations.
* **HTTP-Based Communication**  
  Uses standard HTTP protocols and verbs for client-server interaction.
* **Content Negotiation**  
  Automatically returns data in the requested format (JSON, XML, etc.).
* **Model Binding & Validation**  
  Binds incoming request data to C# models and validates them automatically.
* **Routing Support**  
  Supports both convention-based and attribute-based routing.
* **Authentication & Authorization**  
  Easily integrates with JWT, OAuth, or ASP.NET Identity for secure APIs.
* **Swagger/OpenAPI Integration**  
  Provides built-in support for API documentation and testing.
* **Dependency Injection (DI)**  
  Built-in DI container for better modularity and testing.
* **Middleware Support**  
  Can add custom middleware for logging, error handling, etc.
* **Cross-Platform (ASP.NET Core)**  
  Runs on Windows, Linux, and macOS when using ASP.NET Core.

**Need For Web API**

As you can see in the image below, we have three applications on the left and the database on the right.



We aim to establish communication between all these three applications and the database to manage the data. So, what will we do? We will add a new Web API Project between these three front-end applications and the back-end database. This Web API Project will interact with the database, and all three applications will only interact with the Web API Project, as shown in the image below.



So, the Website, Android, and iOS applications do not have direct access to the database. They only need to communicate with the Web API Project, and it is the Web API project’s responsibility to interact with the database. The entire business logic will be written in the Web API project only, so we need Web API for our project. So, Web API acts as a mediator between the Front-End and Back-End.

**Advantages Of Web API**

* Platform Independence: Web APIs can be consumed by any client that understands standard web protocols like HTTP, regardless of the underlying programming language.
* Low Bandwidth: APIs typically exchange data using JSON or XML, which is much lighter than full HTML views, reducing the amount of data transmitted.
* Reusability: APIs promote code reuse. Once an API is defined and implemented, it can be used by multiple clients or applications without requiring significant changes. That means we write the logic in one place, i.e., in our Web API project, and all applications will use the same logic.
* Security: APIs can be secured using authentication and authorization mechanisms, ensuring that only Authenticated and authorized clients can access sensitive data or perform specific actions.
* Extend Application Functionality: Suppose, first, we develop the website. Then, we can extend and develop an Android App. Again, in the future, if you want to add another type of application, such as iOS, we don’t have to write any logic.

**Testing the API**

We can Test the API by using:

* **Swagger UI** (included by default in .NET Core Web API template)
* **Postman** or **cURL**
* **Your frontend app (React, Angular, etc.)**