CoGrammar

Welcome to this session:

JSON and AJAX with Fetch

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators
answering questions.

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Skills Bootcamp Cloud Web Development

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Skills Bootcamp Cloud Web Development

- For all **non-academic questions**, please submit a query: ***www.hyperiondev.com/support***
- **Report a safeguarding incident: *www.hyperiondev.com/safeguardreporting***
- We would love your feedback on lectures: Feedback on Lectures
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Learning Outcomes

By the end of this lecture, students will be able to:

- Identify the components of an HTTP request and response.
- Explain the function of different HTTP methods (GET, POST, PUT, DELETE) in CRUD operations.
- Use the Fetch API to make asynchronous HTTP requests and handle responses.
- Parse and stringify JSON data using JSON.parse() and JSON.stringify().
- Implement client-side data persistence with sessionStorage and localStorage.

What is the purpose of an HTTP request?

A. To send data to the server.
B. To display data on the browser.
C. To receive data from the server.
D. None of the above.

CoGrammar

Which HTTP method is commonly used to retrieve data from a server?

A. POST
B. GET
C. PUT
D. PATCH
E. None of the above.

CoGrammar

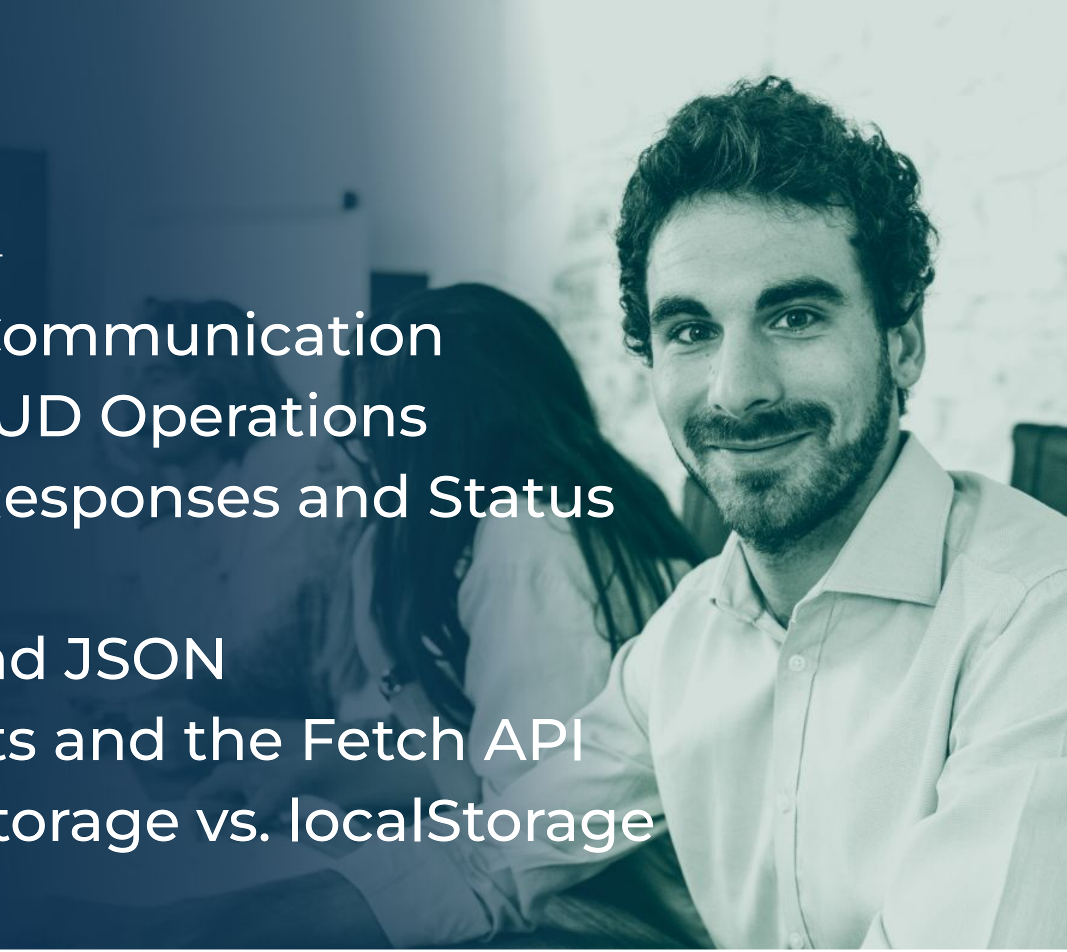
What does the status code 404 indicate in an HTTP response?

