

EQUIPE BACON GRAYBEARS - SIMPLON 2022

STRATÉGIE DE CONCEPTION ET DE SÉCURISATION D'UNE APPLICATION



Authentification et mots de passe

Qu'est qu'une authentification ?

Il s'agit d'un processus de vérification de l'identité. Ce processus peut-être "simple" ou "multifacteurs".

L'authentification est un mécanisme faisant intervenir deux entités distinctes : un prouveur et un vérifieur :

- ✓ Le prouveur cherche à prouver son identité au vérifieur.
- ✓ Le vérifieur doit être capable de s'assurer de la validité de l'identité du prouveur.



Authentification simple ou multifacteurs

AUTHENTIFICATION SIMPLE :

Il s'agit d'une authentification basique avec le couple utilisateur/mot de passe. Le mot de passe peut-être chiffré de manière cryptographique, donnant lieu à une authentification dite "robuste".

AUTHENTIFICATION MULTIFACTEURS :

Authentification mettant en œuvre plusieurs facteurs d'authentification appartenant à des types différents.

Differents facteurs d'authentifications :

FACTEUR DE CONNAISSANCE :

Il s'agit d'une connaissance devant être mémorisée telle qu'une phrase de passe, un mot de passe, un code pin etc...

FACTEUR DE POSSESSION :

Il s'agit d'un élément secret non mémorisable contenu dans un objet physique qui idéalement protège cet élément de toute extraction, tel qu'une carte à puce, un token, un téléphone etc...

FACTEUR INHÉRENTS :

Il s'agit d'une caractéristique physique intrinsèquement liée à une personne et indissociable de la personne elle-même, telle qu'une caractéristique biologique (ADN), morphologique (empreinte digitale, empreinte rétinienne) etc...

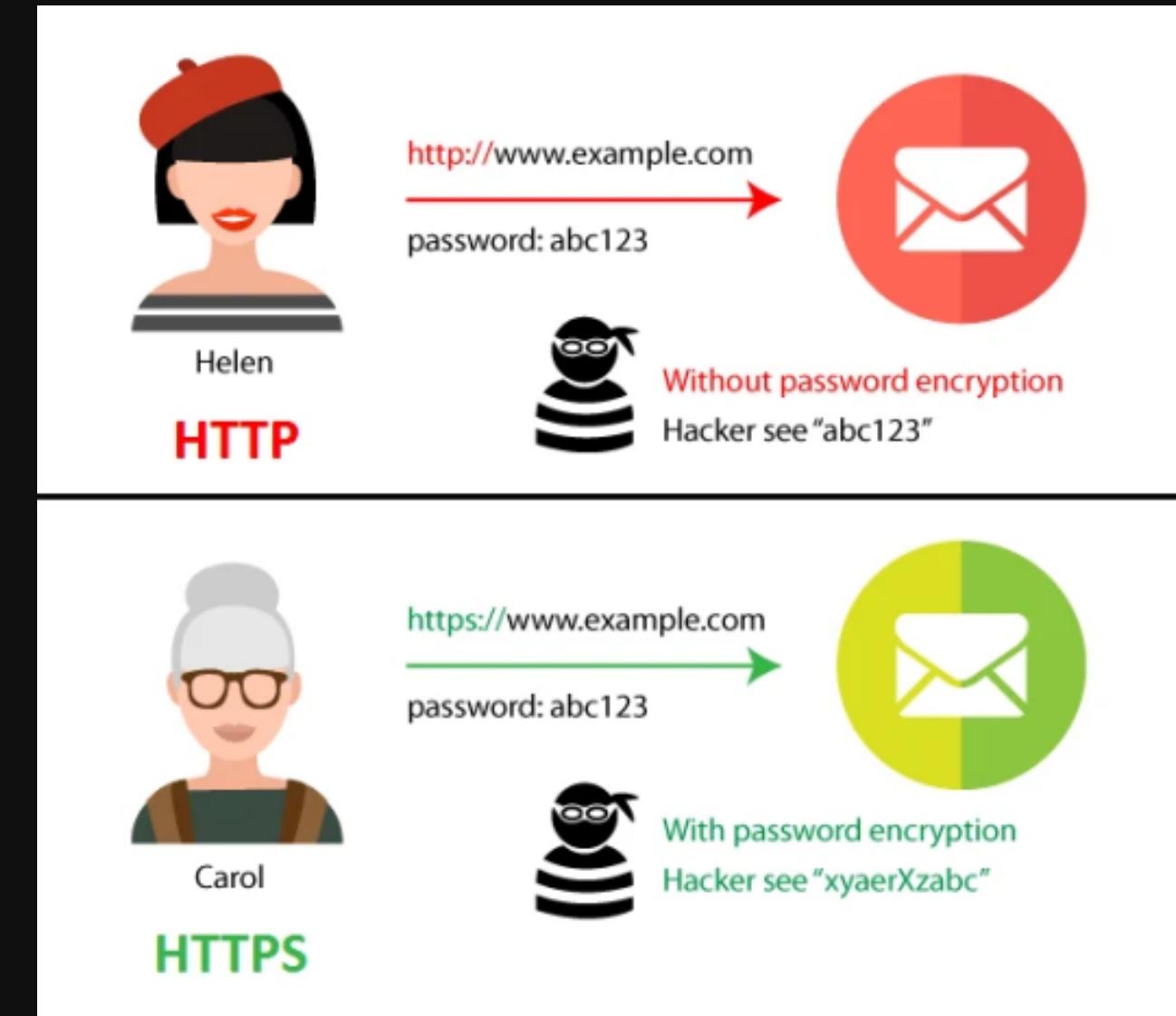
Quelques recommandations concernant l'authentification :

