CORS , Preflighted requests Explained

CORS - Cross-Origin Resource Sharing

Lets Explore in Detail.

srikanth tekmudi

@srikanth tekmudi

Why CORS - CORS allows a server to share resources with browsers having different origins.

Let's try to understand cross-origin requests via an example:

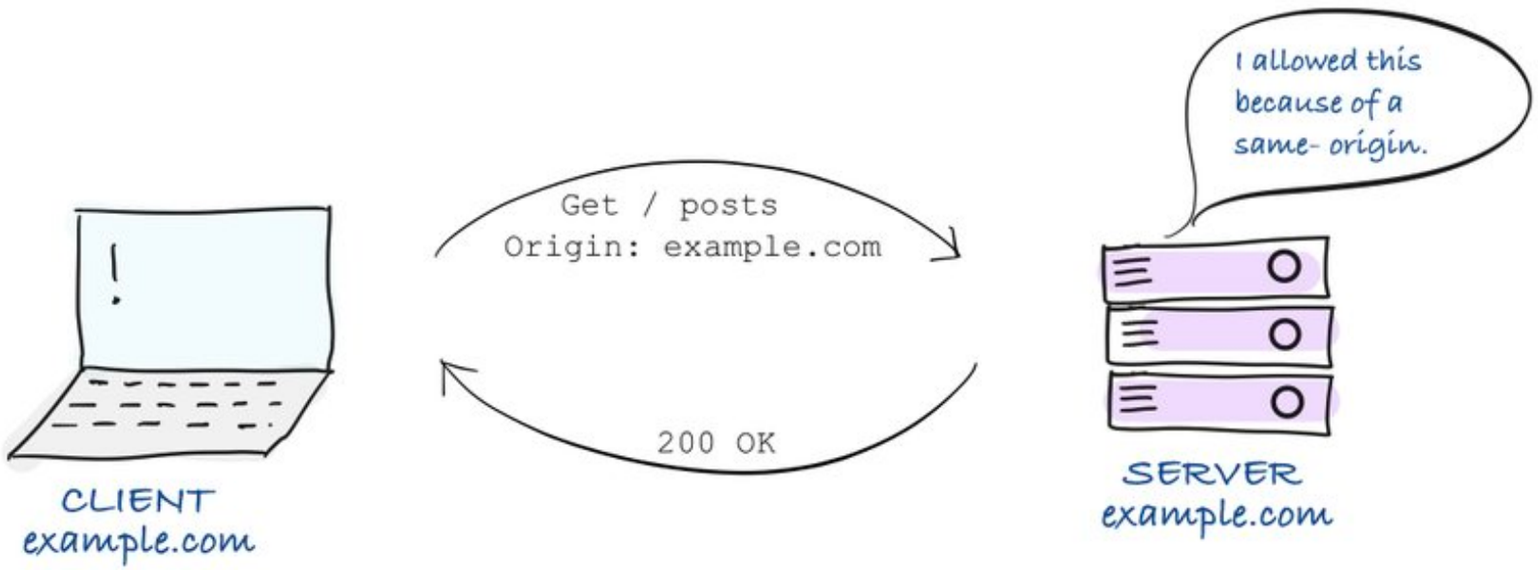
Suppose you're making an HTTP request from "a .com" to "b .com". That's a cross-origin request.

Often, you get an error while you're trying to request from a different URL.

This happens because browsers implement a same-origin policy for security reasons in which clients can make requests to the same origin without any errors.

srikanth tekmudi
@srikanth tekmudi

Same-Origin Policy



srikanth tekmudi
@srikanth tekmudi

📌 How CORS Works

The server adds the `Access-Control-Allow-Origin` header in the response, which must be the same as the `Origin` header of the request. If this is not the case, the browser will prevent the data from being shared with the client.

A few HTTP request methods cause side effects on the server, and these types of request methods must be pre-flighted.