A. OK
B. Not Found
C. Internal Server Error
D. Found
E. None of the above.

CoGrammar

Lecture Overview

- Introduction to HTTP Communication
- HTTP Methods and CRUD Operations
- Understanding HTTP Responses and Status Codes
- Introduction to APIs and JSON
- Asynchronous Requests and the Fetch API
- Web Storage: sessionStorage vs. localStorage

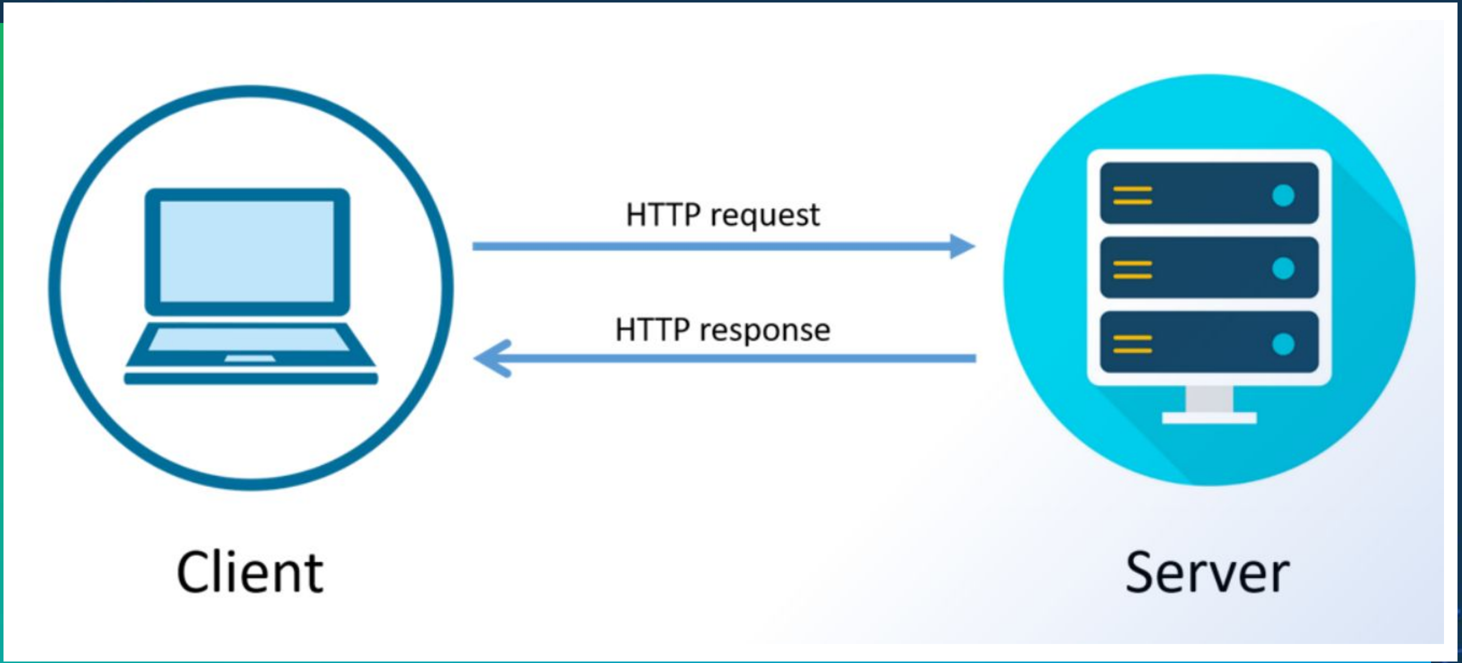


Introduction to Web Requests

- ❖ **Web requests**, also known as **HTTP requests**, are the foundation of communication between **clients** (such as web browsers) and **servers** over the World Wide Web.
- ❖ They facilitate the **exchange** of data and resources, allowing users to access and interact with web applications, websites, and services.
- ❖ **HTTP** (Hypertext Transfer Protocol) is the underlying protocol used for sending and receiving web requests and responses.
- ❖ It operates as a **request-response protocol**, where clients send requests to servers, and servers respond with the requested resources or information.

CoGrammar

Introduction to Web Requests



CoGrammar

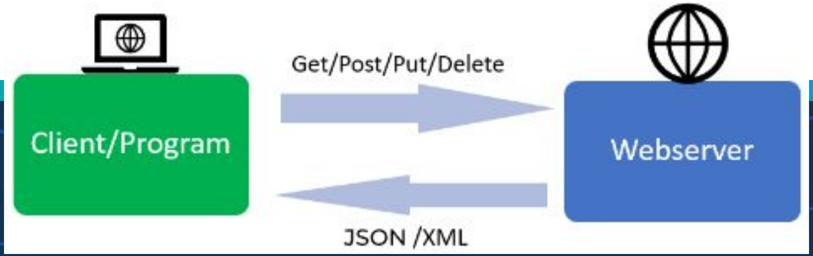
Components of an HTTP Request

- An HTTP **request** consists of several components, including:
  - Request Method (e.g., GET, POST, PUT, DELETE)
    - The request method indicates the desired action to be performed on the specified resource.
  - **URL** (Uniform Resource Locator) or **URI** (Uniform Resource Identifier).
