SHEKHOBABA.COM

4 WAYS TO MAKE AN API CALL IN JS

Burhan Tahir

@iamshekhobaba

JavaScript provides a few built-in methods and external open source libraries to create and interact with API.

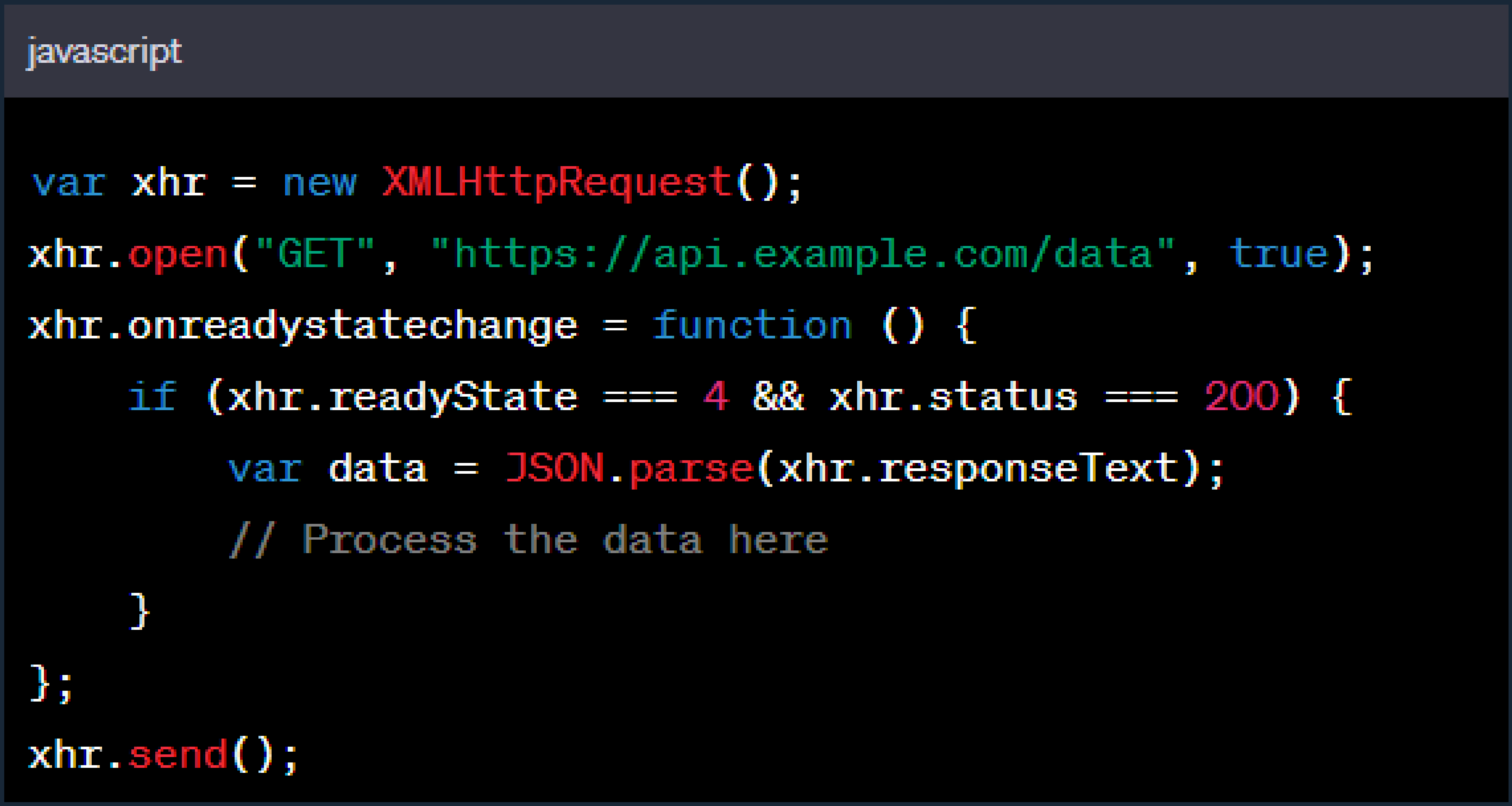
You can make API calls in JavaScript using various methods and libraries. Here are four common ways to do it:

1. XMLHttpRequest
2. fetch
3. axios
4. jQuery.ajax()

Burhan Tahir

@iamshekhobaba

XML HTTP REQUEST



javascript

```javascript
var xhr = new XMLHttpRequest();
xhr.open("GET", "https://api.example.com/data", true);
xhr.onreadystatechange = function () {
    if (xhr.readyState === 4 && xhr.status === 200) {
        var data = JSON.parse(xhr.responseText);
        // Process the data here
    }
};
xhr.send();
```

