Don’t get lost in all of the details.

Before we had complex websites, the web was rather simple.

With the HTTP/1.0 protocol, we could not set custom headers and exotic HTTP methods such as DELETE or PUT. It was only after the year 2000 that the HTTP/1.1 spec introduced headers such as DELETE and PUT.

Until 2006, the **XMLHTTPRequest()** object was not fully formalized, and the **Fetch API** took another decade to develop.

Today, any request that is using a method that isn’t GET, POST, or HEAD or uses a Content-Type that isn’t text/plain, application/x-www-form-urlencoded, multipart/form-data—will trigger a preflight request.

This mechanism is meant to allow web servers to decide if they want to allow the actual request. The browser sets the `Access-Control-Request-Headers` and `Access-Control-Request-Method` headers to tell the server what request to expect and the server answers with corresponding headers.

Bottom line: In the old days, before CORS, the Single-Origin-Policy governed cross-site requests. This meant that servers assumed all cross-site requests would be blocked. This is why the preflight request was created, to make sure that modern request methods (such as DELETE or PUT) are allowed by the server before the actual request is sent.