
SPRINT 1 - POP QUIZ

Tanafaat Abdessamad

SOMMAIRE

Chapitre 1: Challenges Backend	1
Chapitre 2: Challenges Frontend	2

Chapitre 1: Challenges Backend

Prérequis :

Vérifier PHP 8 : `php -v`

Vérifier Composer : `composer -v`

Vérifier Node.js 22 et NPM : `node -v` et `npm -v`

```
backend git:(main) * nvm use 22
Now using node v22.13.1 (npm v10.9.2)
backend git:(main) * node -v
v22.13.1
backend git:(main) * npm -v
10.9.2
backend git:(main) *
```

```
backend git:(main) * php -v
PHP 8.4.3 (cli) (built: Jan 15 2025 01:03:17) (NTS)
Copyright (c) The PHP Group
Built by Homebrew
Zend Engine v4.4.3, Copyright (c) Zend Technologies
with Zend OPcache v8.4.3, Copyright (c), by Zend Technologies
backend git:(main) * composer -v
Composer version 2.8.5 2025-01-21 15:23:40
```

```
backend git:(main) * php -v
PHP 8.4.3 (cli) (built: Jan 15 2025 01:03:17) (NTS)
Copyright (c) The PHP Group
Built by Homebrew
Zend Engine v4.4.3, Copyright (c) Zend Technologies
with Zend OPcache v8.4.3, Copyright (c), by Zend Technologies
backend git:(main) *
```

1. PHP 8 (avec serveur intégré)

- Rôle : Langage principal utilisé pour développer le backend.
- PHP permet de gérer les requêtes, exécuter la logique métier et interagir avec la base de données.
- Serveur intégré (`php -S`) : Permet de lancer le backend sans nécessiter Apache ou Nginx.

2. Composer

- Rôle : Gestionnaire de dépendances pour PHP.
- Installe automatiquement les bibliothèques et packages nécessaires au projet (ex: Laravel, Symfony, etc.).

3. Node.js 22 & NPM (via nvm)

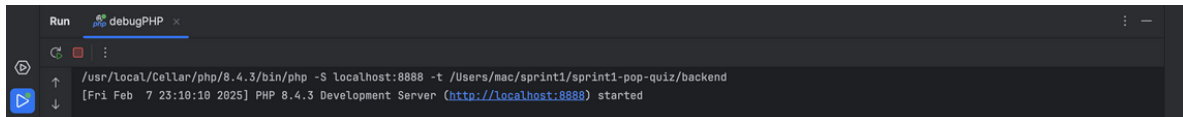
- Rôle : Plateforme permettant d'exécuter du JavaScript côté serveur et de gérer les outils frontend.
- Node.js : Utilisé pour exécuter des scripts et outils de développement (ex: Webpack, Babel, etc.).
- NPM (Node Package Manager) : Installe et gère les bibliothèques JavaScript nécessaires au projet.
- nvm (Node Version Manager) : Permet d'utiliser plusieurs versions de Node.js selon les besoins du projet.

Installation et exécution :

```
cd backend
composer install
php -S localhost:8888 2>/dev/null
```

- `cd backend` → Se positionner dans le dossier du backend.
- `composer install` → Installer les dépendances PHP.
- `php -S localhost:8888` → Lancer le serveur PHP intégré.
- cache les erreurs ou avertissements éventuels du serveur PHP pour une exécution plus propre.

On peut aussi utiliser **PHPStorm** pour lancer le serveur PHP.



Problème : Gestion des erreurs :

- **Problème 1:**

- `http://localhost:8888/broken` returns a 500 Internal Server Error with no error message.

- **Debugging Steps :**

Vérifier l'affichage des erreurs et voir PHP Logs :

```
13 error_reporting( error_level: E_ALL);
14 ini_set( option: 'log_errors', value: 1);
15 ini_set( option: 'error_log', value: '/tmp/sprint1-php-error.log');
16 ini_set( option: 'display_errors', value: 0);
```

```
tail -f /tmp/sprint1-php-error.log
```

tail ?

- Utile pour voir les dernières erreurs sans afficher tout le fichier.
- Pratique pour les fichiers volumineux.

-f ?

