Cross-Domain Javascript

Solution - CORS (Cross-Origin-resource-sharing)

It is a mechanism that allows restricted resources (e.g. file) on a web page to be requested from another domain outside the domain from which the resource originated.

For example, a HTML page served from http://www.domain-a.com makes a <img> src request for http://www.domain-b.com.

Configuration:

Steps:

1. In your httpd.conf file, ensure that **LoadModule rewrite_module modules/mod_rewrite.so** is uncommented.
2. Add the following under **# Main server configuration section**

   #Cross Browser Domain

   # Always set these headers.
   Header always set Access-Control-Allow-Origin "*"
   Header always set Access-Control-Allow-Methods "POST, GET, OPTIONS, DELETE, PUT"
   Header always set Access-Control-Max-Age "1000"
   Header always set Access-Control-Allow-Headers "x-requested-with, Content-Type, origin, authorization, accept, client-security-token"

   # Added a rewrite to respond with a 200 SUCCESS on every OPTIONS request.
   RewriteEngine On
   RewriteCond %{REQUEST_METHOD} OPTIONS
   RewriteRule ^(.*)$ $1 [R=200,L]

Note:

- **Header always set Access-Control-Allow-Headers "x-requested-with, Content-Type, origin, authorization, accept, client-security-token"**
  This line of code determines what headers your requesting server (the one trying to make the remote call) is allowed to send.

- **Header always set Access-Control-Allow-Methods "POST, GET, OPTIONS, DELETE, PUT"**
  It determines what kind of RESTful calls your app is allowed to make. When using POST, we need OPTIONS too.

Resource:

https://awesometoast.com/cors/

https://benjaminhorn.io/code/setting-cors-cross-origin-resource-sharing-on-apache-with-correct-response-headers-allowing-everything-through/

Test as follows:


Headers
Preview
Response
Cookies
Timing
General
Request URL: http://dso.mars.localhost.dso.pentalog.fr/app/controller/mandat/strategieReversement/PeriodiciteReversementContr
ler.js?_dc=1567066575500
Request Method: GET
Status Code: 200 OK
Remote Address: 127.0.0.1:80
Referrer Policy: no-referrer-when-downgrade
Response Headers
view source
Accept-Ranges: bytes
Access-Control-Allow-Headers: x-requested-with, Content-Type, origin, authorization, accept, client-security-token
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
Access-Control-Allow-Origin: *
Access-Control-Max-Age: 1000
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Connection: Keep-Alive
Content-Length: 5773
Content-Security-Policy: script-src 'self' www.google.com/recaptcha/api.js www.gstatic.com/recaptcha/api2/v1565591531251/recapt
__en.js 'unsafe-inline' 'unsafe-eval'; object-src 'self';
Content-Type: application/javascript
Date: Thu, 29 Aug 2019 08:16:17 GMT
Expires: 0
Keep-Alive: timeout=5, max=89
Last-Modified: Thu, 22 Aug 2019 06:02:52 GMT
Pragma: no-cache