- ✓ Mener une analyse de risques lors de la mise en place de moyens d'authentification
- ✓ Privilégier une authentification multifacteurs.
- ✓ Privilégier une authentification sur des facteurs de possession.
- ✓ Adapter la robustesse des mots de passe à son contexte d'utilisation.
- ✓ Utiliser un coffre-fort de mots de passe.

HTTP(S) & CORS

HTTP ou HTTPS ?

- HTTP (Hypertext Transfer Protocol) désigne un protocole de communication entre un client et un serveur.
- HTTPS ou Hypertext Transfer Protocol Secure, il utilise HTTP et SSL qui permet de crypter et de décrypter des informations sensibles.



Méthodes

- GET - permet de recevoir des informations
- POST - soumet des données au serveur
- PUT - sauvegarde des modifications de données au serveur
- DELETE - supprime des données
- HEAD - reçoit des informations du header(en-tête) de la page

Les headers

Les headers contiennent des informations sur les requêtes et réponses HTTP

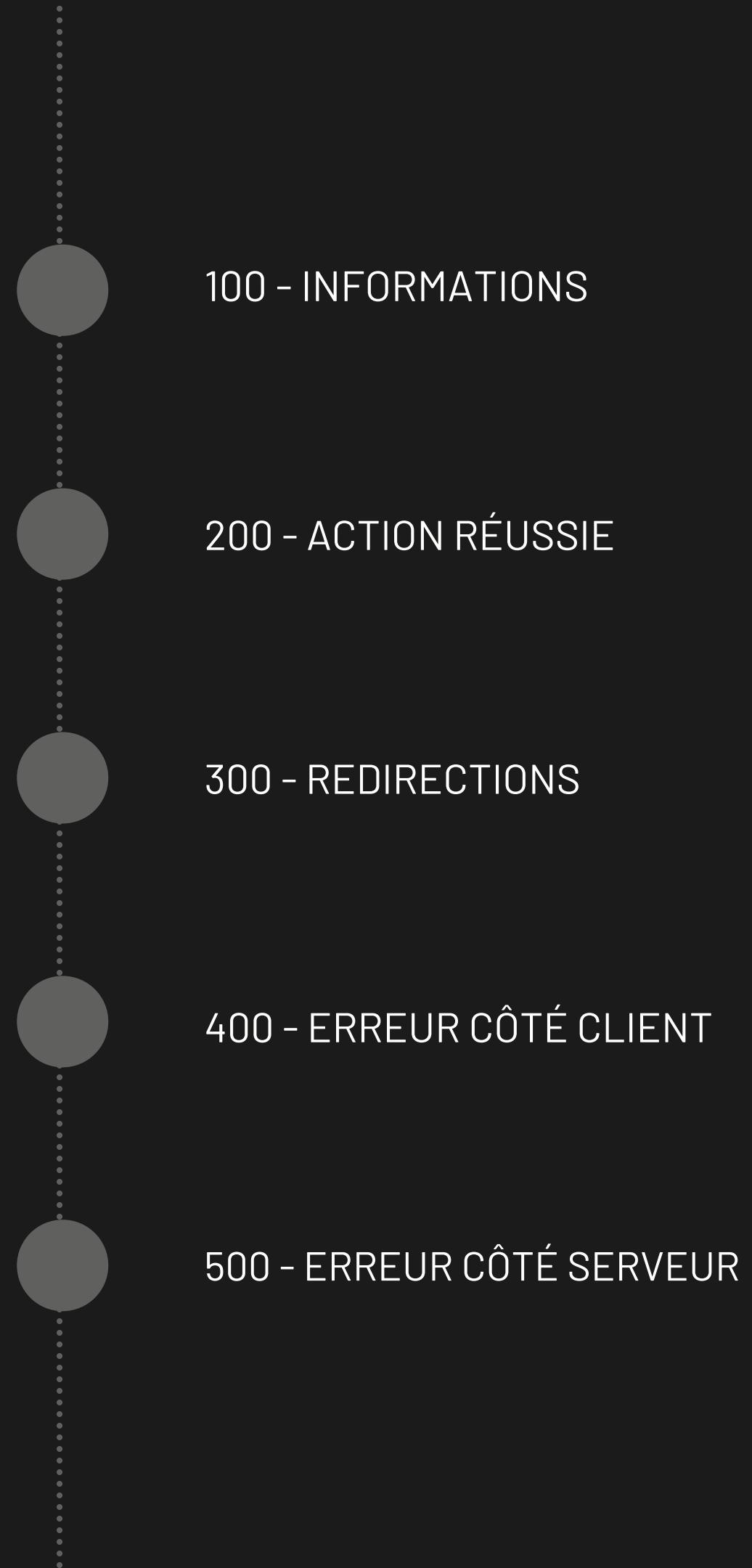
Request Header:

```
→ GET / HTTP/1.1  
→ Host: www.msaleh.co.cc  
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.10)  
Gecko/2009042316 Firefox/3.0.10  
Accept: */*  
Connection: close
```

Response Header:

```
→ HTTP/1.1 200 OK  
Date: Sat, 09 May 2009 12:27:54 GMT  
Server: Apache/2.2.11 (Unix)  
Last-Modified: Thu, 12 Feb 2009 15:29:42  
GMT  
Etag: "c3b-462ba63a46580"-gzip  
Cache-Control: max-age=1200, private,  
proxy-revalidate, must-revalidate ↴  
Expires: Sat, 09 May 2009 12:47:54 GMT  
Accept-Ranges: bytes  
Content-Length: 976  
Content-Type: text/html
```

