

PHP CORS

Lors d'un appel bios sur un service PHP il y a un test sur la compatibilité du nom de domaine. Si le nom de domaine de l'appelant est différent du nom de domaine du serveur interprétant le php il y aura une erreur du type:

```
Access to XMLHttpRequest at 'http://localhost:8000/postjson.php' from origin 'null' has been blocked by CORS policy: Request header field content-type is not allowed by Access-Control-Allow-Headers in preflight response. client2.html:1
POST http://localhost:8000/postjson.php net::ERR_FAILED xhr.js:247
Uncaught (in promise) q {message: 'Network Error', name: 'AxiosError', code: 'ERR_NETWORK', config: {...}, request: XMLHttpRequest, ...} xhr.js:154
```

Pour que php accepte une requête provenant d'un autre domaine modifier l'entête php ainsi

```
<?php
header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Headers: Content-Type, X-Auth-Token, Authorization, Origin');
header('Access-Control-Allow-Methods: GET, POST');
```

Ensuite pour accepter un payload (données envoyées par le POST de axios (ou autre) en brut il faut définir la reception dans php en mode brut :

```
<?php
// postjson.php

header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Headers: Content-Type, X-Auth-Token, Authorization, Origin');
header('Access-Control-Allow-Methods: GET, POST');
$inputJSON = file_get_contents('php://input');
$inputreçu = json_decode($inputJSON, TRUE); //convert JSON into array
print_r($inputreçu);

?>
```

La raison en est que php://input renvoie toutes les données brutes après les en-têtes HTTP de la requête, quel que soit le type de contenu.

Le superglobal PHP \$_POST, est seulement supposé envelopper les données qui sont soit

application/x-www-form-urlencoded (type de contenu standard pour les simples messages de formulaire <form>)

ou

multipart/form-data (principalement utilisé pour les téléchargements de fichiers).

En effet, ce sont les seuls types de contenu qui doivent être supportés par les agents utilisateurs. Le serveur et PHP ne s'attendent donc traditionnellement pas à recevoir d'autres types de contenu (ce qui ne veut pas dire qu'ils ne le pourraient pas).

Tester donc avec le programme php ci-dessus côté serveur (sous xampp/htdocs par exemple) et ce programme client lancé à partir de n'importe où (dml-click à partir du disque par exemple) :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.3/dist/
jquery.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/axios/dist/
axios.min.js"></script>
</head>
<body>

  <div id="container">

    <input type="text" id="nom" placeholder="nom">
    <input type="text" id="prenom" placeholder="prenom">

    <button>POST</button>

  </div>
  <script>
// écouter le bouton et faire axios.post au click

    $('button').click(function(e) {
      lenom = $("#nom").val()
      leprenom = $("#prenom").val()
      // récupérer les nom et prenom
      axios.post('http://localhost:8000/postjson.php',{
        nom:lenom,
        prenom:leprenom
      }).then(function(response)
        { console.log(response.data); })
    })

  </script>
</body>
</html>
```

