Exercise 10: JavaScript and Web APIs

Practice calling APIs and presenting data in the DOM.

Theory

API stands for Application Programming Interface. A web api consists of one or more public endpoints. Most often, the http protocol is used and the endpoint is simply a uri (web address) to where on the server the data we want to retrieve is located.

The entire system is based on request and response. If we send a request to the endpoint https://github.com/orgs/Lexicon-NET-2023/repositories do we get a response back with all the repositories that are set up for the class (feel free to try pasting this into your browser and see what you get back). If we want to build a website for weather forecasts, we can make requests to an endpoint in MetaWeather's API instead of collecting the data ourselves. Different apis support different data formats but json and xml are the most common.

This enables us to use the resources (data) available on the api regardless of which programming language we develop with and regardless of which language the api is built with.

URI stands for Uniform Resource Identifier and is used to make requests to API libraries.

```
https://example.com/api/people?name=Kalle#nose
\___/ \_________/\_________/ \________/ \__/

schema authority              | path          question fragment
```

HTTP request methods, also called *HTTP verbs,* specify which action you want to perform with your request. The five most common are shown in the table below. If, for example, we want to retrieve data, we make a GET request. The only thing you need to use in this exercise is GOAT!

HTTP-VERB	CRUD
POST	Create
GET	Read
PUT	Update/Replace (full update)
PATCH	Update/Modify (partiell update)
DELETE	Delete

Request/ response body is the data that is sent, usually in JSON format.

```
{
    "object": {
        "a": "b",
        "c": "d",
        "e": "f"
    },
    "array": [
        1,
        2
    ],
    "string": "Hello World"
}
```

Postman is a program that can be used to call and test APIs while developing.

https://www.postman.com/downloads/

Task 1: Star Wars Biometrics

The Star Wars API contains lots of data regarding characters, species, spaceships, planets, etc. from the Star Wars films. Documentation: https://www.swapi.tech/documentation

For example, you can list all characters by making a GET request: https://www.swapi.tech/api/people

You can also make a GET request with a query string to get a specific character: https://www.swapi.tech/api/people/?name=chewbacca

Your task is to build a page where the user enters the name of a Star Wars character and gets biometric data back. Some example names are Luke Skywalker, R2-D2, Darth Vader, Leia Organa, Chewbacca.

1. Create a text box to receive the name. Use <input></input> and give it the attribute type=”text”.
2. Create a button with <button></button>.
3. Create a textbox for output with <textarea></textarea> and give it at least 5 five lines with the rows attribute.
4. Write a function in the js file like:
    a. reads the value of the input text box and adds it to the string “https://www.swapi.tech/api/people/?name=” so we get a full URI,
    b. uses the URI as a parameter to a fetch request,
    c. and prints the information to the output text box.

```
function getApi(){

/*Write your code here*/

    fetch(fullUri)
        .then(res => res.json())
        .then(data =>
        {
            /*And here*/
        })
        .catch(err => console.log(err))
}
```

data is a JSON object that contains all retrieved information about the character. Check the documentation or print with console.log to see how the object is built, Extract height, mass, gender and hair_color and use them to build a string using template literals (backslashes).

Example: let s = `Value 1: ${value1}, Value 2: ${value2}`;

5. Add an eventListener that listens to the button and runs the above function at click.

Task 2: Draw a card

The Deck of Cards API allows us to draw cards from one or more decks. In the response body, you get both the value the card has (suit and value) as well as a url to an image of the drawn card. Documentation: http://deckofcardsapi.com/

You must write a page that shows one card at a time when the user clicks a button.

1. Create a <button></button>.
2. Create a <div></div> where the image on the card is to be sliced.
3. Write a similar js file as in the previous task with some differences:
   a. Use this URI for the fetch:
      "https://deckofcardsapi.com/api/deck/new/draw/?count=1"
   b. Check the documentation to see how to extract the url to the image on the card from the data.
   c. Use createElement, setAttribute, innerHTML and appendChild to create an image element, set its src attribute, reset the div and add the image element to it.

4. Add an eventListener that listens to the button and runs the above function at click.

Extra

Build a card game :) Take help from the documentation for the ape. You choose what you build.

A simple Blackjack game perhaps?