- Permet de surveiller en direct un fichier qui se met à jour (ex. logs PHP).
- Évite de relancer la commande à chaque erreur.

```
#2 /Users/mac/sprint1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/MiddlewareDispatcher.php(73): Slim\Routing\RouteRunner->handle(Object(Slim\Http\Request), Object(Slim\Http\Response))
#3 /Users/mac/sprint1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/MiddlewareDispatcher.php(73): Slim\Routing\RouteRunner->handle(Object(Slim\Http\Request), Object(Slim\Http\Response))
#4 /Users/mac/sprint1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/Routing/Route.php(321): Slim\Routing\RouteRunner->handle(Object(Slim\Http\Request), Object(Slim\Http\Response))
#5 /Users/mac/sprint1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/Routing/RouteRunner.php(74): Slim\Routing\RouteRunner->handle(Object(Slim\Http\Request), Object(Slim\Http\Response))
#6 /Users/mac/sprint1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/MiddlewareDispatcher.php(73): Slim\Routing\RouteRunner->handle(Object(Slim\Http\Request), Object(Slim\Http\Response))
#7 /Users/mac/sprint1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/App.php(209): Slim\Routing\RouteRunner->handle(Object(Slim\Http\Request), Object(Slim\Http\Response))
#8 /Users/mac/sprint1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/App.php(193): Slim\App->handle(Object(Slim\Http\Request), Object(Slim\Http\Response))
#9 /Users/mac/sprint1/sprint1-pop-quiz/backend/index.php(128): Slim\App->run()
#10 {main}
      thrown in /Users/mac/sprint1/sprint1-pop-quiz/backend/index.php on line 63
```

Problème : `write` n'est pas une méthode valide

```
39 $app->get( pattern: '/broken', function (Request $request, Response $response, $args) {
40     /** @disregard P1013 because we're just testing */
41     $response->getBody()->write("Hello world!");
42     return $response;
43 });
44
45 $app->get( pattern: '/crash', function (Request $request, Response $response, $args) {
46     logRequestWithRotation("Accessed crash endpoint.");
```

Solution :

```
61 $app->get( pattern: '/broken', function (Request $request, Response $response, $args) {
62     /** @disregard P1013 because we're just testing */
63     $response->getBody()->write( string: "Hello world!");
64     return $response;
65 });
```



Problème : Gestion des connexions simultanées :

Nous allons effectuer un test de charge avec Apache Benchmark (ab) pour identifier pourquoi la page <http://localhost:8888/crash> plante sous forte demande.

```
ab -n 200 -c 10 http://localhost:8888/crash
```

ab → Commande pour exécuter un test de performance avec Apache Benchmark.

-n 200 → Effectue 200 requêtes au total.

-c 10 → Envoie 10 requêtes simultanées à chaque instant.

<http://localhost:8888/crash> → L'URL cible du test.

Debugging Steps :

```
+ backend git:(main) * ab -n 200 -c 10 http://localhost:8888/crash
This is ApacheBench, Version 2.3 <$Revision: 1903618 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Finished 200 requests

Server Software:        localhost
Server Hostname:        localhost
Server Port:            8888

Document Path:          /crash
Document Length:        0 bytes

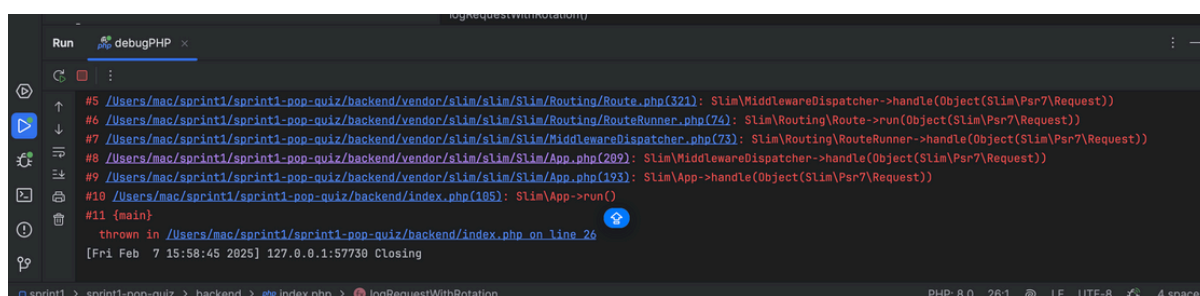
Concurrency Level:      10
Time taken for tests:    0.242 seconds
Complete requests:      200
Failed requests:         0
Non-2xx responses:      200
Total transferred:      71000 bytes
HTML transferred:       0 bytes
Requests per second:    826.98 [#/sec] (mean)
Time per request:       12.092 [ms] (mean)
Time per request:       1.209 [ms] (mean, across all concurrent requests)
Transfer rate:          286.70 [Kbytes/sec] received

Connection Times (ms)
      min     mean[+/-sd] median   max
Connect:    0       0  0.1-0.1      0     1
Processing:  2      11  3.9-10.0     10    24
Waiting:    2      11  3.9-10.0     10    24
Total:       2      12  3.9-10.0     10    24

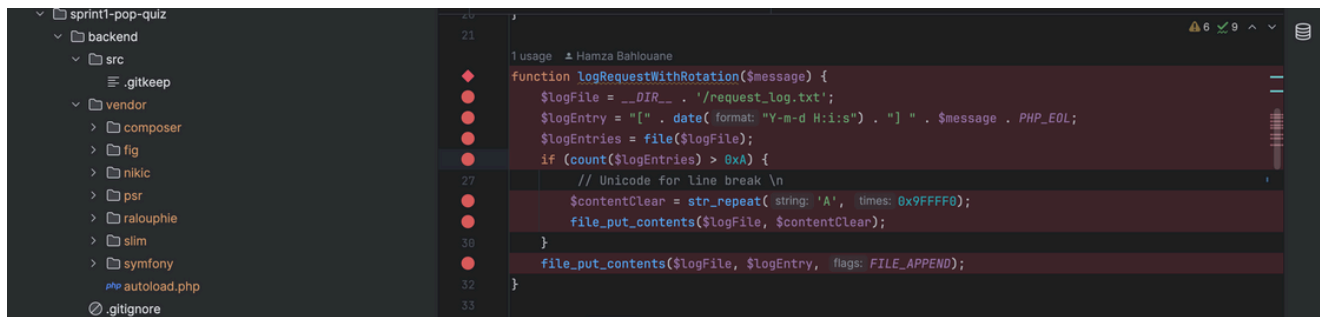
Percentage of the requests served within a certain time (ms)
 50%    10
 66%    11
 75%    12
 80%    12
 90%    18
 95%    23
 98%    23
 99%    23
100%    24 (longest request)
+ backend git:(main) *
```

