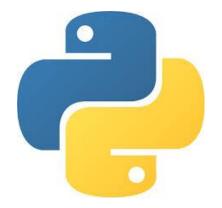
Python – Application Program Interface(API)





API

Definition:

API stands for Application Programming Interface. It is a set of rules and protocols
that allows two applications to communicate with each other.

Key Features:

- Intermediary: Acts as a bridge between different software applications.
- Reusability: Developers can use APIs to integrate existing functionality into their applications.
- Flexibility: APIs allow applications to exchange data securely and efficiently.

API Example

Example:

When you use an app to check the weather, the app communicates with a weather server's API to retrieve and display the weather data.

Work flow:

- You: Send a request (e.g., weather data).
- Your Application: Formats your request into an API call.
- API Server: Receives the API call, processes it, and fetches data.
- API Response: Sends back the requested data to your application.
- Your Application: Displays the result to you.

REST API

Definition:

- REST (Representational State Transfer) is a set of architectural principles used to design networked applications.
- A REST API uses HTTP requests to perform CRUD operations: Create, Read, Update, and Delete data.

Key Features:

- Stateless: Each request is independent and contains all the information needed to process it.
- Uniform Interface: Uses standard HTTP methods like GET, POST, PUT, and DELETE.
- Scalability: Designed to handle large numbers of client-server interactions efficiently.

REST API Example

Example:

- Request: GET https://api.openweathermap.org/data/2.5/weather?q=London&appid=your_api_key
- This REST API fetches weather data for London.

Breakdown:

- GET: This HTTP method tells the API that you want to retrieve data.
- https://api.openweathermap.org/data/2.5/weather: This is the endpoint or URL of the weather API q=London: This is a query parameter that tells the API you're interested in weather data for the city of London.
- appid=your_api_key: This is your API key, a unique identifier that lets the API know who is making the request.

Request

Definition:

A message sent by the client to the server asking for specific information or action.

Key Features:

- HTTP Method: Determines the action (e.g., GET, POST).
- URL/Endpoint: The resource location (e.g., /weather).
- Headers: Provide metadata (e.g., API keys, content types).
- Body: (Optional) Contains data for actions like POST or PUT.

Example:

```
GET https://api.example.com/users/123
Headers: { "Authorization": "Bearer token123" }
```

Response

Definition:

• The message sent by the server back to the client with the requested data or result of the action.

Key Features:

- Status Code: Indicates the result of the request (e.g., 200 OK, 404 Not Found).
- **Headers:** Provide metadata (e.g., content type, server info).
- Body: Contains the requested data or error details.

Example:

```
"id": 123,
"name": "John Doe",
"email": "john.doe@example.com"
}
```

Thank You!