Burhan Tahir

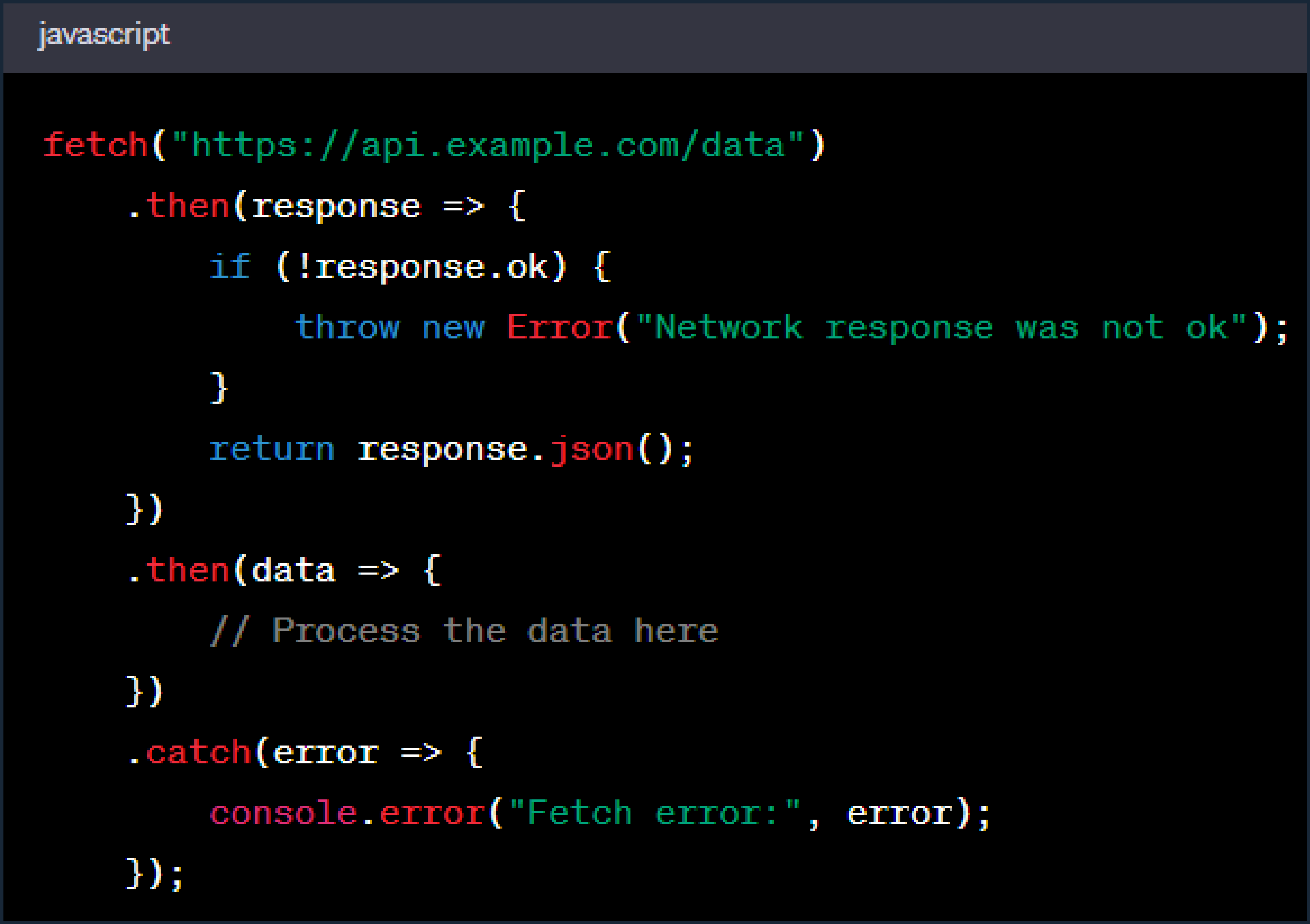
@iamshekhobaba

XML HTTP REQUEST

- XMLHttpRequest is a native JavaScript object that allows you to make HTTP requests.
- You create an instance of XMLHttpRequest, open a connection using open(), specify the HTTP method ("GET" in this case), and provide the API endpoint URL.
- You set up an event handler (onreadystatechange) to handle the response when it's received.
- Once the request is configured, you send it using send().
- When the response is received (readyState === 4), and the status code is 200 (indicating success), you can process the data.