test `ab -n 200 -c 10 http://localhost:8888/crash` montre 200 requêtes complètes sans erreurs .

Problème : quand j'accède à : <http://localhost:8888/crash> !



Pourquoi ? :

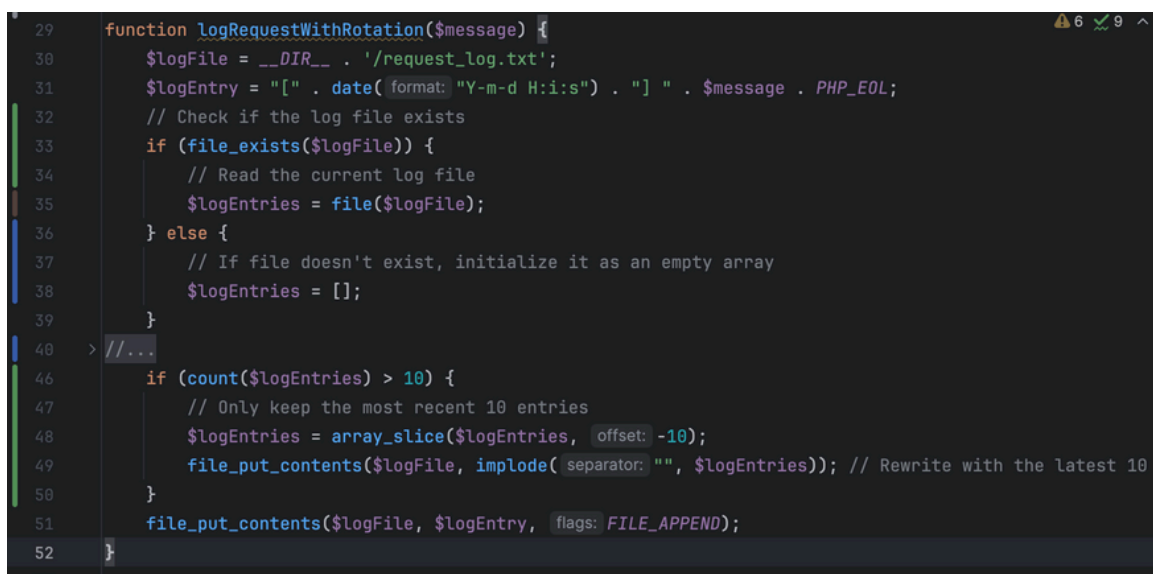


```
1 usage  Hamza Bahlouane
21
22 function logRequestWithRotation($message) {
23     $logFile = __DIR__ . '/request_log.txt';
24     $logEntry = "[" . date( format: "Y-m-d H:i:s") . " ] " . $message . PHP_EOL;
25     $logEntries = file($logFile);
26     if (count($logEntries) > 0xA) {
27         // Unicode for line break \n
28         $contentClear = str_repeat( string: 'A', times: 0x9FFFF0);
29         file_put_contents($logFile, $contentClear);
30     }
31     file_put_contents($logFile, $logEntry, flags: FILE_APPEND);
32 }
33
```

