Open in app Get started

Published in Level Up Coding

Jayanth babu Follow
Jan 1, 2020 · 4 min read · Listen

Save

All possible ways of making an API call in JavaScript.

Photo by Eric Prouzet on Unsplash

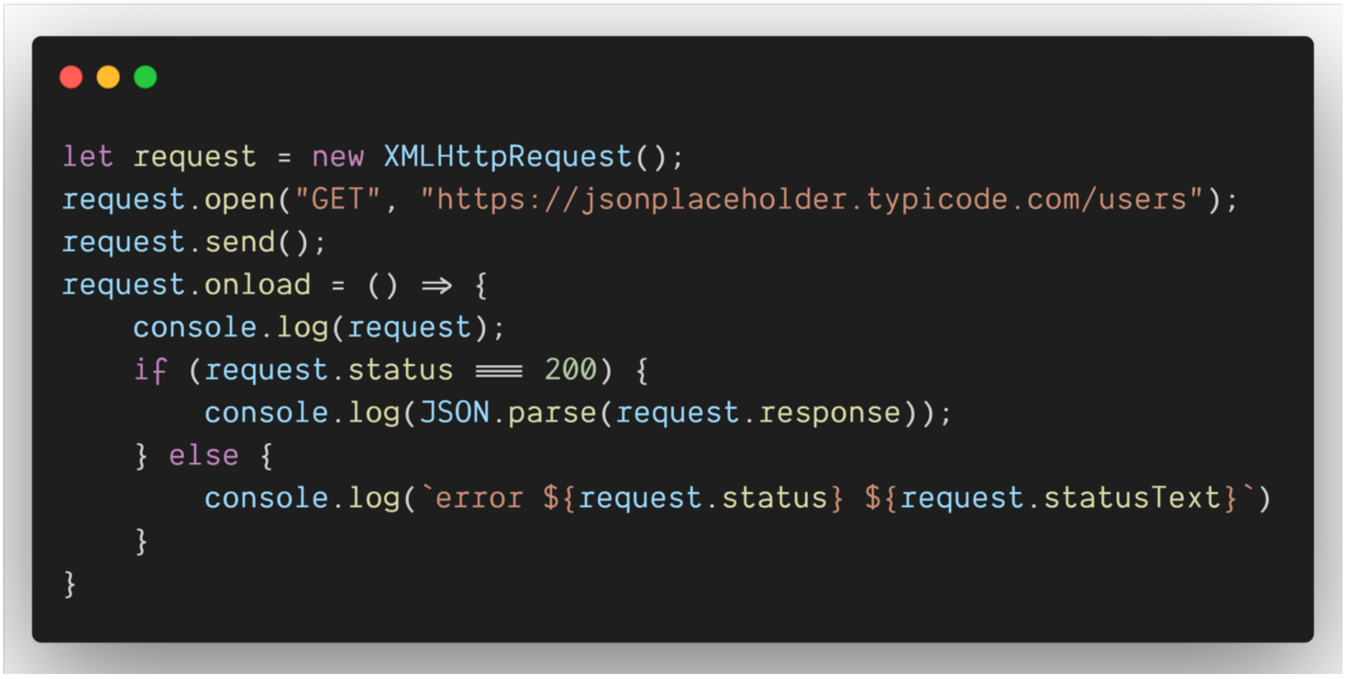
In JavaScript, it was really important to know how to make HTTP requests and retrieve the dynamic data from the server/database.

1. XMLHttpRequest
2. fetch
3. Axios
4. jQuery

XMLHttpRequest

Before ES 6 comes out, the only way to make an HTTP request in JavaScript was `XMLHttpRequest` . It is a built-in browser object that allows us to make HTTP requests in JavaScript.

