

Response for preflight does not have HTTP ok status

Author : Tobias Hofmann

Date : November 26, 2018

Issuing an AJAX request is more complex than you might think.

Issue

When you issue an AJAX request to a server in another domain (CORS), you may get the following error message:

```
Response for preflight does not have HTTP ok status.
```

Problem

The server is configured to allow CORS. The Apache configuration includes

```
Header set Access-Control-Allow-Origin "*"
```

The response header of the service contains the correct header value. With this set you can access the service via CORS.

Solution

Now, why does it not work?

You have to be aware that this works for simple CORS requests. For more complex requests that set custom headers, etc, the service may not work. This is due to the preflight mechanism of the browser that checks if the service accepts the request. Before issuing an AJAX request (e.g. GET or POST), an OPTIONS is triggered to check what the service is accepting. You can see this in the network tab of Chrome.

The request includes two headers:

- Access-Control-Request-Headers
- Access-Control-Request-Method

It's full of stars!

Where documentation meets reality

Before issuing the actual GET request, the browser is checking if the service is correctly configured for CORS. This is done by checking if the service accepts the methods and headers going to be used by the actual request. Therefore, it is not enough to allow the service to be accessed from a different origin, but also the additional requisites must be fulfilled.

Solution

To configure Apache to set the headers, add the missing headers. Include in your HTTP service configuration the header set directive.

```
Header always set Access-Control-Allow-Origin "*"
Header always set Access-Control-Allow-Methods "POST, GET, OPTIONS, DELETE, PUT"
Header always set Access-Control-Allow-Headers "append,delete,entries,foreach,get,has,keys,set,values,Authorization"
```

Still not working!

After setting the values, you may still not able to call the service, as the browser still reports an error. The response code is not 2xx. Returning a 200 HTTP code can be enforced in Apache config using a rewrite rule.

```
RewriteEngine On
RewriteCond %{REQUEST_METHOD} OPTIONS
RewriteRule ^(.*)$ $1 [R=200,L]
```

With this configuration, the service will now work with CORS. The first OPTIONS request will pass:

The following GET request will also pass:

Links

- https://developer.mozilla.org/en-US/docs/Glossary/preflight_request
- <https://developer.mozilla.org/en-US/docs/Glossary/CORS>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Access-Control-Allow-Methods>
- https://httpd.apache.org/docs/2.4/mod/mod_rewrite.html#rewriterule