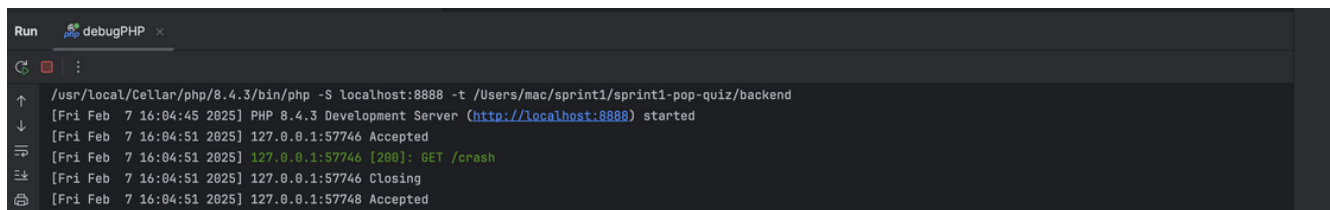
Problème :

La fonction tente de lire \$logFile via file(\$logFile), mais si le fichier n'existe pas encore, cela peut provoquer une erreur.

Solution :



```
29 function logRequestWithRotation($message) {
30     $logFile = __DIR__ . '/request_log.txt';
31     $logEntry = "[" . date( format: "Y-m-d H:i:s") . " ] " . $message . PHP_EOL;
32     // Check if the log file exists
33     if (file_exists($logFile)) {
34         // Read the current log file
35         $logEntries = file($logFile);
36     } else {
37         // If file doesn't exist, initialize it as an empty array
38         $logEntries = [];
39     }
40     // ...
41
42     if (count($logEntries) > 10) {
43         // Only keep the most recent 10 entries
44         $logEntries = array_slice($logEntries, offset: -10);
45         file_put_contents($logFile, implode( separator: " ", $logEntries)); // Rewrite with the latest 10
46     }
47     file_put_contents($logFile, $logEntry, flags: FILE_APPEND);
48 }
49
```



```
Run  debugPHP
/usr/local/Cellar/php/8.4.3/bin/php -S localhost:8888 -t /Users/mac/sprint1/sprint1-pop-quiz/backend
[Fri Feb 7 16:04:45 2025] PHP 8.4.3 Development Server (http://localhost:8888) started
[Fri Feb 7 16:04:51 2025] 127.0.0.1:57746 Accepted
[Fri Feb 7 16:04:51 2025] 127.0.0.1:57746 [200]: GET /crash
[Fri Feb 7 16:04:51 2025] 127.0.0.1:57746 Closing
[Fri Feb 7 16:04:51 2025] 127.0.0.1:57748 Accepted
```

Solution proposée :

assurant la création du fichier request_log.txt si nécessaire et en remettant en place la rotation des logs c'est dire m, La gestion de la rotation des logs (effacement après 10 entrées) en effet , Si plus de 10 entrées, garder uniquement les 10 dernières

Chapitre 2: Challenges Frontend

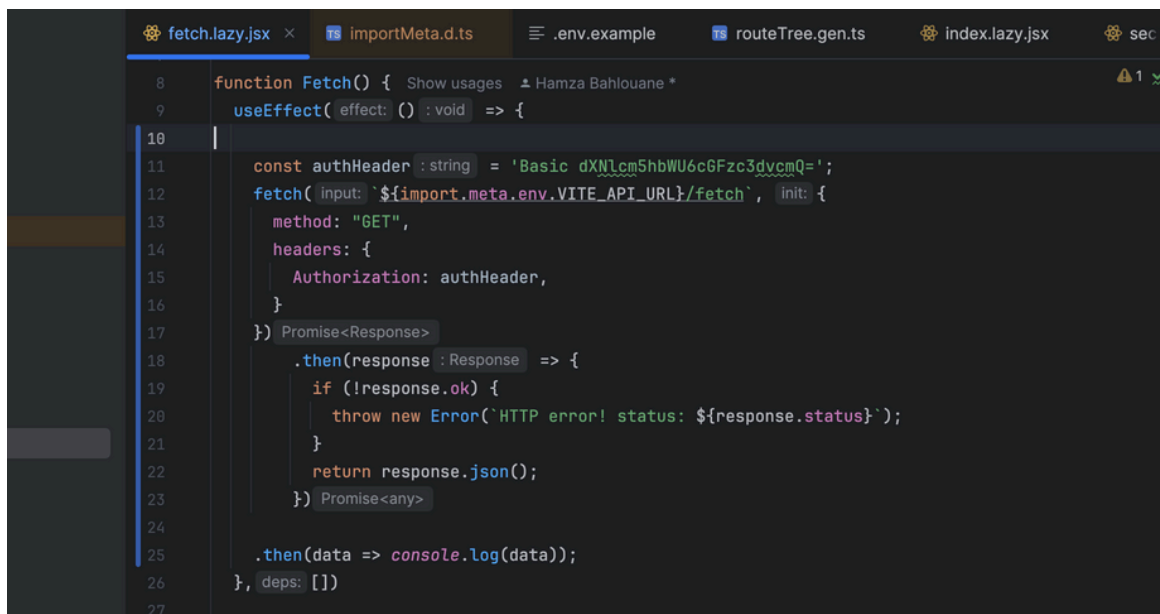
1 - Analyse du problème : Les appels Fetch / XHR échouent sur /fetch