JSONPlaceholder is a free online REST API that you can use whenever you need some fake data.



```
let request = new XMLHttpRequest();
request.open("GET", "https://jsonplaceholder.typicode.com/users");
request.send();
request.onload = () => {
    console.log(request);
    if (request.status === 200) {
        console.log(JSON.parse(request.response));
    } else {
        console.log(`error ${request.status} ${request.statusText}`)
    }
}
```

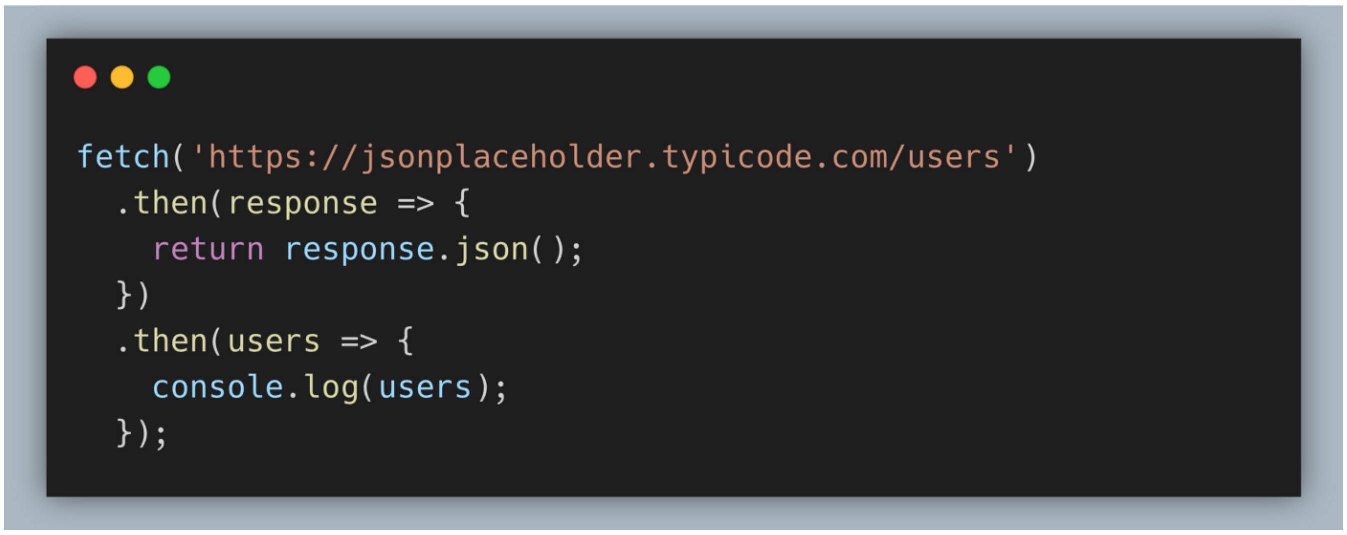
By default we receive the response in the string format, we need to parse into JSON.

XMLHttpRequest was deprecated in ES 6 by the introduction of **fetch.** But still, we are using **XMLHttpRequest** when we need to work with old browsers and don’t want polyfills.

(can be polyfilled), but very well supported among the modern ones. We can make an API call by using fetch in two ways.

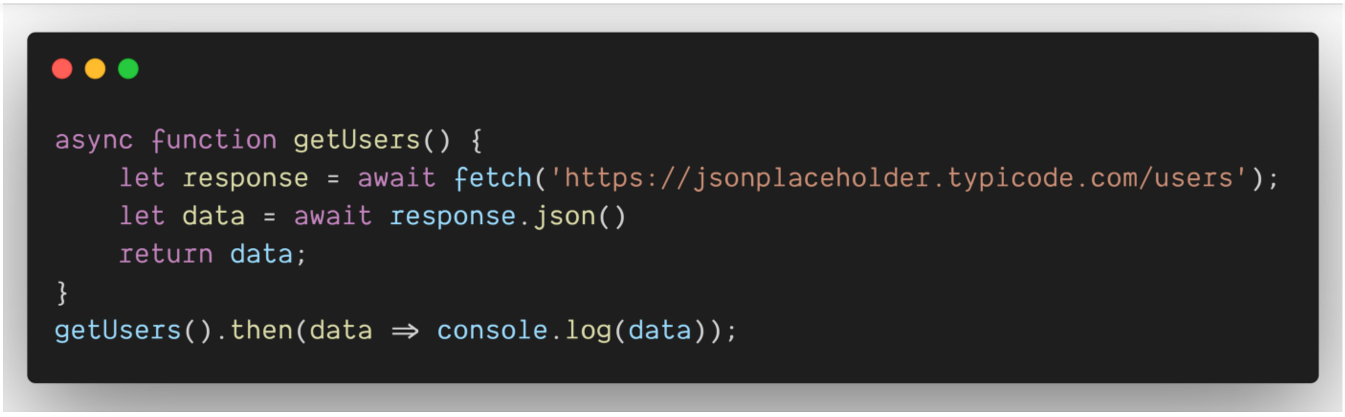
How to make an API call in JavaScript?

