AlgoTutor

4 ways to make an API call

+91-7260058093
www.algotutor.io

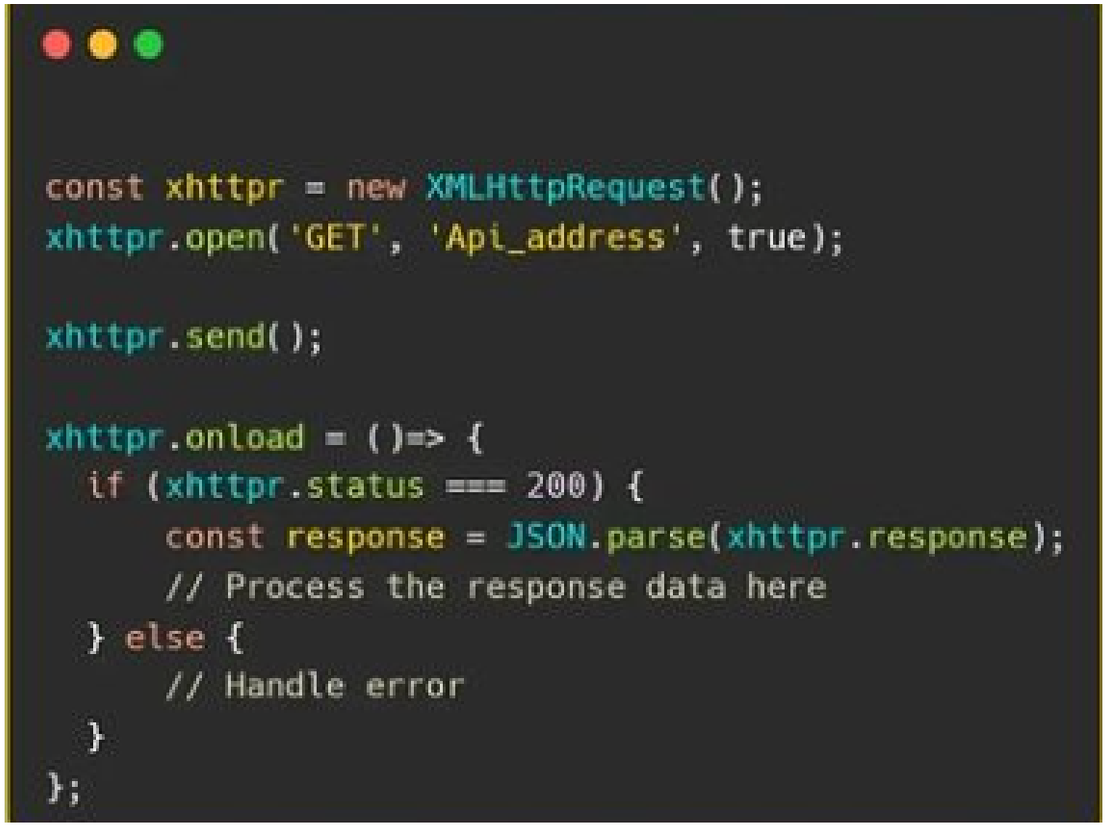
Introduction

In JavaScript, we often use APIs to interact with servers and get or send data.

There are four different ways to make these API calls, each with its own use cases. Let's explore each of them.

1/ Using XMLHttpRequest

Before ES6 and modern libraries like Fetch and Axios, XMLHttpRequest was the only way to make API calls in JavaScript. It's still widely used because it's supported by all browsers.



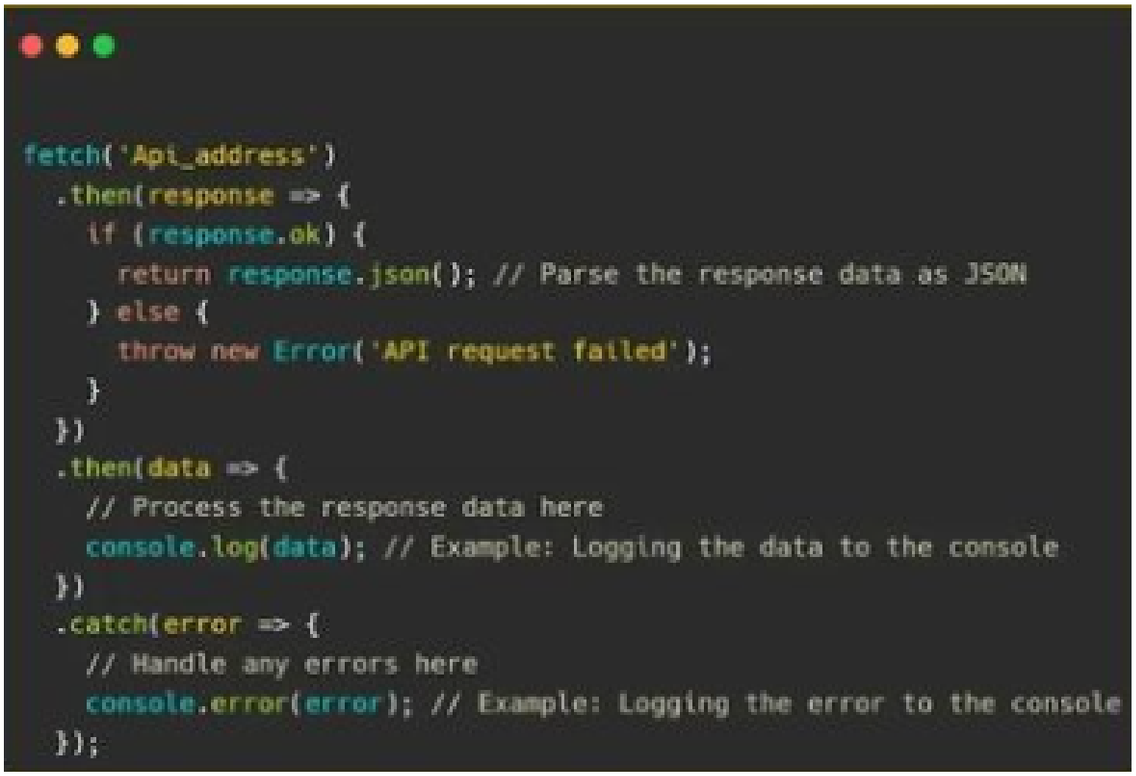
```
const xhttpr = new XMLHttpRequest();
xhttpr.open('GET', 'Api_address', true);

xhttpr.send();

xhttpr.onload = ()=> {
  if (xhttpr.status === 200) {
      const response = JSON.parse(xhttpr.response);
      // Process the response data here
  } else {
      // Handle error
  }
};
```