Problèmes trouvés :

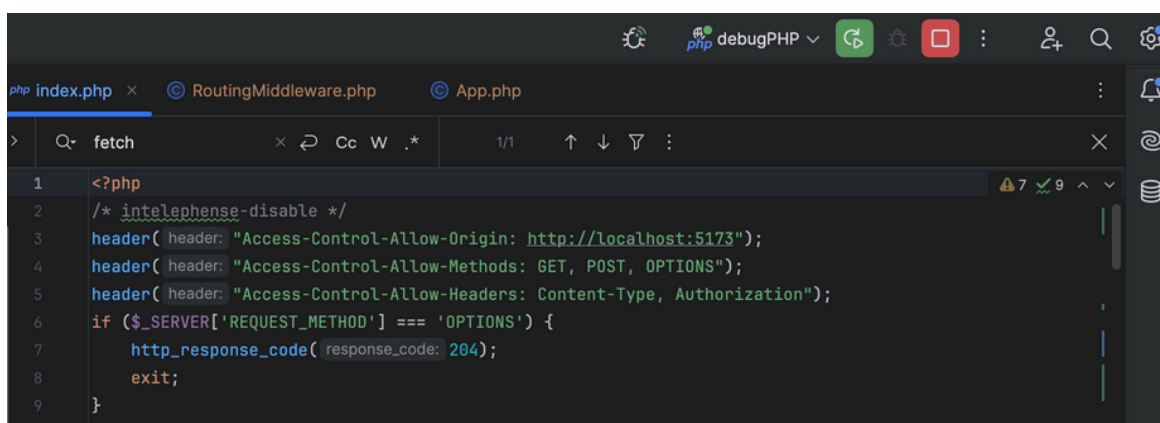
- Les appels Fetch ou XHR (XMLHttpRequest) échouent sur la route /fetch. La cause probable est liée à CORS (Cross-Origin Resource Sharing), un mécanisme de sécurité qui restreint les ressources d'un serveur pour des domaines autres que celui auquel l'utilisateur est actuellement connecté.
- Le backend nécessite une authentification (par exemple avec un token ou des cookies), il est possible que l'appel XHR soit rejeté en raison de l'absence de ce jeton d'authentification dans la requête ou d'un problème de cookies non envoyés dans une requête cross-origin.

Solution :

- **CORS** : l'ajout des en-têtes Access-Control-Allow-Origin, Access-Control-Allow-Methods, et Access-Control-Allow-Headers côté backend pour autoriser les requêtes cross-origin.
- **Authentification** : utilisation un jeton d'authentification pour assurer que on envoi correctement ces informations dans les requêtes Fetch et que le backend les accepte.

