JS

Fetch API Cheatsheet

🚀 What is Fetch API?

- Fetch API is used to make http requests like GET, POST, PUT, DELETE in easy way. By using Fetch method, we can get data from backend/any-api, we can also send data to backend and much more.

- There are AJAX and XMLHttpRequest also for making API request but they are little complex in compare to Fetch method.

📌 Making GET request using Fetch method

- So for GET request we have to just pass the API url in the fetch method.

- Now for handling promise, we use then method. First of all, we have to convert the response(which API send) into json format. So for that we use `res.json()`

- Now this then method again return a promise. So we use another then method to handle it and we get the data as parameter.

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

📌 Making POST request using Fetch method

- Usaually POST request is used to send data to the backend or for creating new data. For example, Register a new user, login user etc.

- So for making POST request we have to pass POST api url as first argument. But we have to pass one more argument which is configuration object.

- This object has few properties. Like method for specify API method, body for sending data with that request but for some API we have to convert our data in

to string. So we use here `JSON.stringify(data)`

- And in header we specify `"Content-type": "application/json"` which is totally depend on API.

```js
const data = {
    title: "This is title",
    body: "This is post body",
    userId: 2,
};

fetch("https://jsonplaceholder.typicode.com/posts", {
    method: "POST",
    body: JSON.stringify(data),
    headers: {
        "Content-type": "application/json",
    },
})
    .then((res) => res.json())
    .then((data) => console.log(data));
```

- Now if you want to make PUT or DELETE request, then you have to only change the method type in configuration object.

🎁 Bonus - Fetch method with Async-await

- Async-await is used to handle promise in much simpler way. By using them we can make our code more simple.

```js
async function getUsers() {
    const res = await fetch("https://jsonplaceholder.typicode.com/users");
    const data = res.json();
    console.log(data);
}

getUsers();
```

- If you don't know promise and async-await, don't worry I created tutorial on Asynchronous JavaScript and in that I explain it in very detailed way.

Thank you so much for accessing this cheatsheet. If you like this then you can watch my more tutorial on My YouTube channel ~ Code Bless You🩶.