

FETCH!

LOGISTICS

- MP3 share back!
- Final Project Intro
- Break
- The Fetch API
- Fetch Activity

MP3 SHARE!

- add your game to [this spreadsheet](#)

FINAL PROJECT ROADMAP

- **2/28:** Proposal due.
 - FP0 writeup posted to portfolio
 - You are **REQUIRED** to check in with me about your project idea
- **3/7:** Prototype due.
 - FP1 writeup posted to portfolio
- **3/9:** Last class, demo day!
 - You are **REQUIRED** to demo your prototype in class.
- **3/17:** Final version due, and final portfolio due.
 - FP2 writeup posted to portfolio

FPO: PROPOSAL

FPO Assignment

- One-sentence pitch
- At least two drawings/mockups of the interface
- Two feature lists: **prototype** and **final version**
- A step-by-step plan of attack
- A list of known unknowns
- A list of stretch goals

WHAT CAN YOU MAKE?

- See list on [FP0 Assignment description](#)
- I must approve your project, in-class next Tuesday!

REMEMBER!

- Work on your portfolio! Make it look nice!
- Ensure it works on mobile!
- Remember to add to your cheat sheets!
- **Help your peers in Discord!**

BREAK

FETCH

- How we ask for data from APIs!
- Modern alternative to XMLHttpRequest
- Accepts the path to the resource you want to fetch
- Returns a promise with the response

BASIC FETCH REQUEST

```
fetch("https://jsonplaceholder.typicode.com/users")  
  .then((response) => response.json())  
  .then((data) => console.log(data));
```

Fetch doesn't directly return the JSON response body.
Instead, it returns a *promise* that resolves with a
Response object.

BASIC FETCH REQUEST

```
fetch("https://jsonplaceholder.typicode.com/users")  
  .then((response) => response.json())  
  .then((data) => console.log(data));
```

The **Response** object also does not directly contain the actual JSON response body. We Use the **json()** method to extract the JSON body content from the **Response**.

RESPONSE BODY TYPES

- `response.text()`
- `response.json()`
- `response.formData()`
- `response.blob()` (binary data)
- `response.arrayBuffer()` (low-level binary data)

READING RESPONSE AS TEXT

```
fetch("https://jsonplaceholder.typicode.com/users")  
  .then((response) => response.text())  
  .then((data) => console.log(data));
```

CORS ERRORS ON LOCALHOST

- **CORS:** Cross-Origin Resource Sharing
- Allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources
- *Super annoying!*
- Chrome does not support localhost CORS requests
- There is an [extension](#) you can use to get around this issue

BIG LIST OF PUBLIC APIS

<https://github.com/public-apis/public-apis>

Some I tried before class:

```
https://api.sunrise-sunset.org/json?lat=36.72&lng=-4.42  
https://dog-api.kinduff.com/api/facts?number=5  
https://emojihub.yurace.pro/api/random
```

LET'S TRY IT!

```
fetch("https://api.sunrise-sunset.org/json?lat=36.72&lng=-4.42")  
  .then((response) => response.json())  
  .then((data) => console.log(data));
```


POSTING WITH FETCH

```
const data = { username: "example" };  
// Specify that we want to POST data, what the type of data is,  
// and the body of the data  
fetch("https://example.com/profile", {  
  method: "POST", // or 'PUT'  
  headers: {  
    "Content-Type": "application/json",  
  },  
  body: JSON.stringify(data),  
})  
  .then((response) => response.json())  
  .then((data) => {  
    console.log("Success:", data);  
  })  
  .catch((error) => {  
    console.error("Error:", error);  
  });
```

CHECKING FOR SUCCESS

```
fetch("flowers.jpg")
  .then((response) => {
    // Checking for success with response.ok
    if (!response.ok) {
      // Throw error if not ok
      throw new Error("Network response was not OK");
    }
    return response.blob();
  })
  .then((myBlob) => {
    myImage.src = URL.createObjectURL(myBlob);
  })
  .catch((error) => {
    // This runs if there is an error
    console.error("Fetch encountered an error:", error);
  });
```

USING HEADERS

```
fetch(myRequest)
  .then((response) => {
    const contentType = response.headers.get("content-type");
    // Check to see if the content type is JSON
    if (!contentType || !contentType.includes("application/json")) {
      throw new TypeError("Oops, we haven't got JSON!");
    }
    return response.json();
  })
  .then((data) => {
    /* process your data further */
  })
  .catch((error) => console.error(error));
```

ACTIVITY: FETCH

```
fetch("https://jsonplaceholder.typicode.com/users")  
  .then((response) => response.json())  
  .then((data) => console.log(data));
```

activities/08 fetch