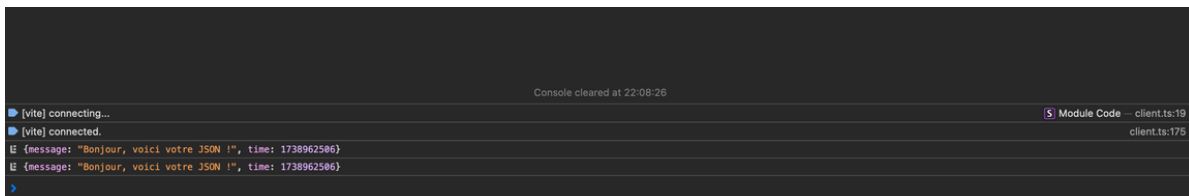


```
8 function Fetch() { Show usages Hamza Bahlouane *
9   useEffect( effect: () : void => {
10
11     const authHeader : string = 'Basic dXNlcm5hbWU6cGFzc3dvcmQ=';
12     fetch( input: `${import.meta.env.VITE_API_URL}/fetch`, init: {
13       method: "GET",
14       headers: {
15         Authorization: authHeader,
16       }
17     }) Promise<Response>
18       .then(response : Response => {
19         if (!response.ok) {
20           throw new Error(`HTTP error! status: ${response.status}`);
21         }
22         return response.json();
23       }) Promise<any>
24
25       .then(data => console.log(data));
26     }, deps: [])
27 }
```



```
1 <?php
2 /* intelephense-disable */
3 header( header: "Access-Control-Allow-Origin: http://localhost:5173");
4 header( header: "Access-Control-Allow-Methods: GET, POST, OPTIONS");
5 header( header: "Access-Control-Allow-Headers: Content-Type, Authorization");
6 if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') {
7     http_response_code( response_code: 204);
8     exit;
9 }
```

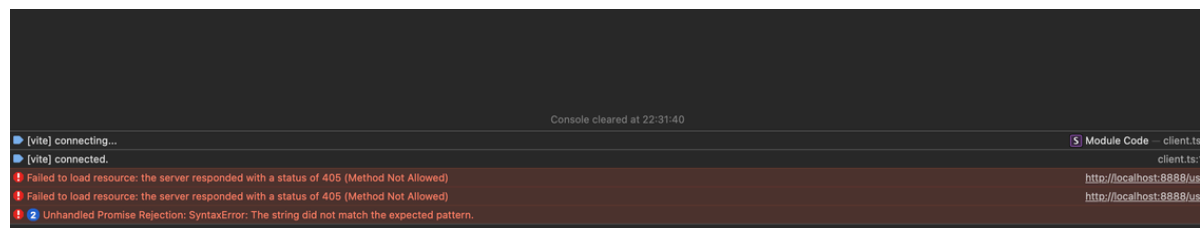
Problème résolu :



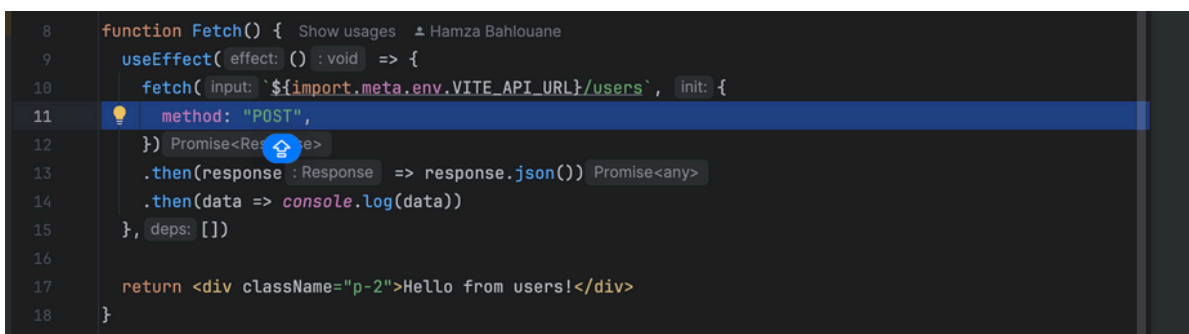
```
[vite] connecting...
[vite] connected.
E {message: "Bonjour, voici votre JSON !", time: 1738962586}
E {message: "Bonjour, voici votre JSON !", time: 1738962586}
```

2. Problème des appels XHR sur la Page /users :

D'après les erreurs et le code que vous avez partagé, le problème principal est lié au fait que la requête GET sur /usersr envoie une erreur 405 (Method Not Allowed). Analysons les causes possibles et les solutions.



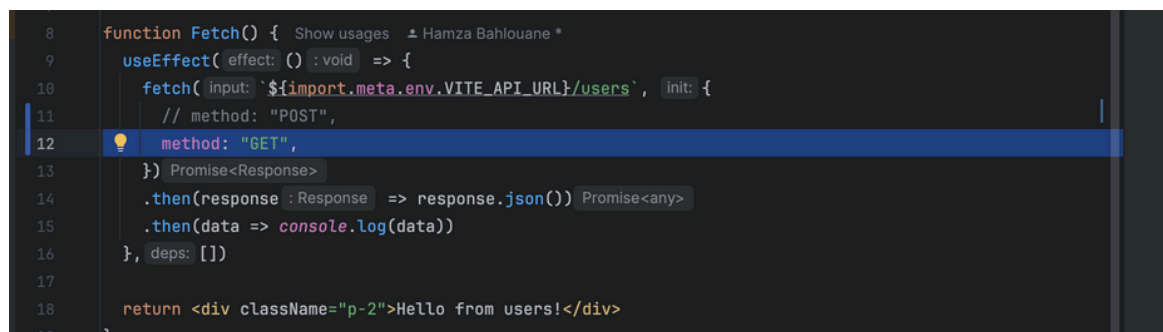
```
[vite] connecting...
[vite] connected.
Failed to load resource: the server responded with a status of 405 (Method Not Allowed)
Failed to load resource: the server responded with a status of 405 (Method Not Allowed)
Unhandled Promise Rejection: SyntaxError: The string did not match the expected pattern.
```