Burhan Tahir
@iamshekhobaba

FETCH



```javascript
fetch("https://api.example.com/data")
    .then(response => {
        if (!response.ok) {
            throw new Error("Network response was not ok");
        }
        return response.json();
    })
    .then(data => {
        // Process the data here
    })
    .catch(error => {
        console.error("Fetch error:", error);
    });
```

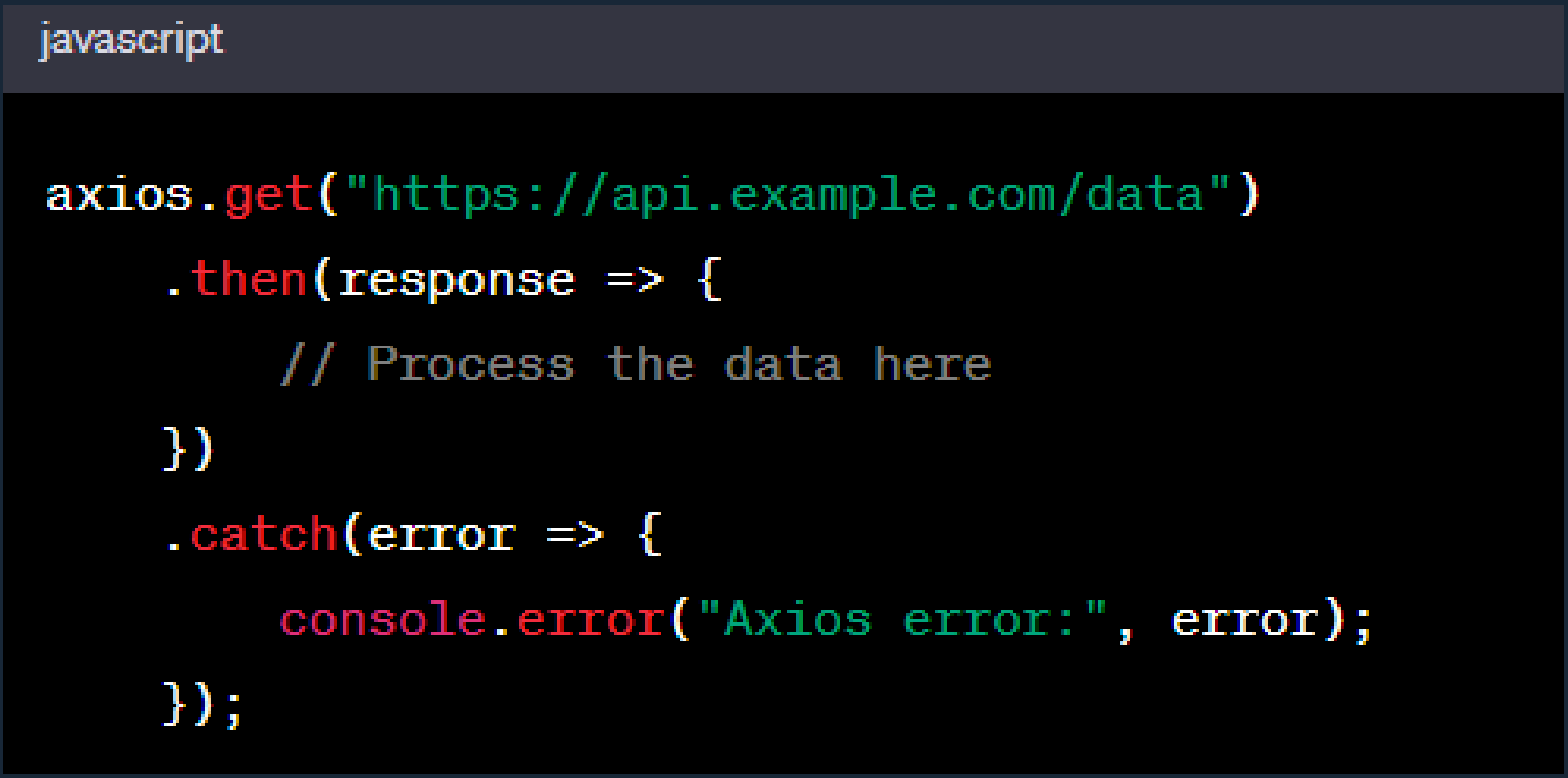
Burhan Tahir

@iamshekhobaba

FETCH

- The fetch API is a modern way to make HTTP requests and returns Promises.
- You use fetch to make a GET request to the API endpoint.
- The .then() method is used to handle the response, including error checking and parsing the JSON response.
- You can chain multiple .then() blocks for additional processing.
- The .catch() block handles any errors that occur during the request.

