Step8Up Academy

HTTP AND 3RD PARTY APIS

FULL STACK SKILLS BOOTCAMP

HTTP AND 3RD PARTY APIS

- **Lesson Overview:**
- In this lesson, we will be introduced to:

1. The basics of HTTP
2. How to make HTTP requests
3. Using APIs with tools like Postman
4. Fetching and using JSON data in JavaScript

THE FUNDAMENTALS OF HTTP

- **What is HTTP**:
  - HyperText Transfer Protocol (HTTP) is the foundation of communication on the web.
  - It enables browsers and servers to exchange data.
- **Key Concepts**:
  - Clients (your browser, Postman) request resources.
  - Servers respond to those requests.

HOW HTTP REQUESTS WORK

- **The Request-Response Cycle**:

  Request: Made by the client (browser or API tool) to request data.

  Response: Sent by the server, often in HTML, JSON, or another format.
- **Components of an HTTP Request**:

  URL: The address of the resource.

  Method: GET, POST, etc.

  Headers: Additional information, like content type.

  Body: (Optional) Data sent with requests (mainly POST).

Example:

https://www.oreilly.com/library/view/restful-java-web/9781788294041/1889f99d-f907-41c3-a0f0-925bbf1d3825.xhtml

TYPES OF HTTP REQUESTS

- **GET**: Retrieves data (read-only).
- **POST**: Sends new data to the server.
- **PUT**: Updates existing data.
- **DELETE**: Removes data.

Example:

- GET request to fetch weather data from an API.
- POST request to submit a form.

WHAT IS POSTMAN?

- **Postman** is a powerful tool for testing APIs and making HTTP requests.
- **Key Features**:

  Send different types of requests (GET, POST, PUT).

  Test and visualize responses.

  Set headers, parameters, and body data.
- **Why use it?**Makes working with APIs easier without writing code.

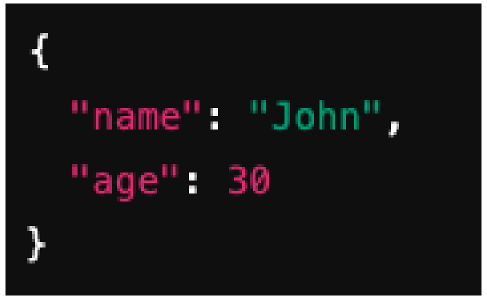
USING POSTMAN FOR BASIC HTTP REQUESTS

- **Step 1**: Install Postman from the official website.
- **Step 2**: Create a new request and set the method (GET, POST).
- **Step 3**: Enter the URL of the API endpoint.
- **Step 4**: Send the request and view the response (in JSON or other formats).

demo…

WHAT IS JSON?

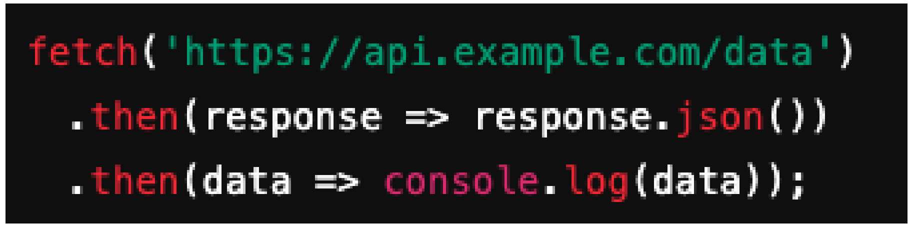
- **JavaScript Object Notation (JSON)** is a lightweight format for data exchange.
    - Syntax: key-value pairs (like JavaScript objects).
    - Easy to read and write.



```
{
  "name": "John",
  "age": 30
}
```

JSON REQUESTS WITH THE FETCH METHOD

- **Fetch**: A built-in JavaScript method for making HTTP requests.
- **Basic Structure**



```
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => console.log(data));
```

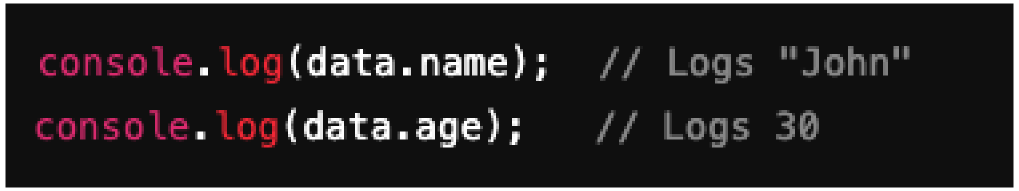
- **How it works**:
  - Fetch makes a request and receives a response.
  - The .json() method parses the JSON data.

USING JSON DATA IN JAVASCRIPT

- **Parsing JSON**:

  After fetching the data, you can access it in your JavaScript code.

- Example of accessing fields in JSON



```
console.log(data.name);   // Logs "John"
console.log(data.age);    // Logs 30
```

UPDATING DOM ELEMENTS WITH JSON DATA

- **DOM Manipulation**:

  Use JSON data to update elements on the web page dynamically.

  Practical Example:

  - Fetch weather data and update a weather dashboard.

```
document.getElementById('name').textContent = data.name;
```

CONCLUSION

- HTTP is essential for web communication.
- You can make HTTP requests using Postman or JavaScript (Fetch).
- JSON is the common data format for APIs.
- Learn how to update your web page dynamically using data from APIs.

QUESTIONS?