Method 1:



```
fetch('https://jsonplaceholder.typicode.com/users')
  .then(response => {
    return response.json();
  })
  .then(users => {
    console.log(users);
  });
```

By using Async and Await



```
async function getUsers() {
    let response = await fetch('https://jsonplaceholder.typicode.com/users');
    let data = await response.json()
    return data;
}
getUsers().then(data => console.log(data));
```

The fetch API is very powerful. We can easily send AJAX requests using the browser fetch API. The major disadvantage of fetch API is error handling.

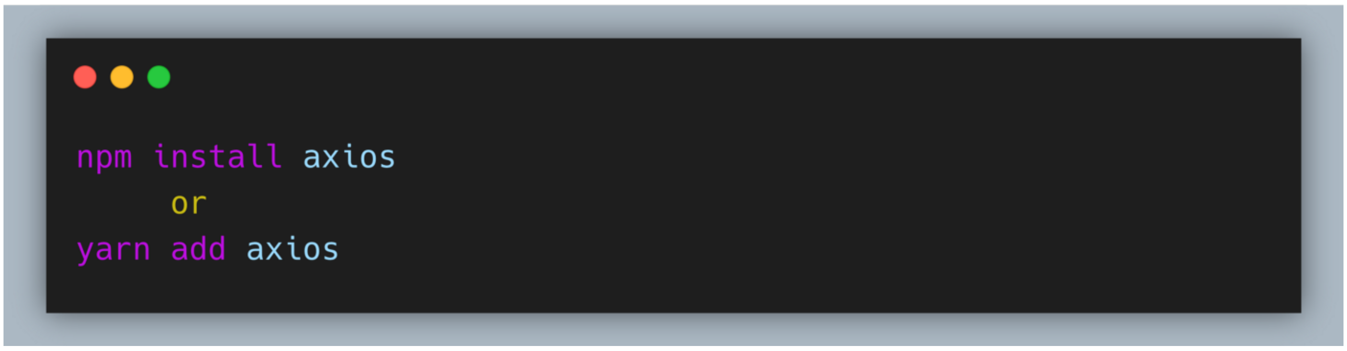
Axios

Axios is an open-source library for making HTTP requests and provides many great features, and it works both in browsers and Node.js. It is a promise-based HTTP client that can be used in plain JavaScript and advanced frameworks like React, Vue.js, and Angular.

It supports all modern browsers, including support for IE 8 and higher.

Installation:

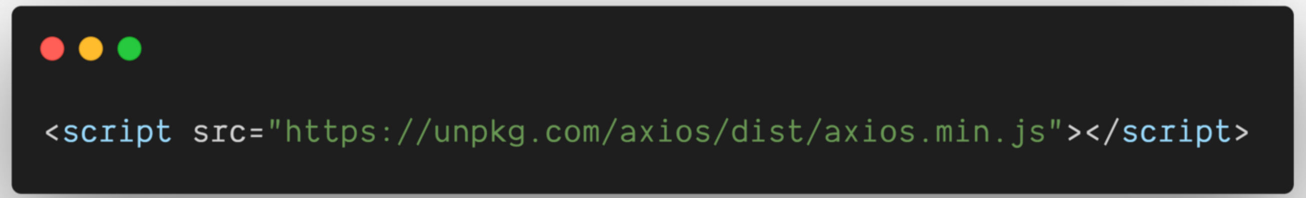
If you are using any one of the package managers like **npm or yarn**.



```
npm install axios
      or
yarn add axios
```