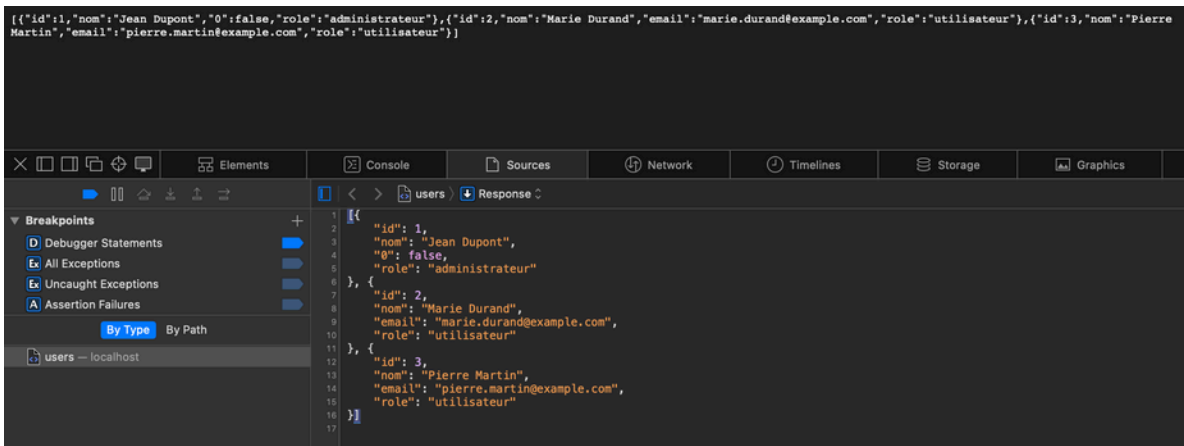
```
8 function Fetch() { Show usages Hamza Bahlouane
9   useEffect( effect: () : void => {
10     fetch( input: `${import.meta.env.VITE_API_URL}/users`, init: {
11       method: "POST",
12     }) Promise<Response>
13     .then(response : Response => response.json()) Promise<any>
14     .then(data => console.log(data))
15   }, deps: [])
16
17   return <div className="p-2">Hello from users!</div>
18 }
```

Solution :

- Vérification de CORS (Comme le problème de /fetch).
- Modifier le code cote frontend par utiliser la Méthode GET et pas POST .



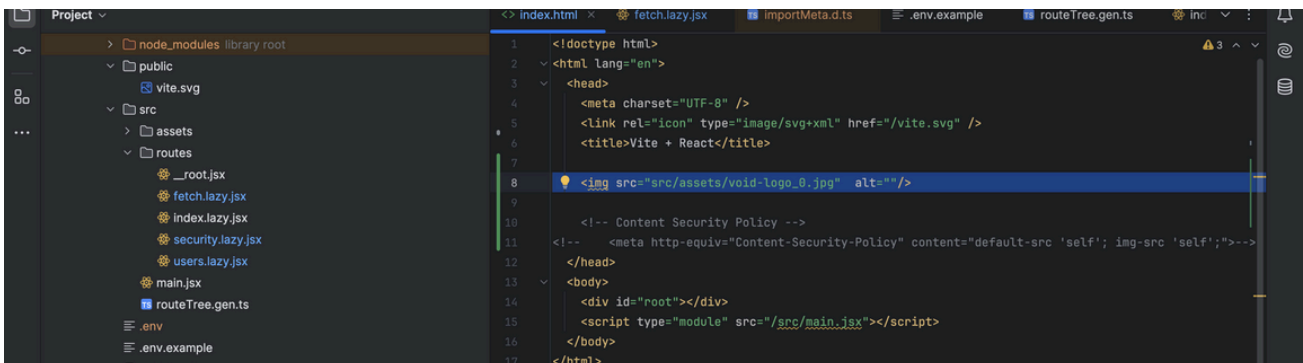
```
8 function Fetch() { Show usages Hamza Bahlouane *
9   useEffect( effect: () : void => {
10     fetch( input: `${import.meta.env.VITE_API_URL}/users`, init: {
11       // method: "POST",
12       method: "GET",
13     }) Promise<Response>
14     .then(response : Response => response.json()) Promise<any>
15     .then(data => console.log(data))
16   }, deps: [])
17
18   return <div className="p-2">Hello from users!</div>
19 }
```

