

What is Web API?

A **Web API (Application Programming Interface)** is a set of rules and protocols that allow two software applications to communicate over the internet. It defines the methods and data formats that applications can use to interact with external systems, databases, or services.

Key Characteristics of a Web API:

1. Stateless Communication:

- Each request from a client to the server is treated as independent.
- No client information is stored between requests.

2. Standard Protocols:

- Web APIs typically use **HTTP/HTTPS** as the communication protocol.

3. Data Formats:

- Common formats include:
 - **JSON (JavaScript Object Notation)** – Lightweight and easy to parse.
 - **XML (Extensible Markup Language)** – Used in older APIs but still supported.

4. RESTful Design:

- Most modern Web APIs follow the **REST (Representational State Transfer)** architecture.
- It uses standard HTTP methods such as:
 - **GET** – Retrieve data
 - **POST** – Create new data
 - **PUT/PATCH** – Update existing data

- DELETE – Remove data
-

How Does a Web API Work?

✓ Step 1: Client Sends a Request

- A client (browser, mobile app, etc.) sends an HTTP request to the API endpoint.

```
GET /api/books/ HTTP/1.1
Host: example.com
```

✓ Step 2: Server Processes the Request

- The API processes the request, retrieves the necessary data, and performs the requested action.

✓ Step 3: Server Sends a Response

- The API sends the requested data back to the client.

```
{
  "id": 1,
  "title": "Python for Beginners",
  "author": "Sanket Chauhan"
}
```

Types of Web APIs:

1. REST API (Representational State Transfer)

- Stateless and resource-oriented.

- Uses standard HTTP methods.

2. SOAP API (Simple Object Access Protocol)

- XML-based protocol for exchanging structured information.
- Strict standards, commonly used in enterprise systems.

3. GraphQL API

- Allows clients to request only the data they need.
- Single endpoint with flexible querying.

4. gRPC API (Google Remote Procedure Call)

- High-performance API using Protocol Buffers.
 - Ideal for microservices and real-time applications.
-



Example of a RESTful Web API



1. GET Request (Retrieve Data)

GET /api/books/1/ HTTP/1.1

Host: example.com

Response:

```
{  
  "id": 1,  
  "title": "Django REST API",  
  "author": "Sanket Chauhan"  
}
```



2. POST Request (Create Data)

POST /api/books/ HTTP/1.1

Host: example.com

Content-Type: application/json

```
{  
  "title": "Python for Data Science",  
  "author": "John Doe"  
}
```

Response:

```
{  
  "id": 2,  
  "title": "Python for Data Science",  
  "author": "John Doe"  
}
```

3. PUT Request (Update Data)

PUT /api/books/2/ HTTP/1.1
Host: example.com
Content-Type: application/json

```
{  
  "title": "Advanced Python",  
  "author": "John Doe"  
}
```

4. DELETE Request (Remove Data)

DELETE /api/books/2/ HTTP/1.1
Host: example.com

Response:

```
{  
  "message": "Book deleted successfully!"  
}
```

Advantages of Web API

- ✓ **Platform Independence:** Can be accessed from any client or device.
 - ✓ **Scalability:** APIs allow modular application development.
 - ✓ **Security:** Can use authentication protocols like OAuth, JWT, and API keys.
 - ✓ **Integration:** Allows seamless integration between different platforms and services.
-

Use Cases of Web APIs

- Social media integrations (Facebook, Twitter, Instagram APIs).
 - Payment gateways (PayPal, Stripe APIs).
 - Weather services (OpenWeather API).
 - IoT device communication.
 - E-commerce product catalog management.
-

Interview Question:

What is the difference between REST API and SOAP API?
How does JWT authentication enhance Web API security?

Happy Learning! 🎉🚀