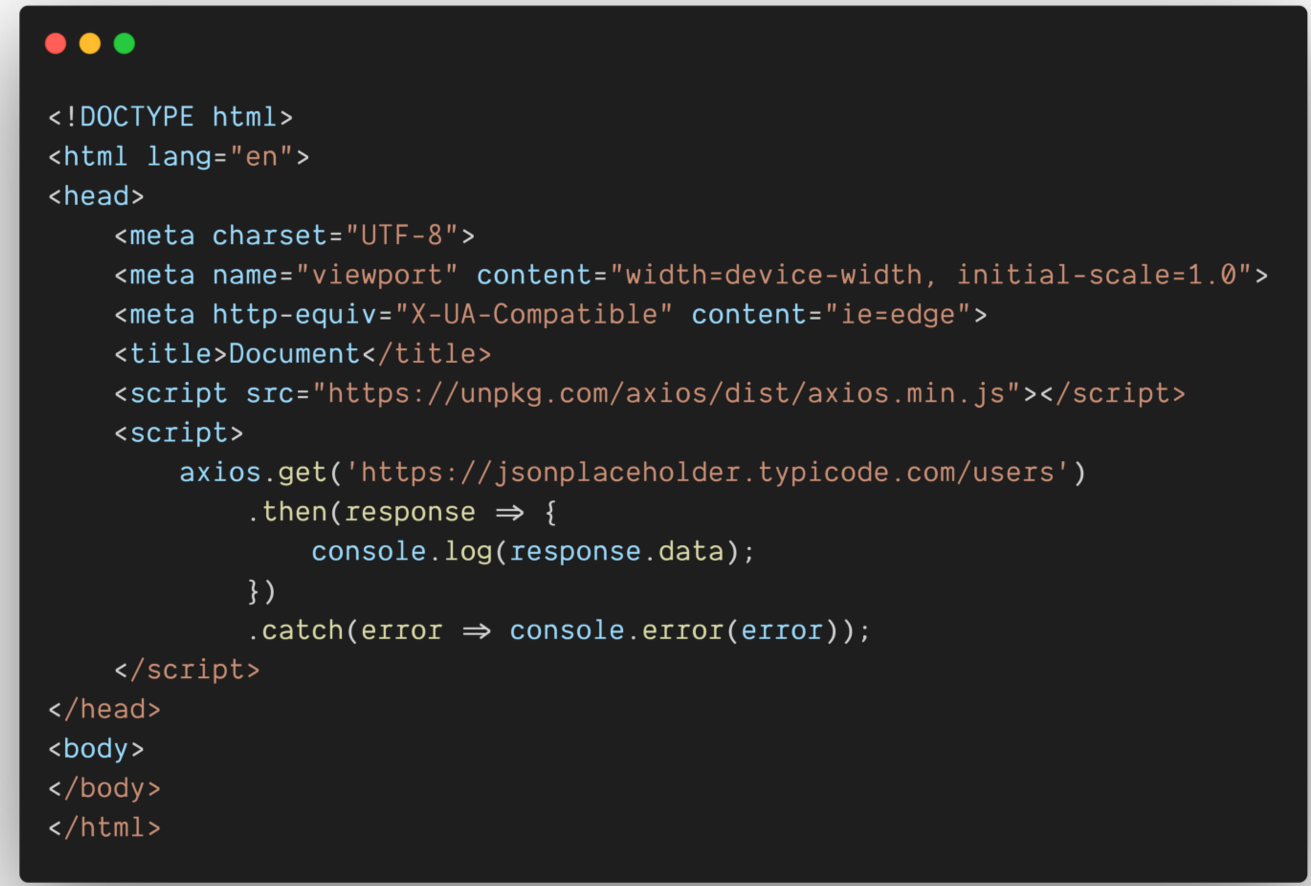
And include it in HTML file like this

The easiest way to include Axios is by using external CDN:



```html
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
```

Now you can start sending HTTP request by including the following script in your HTML file.



