

# Day 7: External Libraries - APIs & Web Scraping

## 1. requests Library

The `requests` library is a Python module that allows your program to communicate with web servers over HTTP. It is widely used to send and receive data from the internet.

### Key Concepts:

- **GET Requests:** Used to retrieve information from a server. Parameters can be included in the URL to filter or customize the data.
- **POST Requests:** Used to send data to a server, often to create or update resources. This data is usually sent in the request body.
- **Headers:** Metadata sent along with the request, such as content type, authentication tokens, or user-agent information.
- **Parameters:** Key-value pairs that customize requests, often used in GET requests to filter results.

## 2. Working with APIs

APIs (Application Programming Interfaces) allow applications to communicate with each other. REST APIs are a popular type that follow specific rules using HTTP methods like GET, POST, PUT, and DELETE.

### Key Concepts:

- **Understanding REST APIs:** REST (Representational State Transfer) APIs provide structured endpoints for accessing or manipulating resources.
- **API Keys:** Many APIs require authentication using an API key, which is a unique identifier for the client.
- **Rate Limits:** APIs often restrict the number of requests per unit of time to prevent abuse and ensure fair use.

### **3. JSON Response Handling**

APIs typically return responses in JSON (JavaScript Object Notation), a lightweight data format. Handling JSON responses is crucial for extracting meaningful information.

#### **Key Concepts:**

- **Parsing API Responses:** Convert JSON responses into a usable format like Python dictionaries.
- **Extracting Data:** Access specific values from the JSON structure, often nested, to retrieve the needed information.
- **Error Handling:** Check for error codes and messages in the response to handle failed requests gracefully.

### **4. BeautifulSoup**

BeautifulSoup is a Python library for parsing HTML and XML documents. It is commonly used in web scraping to extract structured data from web pages.

#### **Key Concepts:**

- **HTML Parsing:** Convert raw HTML content into a navigable tree structure.
- **Finding Elements:** Search for specific tags, attributes, or text within the HTML.
- **Extracting Text:** Retrieve the visible content or values from elements for further processing.

### **5. Web Scraping Ethics**

Web scraping should be performed responsibly and legally. Respecting the website's rules and server load is important.

#### **Key Concepts:**

- **robots.txt:** A file that indicates which parts of a website can be accessed by bots or crawlers.
- **Rate Limiting:** Avoid sending too many requests in a short period to prevent overloading servers.
- **Legal Considerations:** Ensure compliance with terms of service and copyright laws to avoid legal issues.

## 6. Environment Variables

Storing sensitive information like API keys directly in code is risky. Environment variables allow secure storage outside the code.

### Key Concepts:

- **Security:** Keep API keys and credentials safe from accidental exposure.
- **python-dotenv:** A popular library to load environment variables from a .env file into your Python application.
- **Best Practices:** Never commit .env files to public repositories; instead, use them locally or in secured deployment environments.