This JavaScript code uses XMLHttpRequest to make a GET request to an API. After sending the request, it checks if the status is 200 (OK).

If true, it parses the JSON response; otherwise, it handles errors.

2/ Using fetch() method

Fetch is a method to call an API in JavaScript.

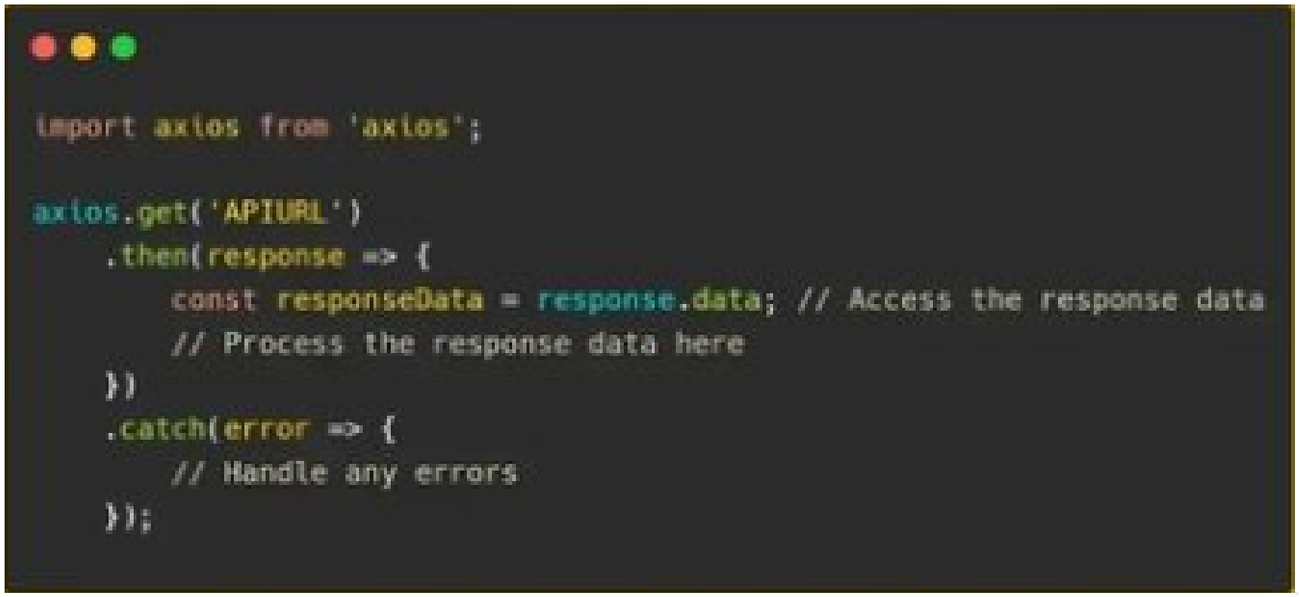


```
fetch('Api_address')
  .then(response => {
    if (response.ok) {
      return response.json(); // Parse the response data as JSON
    } else {
      throw new Error('API request failed');
    }
  })
  .then(data => {
    // Process the response data here
    console.log(data); // Example: Logging the data to the console
  })
  .catch(error => {
    // Handle any errors here
    console.error(error); // Example: Logging the error to the console
  });
```

It returns a promise, which contains a single value, either response data or an error.

3/ Using Axios

Axios is an open-source library for making HTTP requests to servers. It is a promise-based approach.
It supports all modern browsers and is used in real-time applications. It is easy to install using the npm package manager.



```
import axios from 'axios';

axios.get('APIURL')
    .then(response => {
        const responseData = response.data; // Access the response data
        // Process the response data here
    })
    .catch(error => {
        // Handle any errors
    });
```

4/ Using the jQuery AJAX

jQuery is a library used to make JavaScript programming simple and if you are using it then with the help of the $.ajax() method you can make asynchronous HTTP requests to get data.