**What is the difference between Web API and WCF?**

**WCF**(Windows Communication Foundation): It is a framework used for developing SOAP (Service-oriented applications). This framework is used for developing, configuring, and deploying, or implementing network-distributed services.   
  
**Web API**: Web API is generally considered as a service that basically provides us information or data from the server.

| **Web API** | **WCF** |
| --- | --- |
| It is used to develop both SOAP-based services and RESTful services. | It is used to deploy only SOAP-based services. |
| It supports various MVC features such as routing, model binding, etc. | It does not support any MVC features. |
| It only supports HTTP protocol. | It supports various protocols such as HTTP, UDP, custom transport. |
| It is considered best for developing RESTFUL services. | It supports only limited RESTFUL services. |
| It is good when one wants to expose an expensive range of clients such as iPhones, browsers, mobile phones, tablets, etc. | It is good for creating services that uses expedite transport channels such as TCP, UDP, Named pipes, etc. |
| It offers support for UTF-8 encoding format. | It offers TEXT, Binary encoding support, MTOM (Message Transmission Optimization Mechanism), etc. |
|  |  |

**What is Web Services?**

Web services provide a common platform that allows multiple applications built on various programing languages to have the ability to communicate with each other.

**There are two types of we services:**

1.SOAP(Simple Object Access Protocal) Web Services

2. RESTful Web Services.

**What is REST?**

REST stands for Representational State Transfer. It is a set of rules that developers follows when they create their API’s. In simple word rest is an architecture to create restful services.

**What is Restful API?**

The API which is built on REST architecture is called as RESTful API.

**What is Versioning in API?**

While working on an existing application or creating a new one, we may create multiple APIs that may be consumed by many clients. When the business has started to grow and expand, new requirements arise. Due to this, we may need to provide more functionality in the existing APIs. However, existing Web API can be consumed by many clients so how to implement the new feature without impacting the existing consumers? We can solve this problem by versioning our API.

We can do versioning of Web API using the following methods:

1. Query string
2. URL
3. HTTP header

**What are the 4 types of requests in API?**

GET : Get object

PUT : Modify Object

DELETE: Delete Object

POST : Create Object

**How to make Web API more secure?**

**Data Encryption through TLS**

For security concerns, it is recommended that the Web APIs should use the HTTPS (HTTP secure) endpoints to ensure that the data communication is encrypted using TLS/SSL (Transport Layer Security).

**Access Control**

Some Web APIs are used internally and only available to authenticated users like the *Payment service* API. In RESTful Web APIs

Below are some authentication methods used in RESTful Web APIs:

HTTP Basic Authentication – Checking credentials

JSON Web Tokens (JWT)

OAuth

**Throttling and Quotas**

Throttling limits and quotas prevents the system from different cyber security attacks and reduces the overburden of processing so that the system operates effectively.

Throttling prevents the system from overloaded requests. You can set the limit on the number of requests per second to protect the backend data bandwidth according to the server’s capability.

Throttling limits also helps in preventing attacks from flooding the system with a large number of requests – also known as a *DDOS (Distributed Denial of Service)* attack.

**Sensitive Information in the API Communication**

API often makes use of confidential data such as usernames, passwords, session tokens, or API keys. If they are directly placed into the URL then these details might get saved to server logs and from there, intruders can easily access them.

So it is highly recommended that any credential or sensitive information should be sent in the HTTP request headers (for *GET* requests) or the request body (for *POST* or *PUT* requests).

**Using Hashed Passwords**

Organizations should ensure that the passwords they use in an API should be hashed. There are various mechanisms you can use to secure the passwords, including: *MD5*, *SHA256*, *SHA512*, *PBKDF2*, etc.

**What is CORS issue?**

CORS(Cross Origin Resource Sharing) when we request data from two different domains then we will face this issue.

To resolve this issue we have to add middleware like this:

res.header(“Access-Control-Allow-Origin”, “\*”);