    - The URL or URI specifies the **location** of the resource being requested by the client.
  - Headers (optional metadata).
  - Body (optional data for POST and PUT requests)

CoGrammar

Common Request Methods

- Requests are used to send your information to the server, asking it to perform a set of tasks.
- Examples of requests of CRUD (Create, Read, Update, and Delete) operations:
  - **GET** - Request or **read** data that **exists** within the database.
  - **POST** - We may want to **create** new (add new)items to the database.
  - **PUT** - This method allows you to **update existing data** on the server, such as changing the image of an item or updating the entire item’s details.
  - **PATCH** - Unlike PUT, which updates the entire resource, PATCH is used to make **partial updates** to **existing data**.
  - **DELETE** - As the name suggests, this **deletes** an item from a resource.



CoGrammar

Components of an HTTP Response

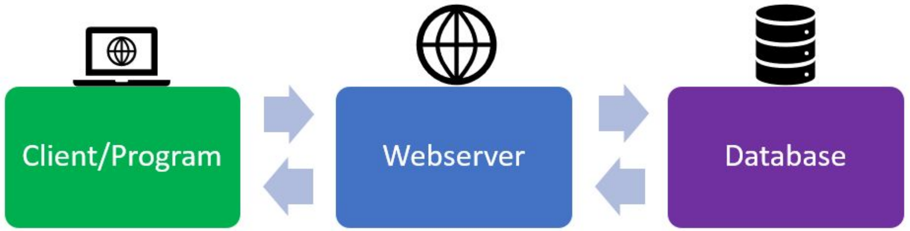
- An HTTP **response** consists of several components, including:
  - Status Code (e.g., 200 OK, 404 Not Found)
    - The status code indicates the outcome of the server's attempt to fulfill the client's request.
  - Headers (metadata): HTTP headers provide metadata about the response, including information about the server, content type, content length, caching directives, and more.
  - Body (response data): The response body contains the actual data or resource requested by the client, such as HTML content, JSON data, images, or files.

