# **API**

An **Application Program Interface (API)** is a set of rules and protocols that allows two software applications to communicate with each other.

## **Types of APIs**

1. **Open API (Public API)**

Available for everyone to use

Companies provide them for integration with their services

1. **Private API (Internal API)**

Used only inside an organization

Not available to the public

1. **Partner API**

Shared only with trusted business partners

Protected by authentication (API keys or tokens)

1. **Composite API**

Combines multiple APIs into a single call

Used to improve performance and reduce requests

**Types Based on Architecture**

* **REST API**

Most commonly used

Uses HTTP methods like GET, POST, PUT, DELETE

Data format: usually JSON

* **SOAP API**

Older, more strict

Uses XML format

High security (used in banking systems)

* **GraphQL API**

Client can ask for specific data only

Faster and more flexible than REST

* **gRPC API**

Uses Protocol Buffers

Very fast, used in microservices

## **HTTP Status codes**

HTTP status codes are 3-digit numbers that indicate the result of a client’s request to a server. They are grouped into **5 main**

1. **1xx – Informational**

These indicate that the request has been received and is still being processed.

* 100 Continue
* 101 Switching Protocols

1. **2xx – Success**

The request was successfully received, understood, and processed.

* **200 OK** **:** Successful request
* **201 Created** **:** New resource created (e.g., after POST)
* **204 No Content** **:** Success but no response body

1. **3xx – Redirection**

The client is redirected to another URL.

* 301 Moved Permanently
* 302 Found (Temporary redirect)
* 304 Not Modified (Use cached version)

1. **4xx – Client Errors**

Something is wrong with the request made by the client.

* **400 Bad Request :** Invalid input
* **401 Unauthorized** **:** Authentication required
* **403 Forbidden** **:** Access denied
* **404 Not Found** **:** URL/resource doesn't exist
* **405 Method Not Allowed** **:** Wrong HTTP method used

1. **5xx – Server Errors**

The server failed to process a valid request.

* 500 Internal Server Error
* 502 Bad Gateway
* 503 Service Unavailable
* 504 Gateway Timeout

## **Response Formats**

When a server responds to a client’s request the **response can come in different formats**. These are the most common:

1. **JSON (JavaScript Object Notation)**

Most commonly used in APIs (modern standard)

Lightweight, easy to read, supports key-value pairs

1. **XML (eXtensible Markup Language)**

Older format, still used in SOAP APIs

More verbose than JSON

1. **HTML**

Used when a server sends full web pages

1. **Plain Text (Text/Plain)**

Simple text messages

1. **CSV (Comma-Separated Values)**

Often used for spreadsheets or bulk data export/import

1. **YAML (YAML Ain’t Markup Language)**

Used in configuration files and some APIs

1. **Binary / File Responses**

Used when downloading files like:

* PDFs
* Images
* ZIP files
* Excel sheets

1. **Form Data / Multipart**

Used mainly for file uploads or form submissions

## **API Versioning**

API versioning helps you update or improve an API **without breaking existing users' applications**.

**Why Versioning is Needed**

* To add new features
* To fix bugs or improve performance
* To make changes safely without affecting existing clients

**Common Ways to Version an API:**

* In the URL (Most common)
* In Request Headers
* Query Parameters
* Subdomain (Rare)

## **API Security**

To protect APIs from unauthorized access, data theft, and misuse, several security methods are used.

**Major API Security Methods:**

* **API Key**

A unique key is passed in headers or query params.

* **OAuth 2.0 (Most popular)**

Used by apps like Google, Facebook, GitHub.

It uses **access tokens** to grant secure access.

* **JWT (JSON Web Token)**

Token-based authentication, contains encoded user info.

Used in login systems, mobile apps, web apps

* **HTTPS (TLS/SSL)**

Encrypts data during transfer to prevent hacking.

* **Rate Limiting**

Prevents abuse by limiting number of requests.

* **IP Whitelisting / Restriction**

Allow requests only from trusted IPs.

* **Input Validation & Data Encryption**

Protects against attacks like:

1. SQL Injection
2. XSS
3. Man-in-the-middle attacks

## **CRUD operations**

CRUD refers to the **four basic operations** used to interact with databases or APIs:

1. **Create**
2. **Read**
3. **Update**
4. **Delete**

These map directly to HTTP methods in REST

| **CRUD Operation** | **HTTP Method** | **Description** | **Example Endpoint** |
| --- | --- | --- | --- |
| **Create** | POST | Add new data | POST /users |
| **Read** | GET | Retrieve data | GET /users or GET /users/5 |
| **Update** | PUT / PATCH | Modify existing data | PUT /users/5 |
| **Delete** | DELETE | Remove data | DELETE /users/5 |

## **Explore POSTMAN**

**Postman** is a popular tool used for testing, building, and managing APIs. It allows developers to send requests to APIs and view the responses easily without writing code.

**Why is Postman Used?**

* Test REST APIs (GET, POST, PUT, DELETE, etc.)
* Send data in JSON, form-data, or params
* Add authentication (API Keys, Tokens, OAuth)
* Automate API testing
* Save and organize API requests into collections
* View response status, headers, time, and data

**Key Features of Postman:**

* Send API Requests
* Supports Different Request Types
* Authentication
* Environment Variables
* Collections
* Testing & Automation

## **Optimization**

**Optimization** means improving a system, program, or API to make it work better without changing what it does.

**Goals of Optimization:**

* Reduce response time
* Lower resource usage (CPU, RAM, bandwidth)
* Handle more users (scalability)
* Reduce costs

## **Efficiency**

Efficiency is about using the available resources in the best possible way to get maximum output with minimum waste.

**An efficient system:**

* Uses less processing power
* Loads faster
* Makes fewer database/API calls
* Avoids unnecessary repetition

## **Requests lib in Python**

The **requests** library in Python is a simple and powerful module used to make HTTP requests (like GET, POST, PUT, DELETE) to web servers or APIs.

It’s one of the most commonly used libraries for interacting with REST APIs.

## **RBAC**

**RBAC (Role-Based Access Control)** is a security model where access to features or data is based on a user’s role.

Instead of assigning permissions to each user individually, you assign:

* Roles : Permissions
* Users : Roles

This makes access control easier and cleaner.