AXIOS



javascript

```javascript
axios.get("https://api.example.com/data")
    .then(response => {
        // Process the data here
    })
    .catch(error => {
        console.error("Axios error:", error);
    });
```

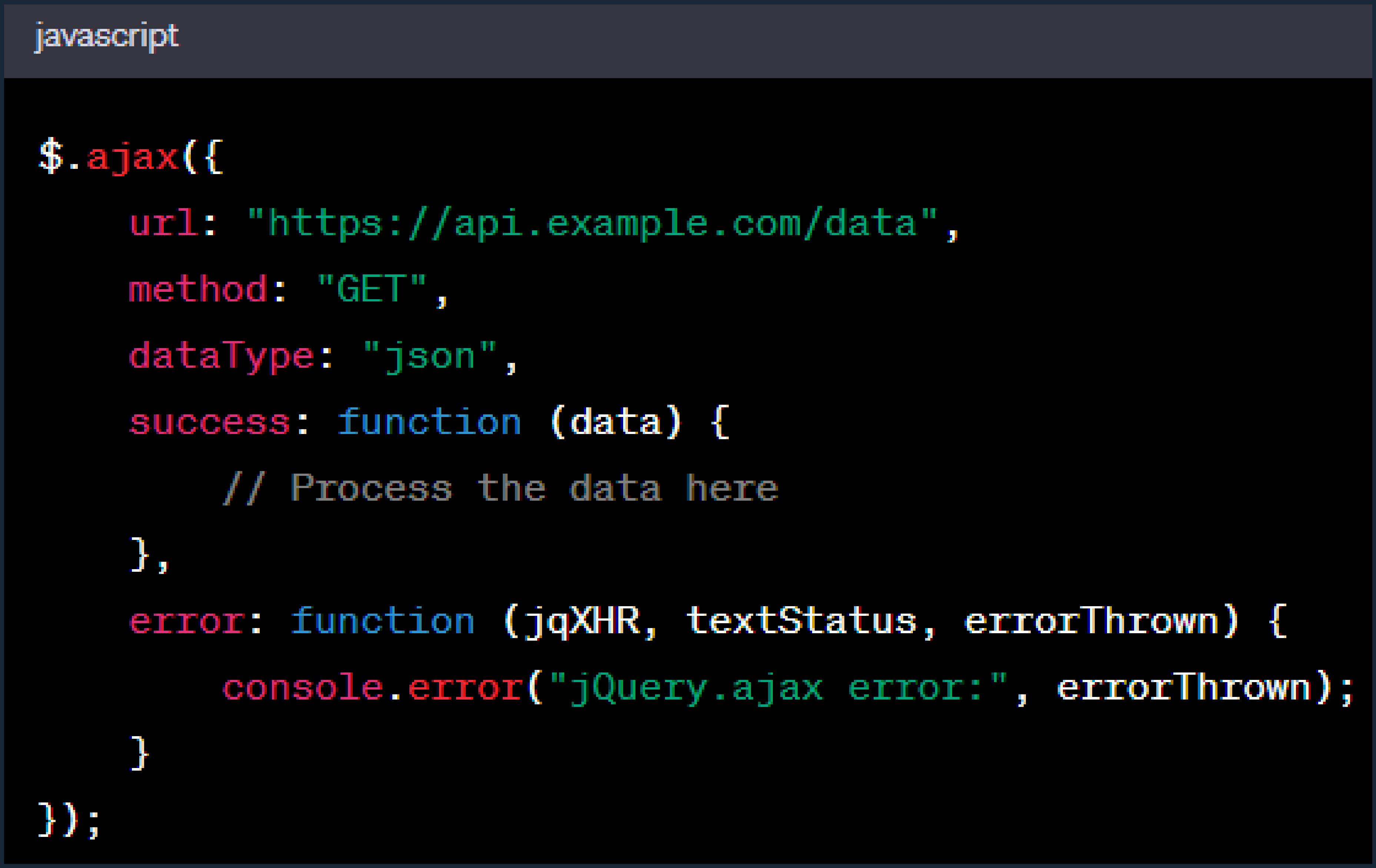
Burhan Tahir

@iamshekhobaba

AXIOS

- Axios is a popular JavaScript library for making HTTP requests.
- You make a GET request using axios.get() and provide the API URL.
- You use .then() to handle the response data and .catch() for error handling.
- Axios simplifies the request process and automatically parses JSON responses.

JQUERY.AJAX()



javascript

```javascript
$.ajax({
    url: "https://api.example.com/data",
    method: "GET",
    dataType: "json",
    success: function (data) {
        // Process the data here
    },
    error: function (jqXHR, textStatus, errorThrown) {
        console.error("jQuery.ajax error:", errorThrown);
    }
});
```

Burhan Tahir

@iamshekhobaba

JQUERY.AJAX()

- If you're using jQuery, you can use $.ajax() to make AJAX requests.
- You specify the request details in the form of an options object, including the URL, HTTP method, data type, success callback, and error callback.
- The success callback handles the response data, and the error callback handles any request errors.
- jQuery simplifies AJAX requests but is less commonly used in modern web development due to native alternatives like fetch.

Burhan Tahir
@iamshekhobaba

Did you find it Useful?

Leave a comment!

Alamin CodePapa

@CodePapa360

FOLLOW FOR MORE

Like

Comment

Repost