External APIS and ReactJS

A practical intro

API's and You

What is an API?

- API stands for Application Programming Interface.
- It allows different software applications to communicate and interact with each other.
- APIs provide a set of rules and protocols for how different software components should interact.

React and API's

Why use External APIs in React?

- External APIs provide a vast range of data and functionality that can enhance your React applications.

- By integrating external APIs, you can access real-time data, perform actions on remote systems, and leverage existing services.

Making API Requests in React

External API's and React

- React provides different methods and libraries to make API requests.

- Popular options include the Fetch API, Axios, and the built-in fetch() function.

The Fetch API

- The Fetch API is a browser-based API for making HTTP requests.
- It provides a simple and flexible way to interact with APIs.
- You can use the fetch() function to make GET, POST, PUT, DELETE, and other HTTP requests.
- The fetch() function returns a Promise that resolves to the response from the API.

Example: Making a GET Request with Fetch

```
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => {
    // Handle the data returned by the API
    // Perform operations like updating state or rendering the data in your UI
  })
  .catch(error => {
    // Handle errors if the API request fails
  });
```

- In this example, we make a GET request to https://api.example.com/data.
- The response is converted to JSON format using the response.json() method.
- We then handle the data in the subsequent .then() block.
- If there's an error during the request, it is caught in the .catch() block.

Axios Library

- Axios is a popular JavaScript library for making HTTP requests.
- It provides a simpler and more expressive API compared to the Fetch API.
- Axios supports features like request cancellation, interceptors, and automatic JSON parsing.

Example: Making a GET Request with Axios

```
axios.get('https://api.example.com/data')
  .then(response => {
    // Handle the response data
    // Update state or render the data in your UI
  })
  .catch(error => {
    // Handle errors if the API request fails
  });
```

- In this example, we use Axios to make a GET request to https://api.example.com/data.
- The response data is handled in the .then() block.
- Any errors during the request are caught in the .catch() block.

Handling API Responses

- Once you receive a response from the API, you can handle it based on your application's requirements.

- Common tasks include parsing and manipulating the data, updating React component state, and rendering the data in your UI.
- In our case, we will usually handle a JSON response.
  - JSON: **J**ava**S**cript **O**bject **N**otation

Recap

- External APIs provide additional data and functionality to enhance React applications.
- React offers various methods and libraries like Fetch API and Axios to make API requests.
- Remember to handle API responses appropriately and consider error handling.
- NOTE: This is very high level and we will go deeper into promises in the next lecture

Let’s Build Something!