How Web Requests Work

- The client (e.g., web browser) sends an HTTP request to the server, specifying the desired action and resource.
- The server receives the request, processes it, and generates an appropriate response based on the request method and resource availability.
- The server sends an HTTP response back to the client, containing the requested data or indicating the outcome of the request (success, error, redirection, etc.).

What are APIs?

- An API (**A**pplication **P**rogramming **I**nterface) allows different software systems to communicate with each other.
- It defines the methods and data formats that applications can use to request and exchange information.
- APIs are fundamental for web development, enabling interaction with remote servers and services.



CoGrammar

Sending Objects between a Web Server and Clients

- **HTTP** allows information to be transferred across the internet. You need to keep in mind these 2 important facts about HTTP:
  - HTTP is a **stateless protocol** — it doesn't see any link between two requests being successively carried out on the same connection. **Cookies** and the **Web Storage API** are used to store necessary state information.
  - HTTP **transfers text** (not objects or other complex data structure). To send objects or other structured data, we need to turn them into text so they can be transferred using HTTP. Later, we can convert that text back into the original data structure once it's received.

CoGrammar

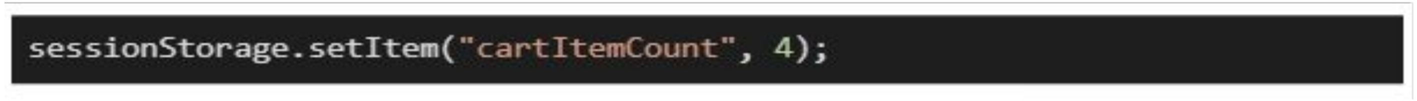
Web Storage API
Session & Local Storage

- The Web Storage API allows us to save data in the browser using key-value pairs.
  - sessionStorage
    - Stores data only for the duration of the current session. In other words, the data remains accessible only while the browser is open. Once the browser is closed, any data stored in sessionStorage is deleted.
  - localStorage
    - Retains data across multiple sessions. Unlike sessionStorage, the data remains accessible even after the browser is closed and reopened, staying available until it is manually deleted.

CoGrammar

Web Storage API
Session Storage syntax

❖ Adding a value to sessionStorage:

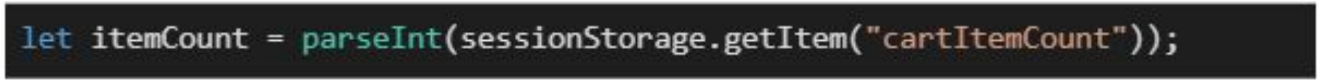


```
sessionStorage.setItem("cartItemCount", 4);
```

- .setItem() is a method used to add or update data in sessionStorage.
- The first argument, "cartItemCount", is the key.
- The second argument, 4, is the value.

Web Storage API
Session Storage syntax

❖ Retrieving a value from sessionStorage:



```javascript
let itemCount = parseInt(sessionStorage.getItem("cartItemCount"));
```

- .getItem() is a method used to get the value of a specific item stored in sessionStorage.
- The argument "cartItemCount" is the key used to identify the data you want to retrieve.
    - If there's no value stored with that key, itemCount will be null.

CoGrammar

What is JSON?

- ❖ JSON (**J**avaScript **O**bject **N**otation) is a lightweight, language-independent data format used to exchange data between a web server and a client.
- ❖ Key features of JSON:
  - ➢ Lightweight.
  - ➢ Language independent.
  - ➢ Easy to parse.