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
    <script>
        axios.get('https://jsonplaceholder.typicode.com/users')
            .then(response => {
                console.log(response.data);
            })
            .catch(error => console.error(error));
    </script>
</head>
<body>
</body>
</html>
```

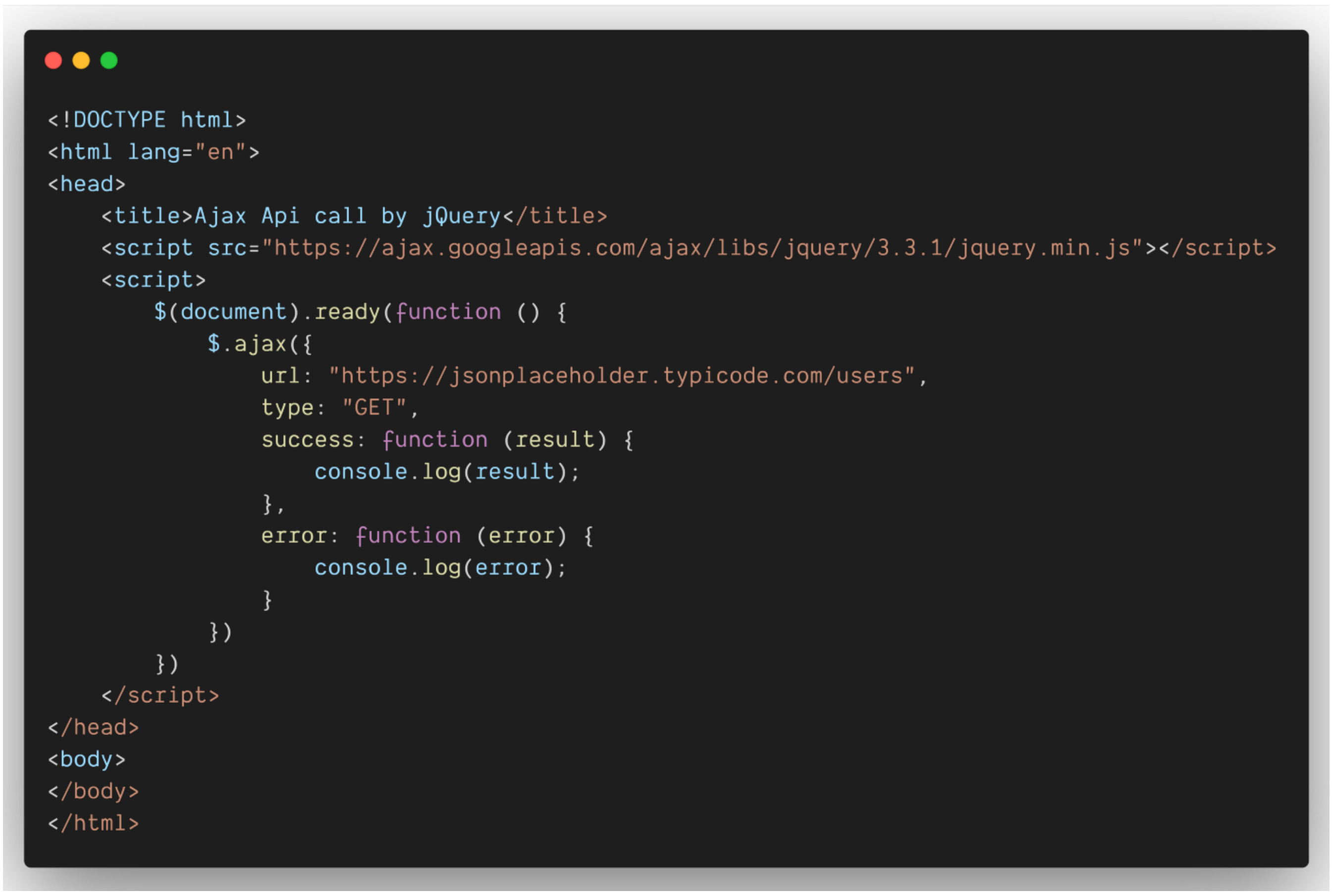
The following are the advantages of **Axios**

1. Axios performs automatic transformations and returns the data in JSON format.

2. Better error handling

jQuery has many methods to handle asynchronous HTTP requests. In order to use jQuery, we need to include the source file of jQuery, and $.ajax() method is used to make the HTTP request.

$.ajax()



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Ajax Api call by jQuery</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script>
        $(document).ready(function () {
            $.ajax({
                url: "https://jsonplaceholder.typicode.com/users",
                type: "GET",
                success: function (result) {
                    console.log(result);
                },
                error: function (error) {
                    console.log(error);
                }
            })
        })
    </script>
</head>
<body>
</body>
</html>
```

The $.ajax method takes many parameters, some of which are required and others optional. It contains two callback functions `success` and `error` to handle the response received.

Conclusion

Most of the real-time applications are using Axios to make HTTP requests. Axios is very easy and an open-source library for making HTTP requests.

I have covered the most popular ways to make HTTP requests in JavaScript.

For your convenience, I am adding source code here.

Coding Interview Questions | Skilled.dev

A full platform where I teach you everything you need to land your next job and the techniques to…

skilled.dev

1.2K 3

Sign up for Top Stories

By Level Up Coding

A monthly summary of the best stories shared in Level Up Coding Take a look.