Let's see what exactly Preflighted requests are

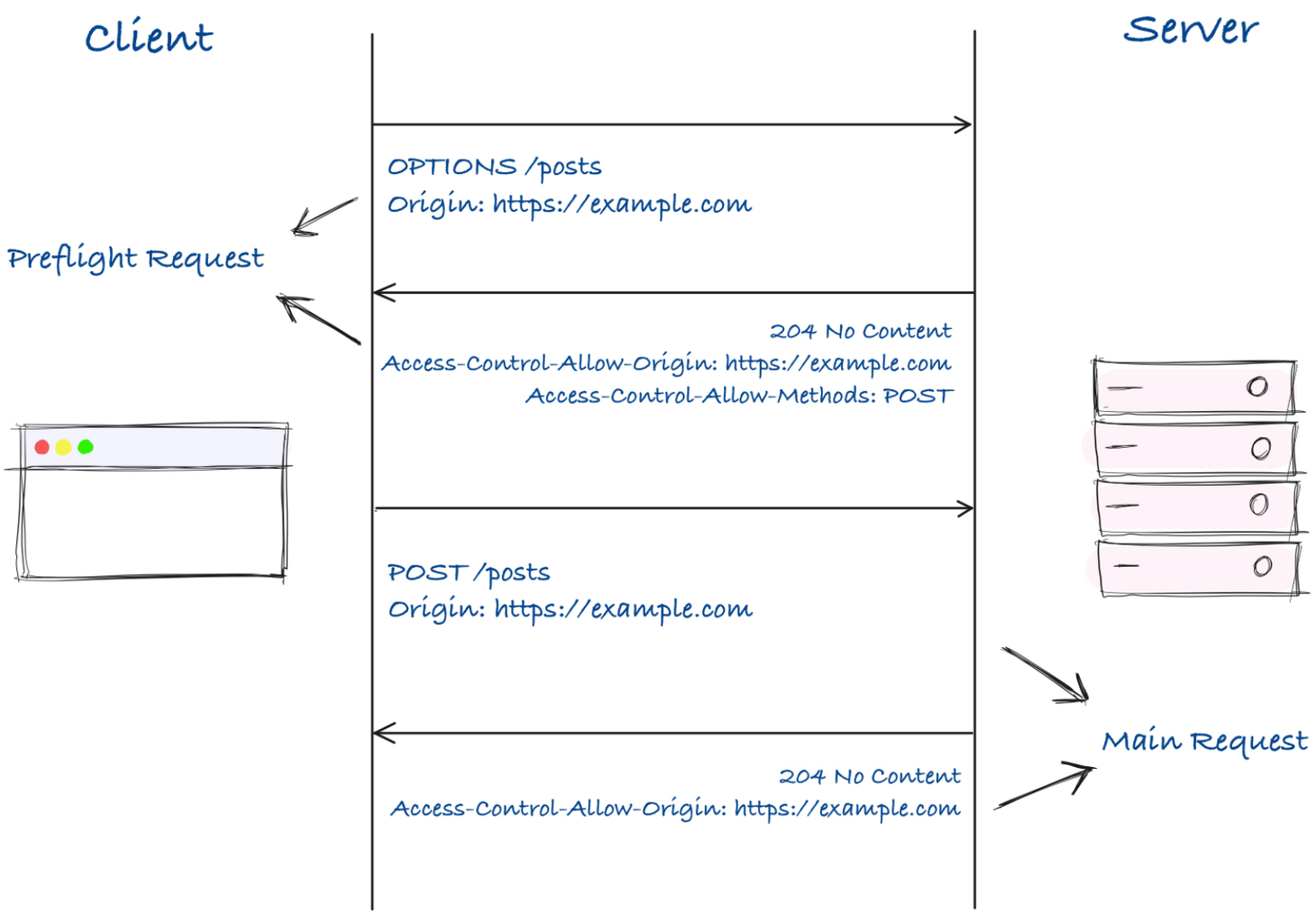
📌 Preflighted requests

The browser first sends the OPTIONS HTTP request to the server to ensure the actual request is safe to send.

In response, the server sends the `Access-Control-Allow-Methods` header with the allowed HTTP request methods values.

srikanth tekmudi
@srikanth tekmudi

Preflighted Request



srikanth tekmudi
@srikanth tekmudi