CoGrammar

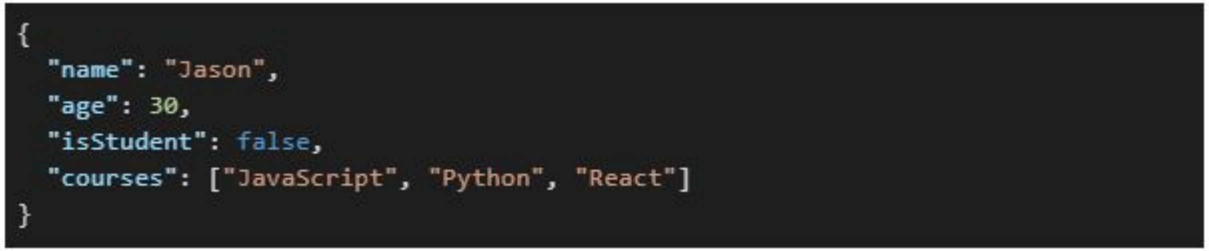
JSON’s syntax

❖ JSON’s syntax is similar to JavaScript object notation, but it has its own rules and conventions:

➢ Data in key-value pairs.

➢ Property names and values.

➢ Objects and arrays.



```
{
  "name": "Jason",
  "age": 30,
  "isStudent": false,
  "courses": ["JavaScript", "Python", "React"]
}
```

CoGrammar

JSON’s Methods: JSON.parse()

- **Reminder**: HTTP transfers **text** (not objects or other complex data structures).
- JSON.parse()
  - The JSON.parse() method in JavaScript is used to convert a JSON string into a corresponding JavaScript object.
  - Key Points:
    - **Input**: A string in valid JSON format.
    - **Output**: A JavaScript object or array based on the JSON data.
    - Throws an error if the string is not valid JSON.

CoGrammar

JSON's Methods: JSON.stringify()

- **Reminder**: HTTP transfers **text** (not objects or other complex data structures).
- JSON.stringify()
  - The JSON.stringify() method in JavaScript is used to convert a JavaScript object or value into a JSON string.
  - Key Points:
    - **Input**: A JavaScript object or value (e.g., array, boolean, number).
    - **Output**: A JSON-formatted string.

CoGrammar

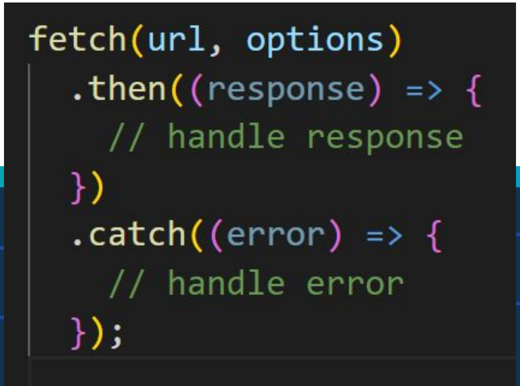
Overview of the Fetch API

- ❖ The Fetch API is a modern JavaScript interface for making **asynchronous** HTTP requests.
- ❖ Fetch follows a **promise-based** approach, which simplifies the handling of asynchronous operations, making code more readable and maintainable.
- ❖ Fetch returns a **Promise** that resolves to a **Response** object, allowing for seamless chaining of asynchronous operations using .then() and .catch() methods.

CoGrammar

fetch() Method

- The fetch() function is used to initiate a request to the specified url.
- It returns a promise that resolves to the Response object representing the response to the request.
- The options parameter is an optional object containing settings for the request such as method, headers, body, etc.



```
fetch(url, options)
  .then((response) => {
    // handle response
  })
  .catch((error) => {
    // handle error
  });
```