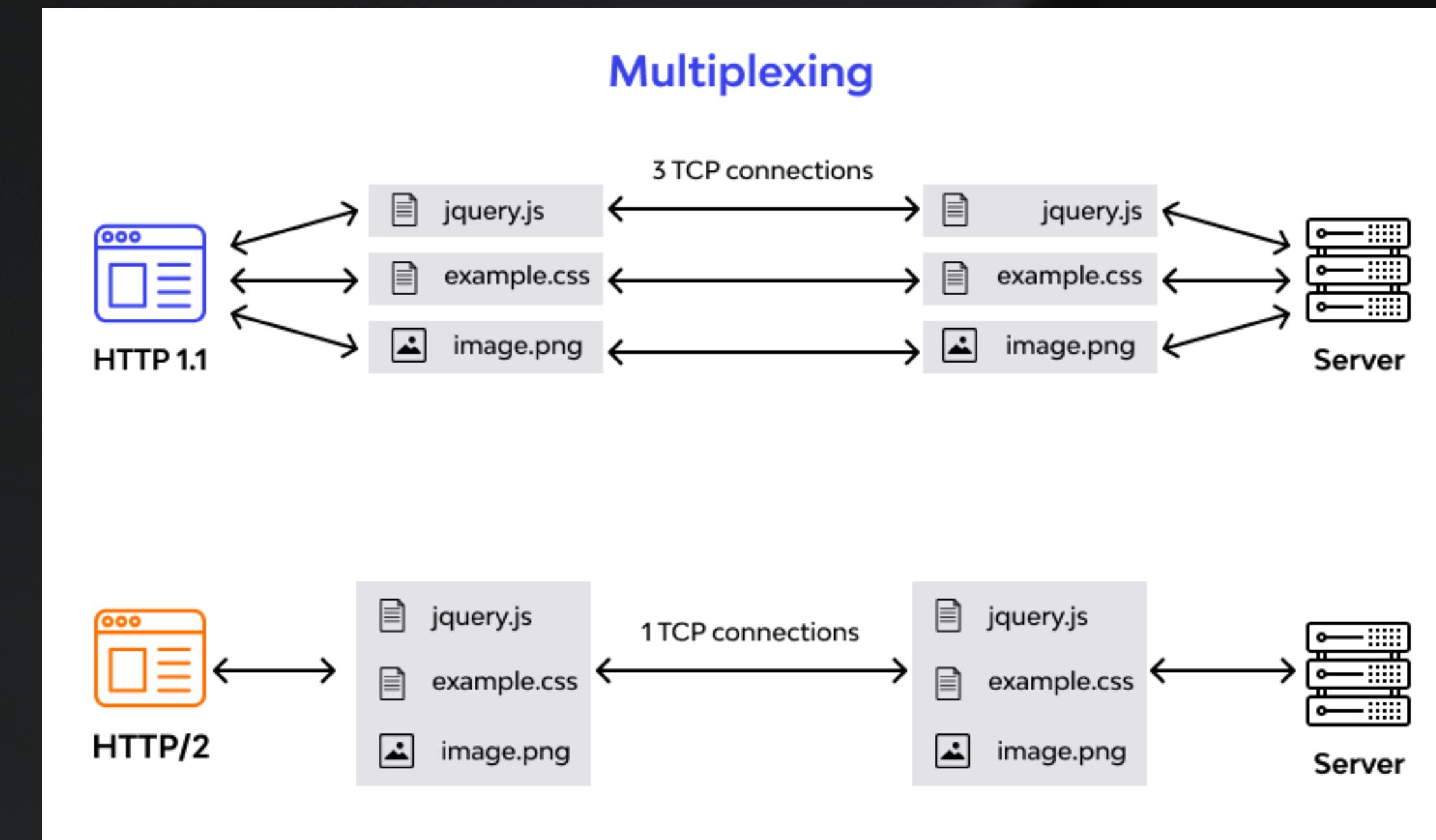
CODE STATUS



Différences entre HTTP 1.1 et HTTP/2

HTTP/2 permet d'effectuer plusieurs requêtes en même temps

HTTP/2 rend les requêtes et réponses 2.29x plus rapides qu'en utilisant HTTP 1.1



CORS

CORS ou Cross-origin Ressource Sharing est un mécanisme qui autorise un site sur une URL à faire une requête de donnée sur une autre URL.



```
▼ Response Headers (278 B)
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: http://localhost:5000
Vary: Origin
Content-Type: application/json; charset=utf-8
Content-Length: 23
ETag: W/"17-5bdkTC043EEL9K1bdRBu8k2HqAU"
Date: Mon, 29 Mar 2021 12:47:12 GMT
Connection: keep-alive

▼ Request Headers (341 B)
GET / HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
Accept: /*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:5000/
Origin: http://localhost:5000
```

Autorisation des CORS via Express.JS

```
const cors = require('cors');
💡
app.use(cors({ origin: 'https://foo.com' }))
```

Autorisation des CORS via PHP

```
1 <?php
2
3 header('Access-Control-Allow-Origin: *');
4 header('Content-Type: application/json; charset=UTF-8');
5 header('Access-Control-Allow-Methods: GET,POST,PUT,PATCH,DELETE,OPTIONS');
```