CoGrammar

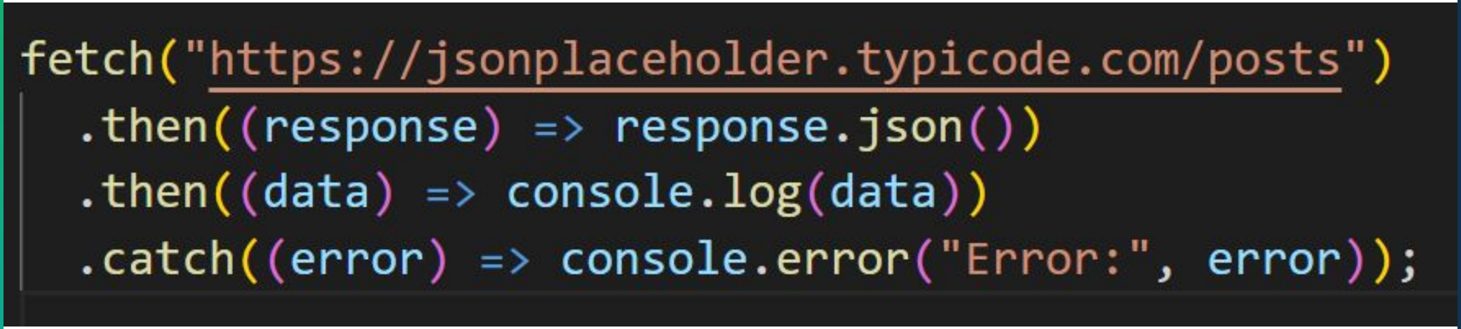
fetch() Method Parameters

- URL (required):
  - Specifies the URL to which the request is made.
- Options (optional): An object containing various settings for the request such as:
  - method: HTTP method (GET, POST, PUT, DELETE, etc.)
  - headers: Headers to include in the request.
  - body: Data to send with the request (e.g., JSON, FormData)

Response Handling

- ❖ Response Object:
  - ➢ Represents the response to the request made by fetch().
  - ➢ Contains properties and methods to access response data and metadata.
- ❖ .json():
  - ➢ Parses the response body as JSON.

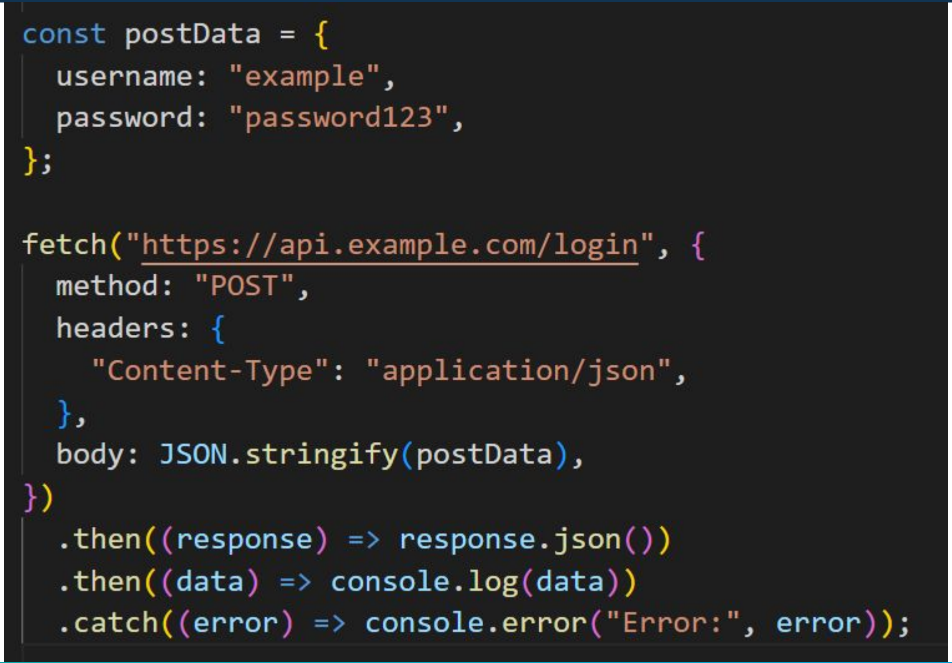
Making a GET Request



```
fetch("https://jsonplaceholder.typicode.com/posts")
  .then((response) => response.json())
  .then((data) => console.log(data))
  .catch((error) => console.error("Error:", error));
```

CoGrammar

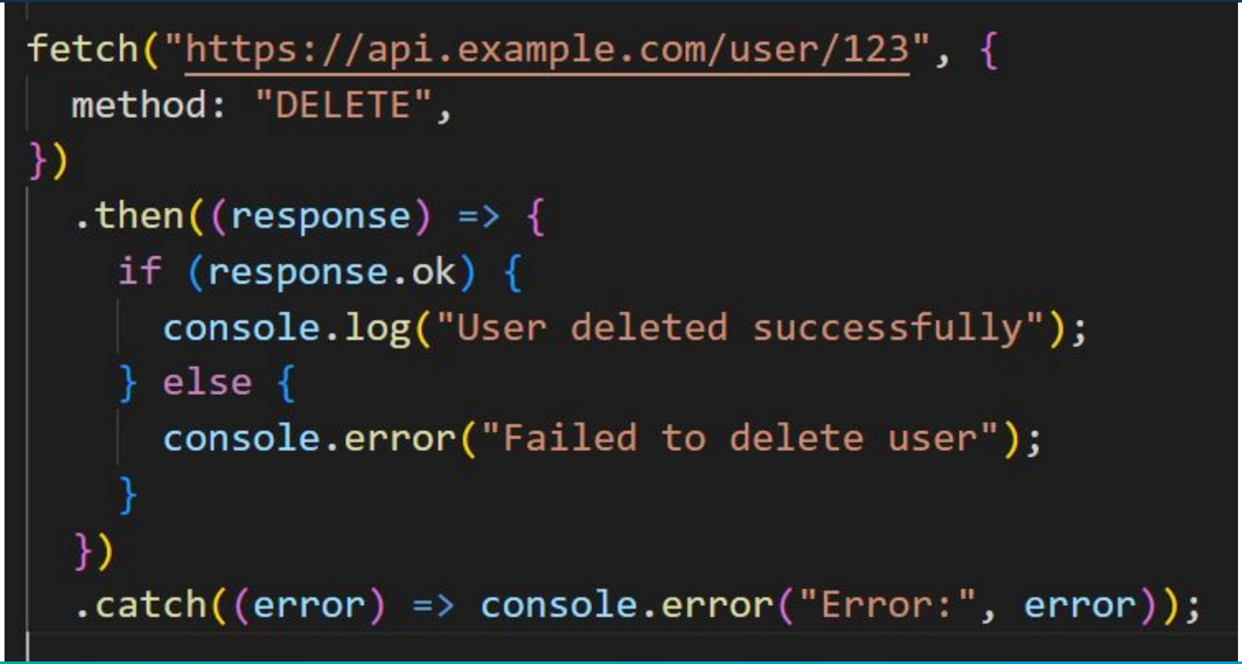
Making a POST Request



```
const postData = {
  username: "example",
  password: "password123",
};

fetch("https://api.example.com/login", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify(postData),
})
  .then((response) => response.json())
  .then((data) => console.log(data))
  .catch((error) => console.error("Error:", error));
```

Making a DELETE Request



```
fetch("https://api.example.com/user/123", {
  method: "DELETE",
})
  .then((response) => {
    if (response.ok) {
      console.log("User deleted successfully");
    } else {
      console.error("Failed to delete user");
    }
  })
  .catch((error) => console.error("Error:", error));
```

CoGrammar

What is the role of the Fetch API in web development?

A. To send HTTP requests asynchronously.
B. To store data in the browser.
C. To display data on the webpage.
D. None of the above.

Which of the following is true about JSON?

A. JSON can only store string data.
B. JSON is a lightweight data-interchange format.
C. JSON is used only for storing data in the browser.
D. None of the above.

CoGrammar

What does the JSON.parse() method do in JavaScript?

A. Converts a JavaScript object into a JSON string.
B. Converts a JSON string into a JavaScript object.
C. Converts a string into an HTML element.
D. None of the above.

CoGrammar

Let's take a break

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

CoGrammar

SKILLS FOR LIFE SKILLS BOOTCAMPS

Department for Education