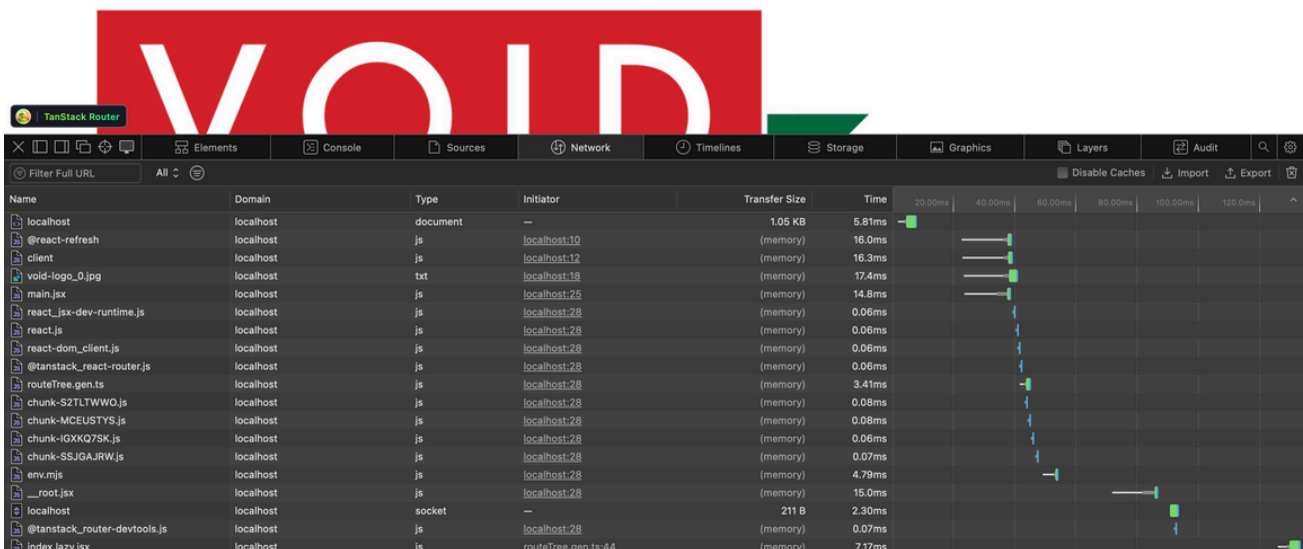
3. Optimisation du téléchargement des Assets :

J'ai ajouter une photo dans le projet pour voir le comportement .

Avant de mettre en place la mise en cache, on peut vérifier si les fichiers sont téléchargés à chaque recharge de la page. Cela permettra de confirmer qu'il n'y a pas de cache appliqué avant l'optimisation.



l'image est déjà mise en cache dès le premier accès ! même sans optimisation explicite. Cela signifie que le navigateur utilise déjà une forme de cache par défaut pour cette image !



Pourquoi une image peut être mise en cache par défaut :

1.Cache par défaut du navigateur :

- Les navigateurs modernes, y compris Safari, mettent souvent en cache certains types de ressources comme les images, CSS, et JavaScript si elles ne sont pas explicitement configurées pour ne pas être mises en cache.
- Si l'image est utilisée dans le code HTML, elle peut être stockée dans le cache automatiquement pour être réutilisée lors des prochaines visites, sans avoir besoin de configurations supplémentaires.

2.Durée de vie du cache (Cache-Control) :

- Les images peuvent avoir une durée de vie longue définie par les headers HTTP, ce qui permet aux navigateurs de les conserver dans le cache pendant un certain temps. Si tu as observé que l'image est déjà présente dans le cache dès le premier accès, cela pourrait être dû à des valeurs par défaut appliquées par le navigateur.

4. Correction de la Faille XSS sur la Page /security :

Problème :

- La page /security charge une image depuis Unsplash, qui est une source externe. Cela pose un risque de sécurité car : XSS via image malveillante : Si la source de l'image est compromise, un attaquant pourrait injecter du JavaScript via une image trafiquée.
- Violation des règles de sécurité : on souhaite autoriser uniquement les images provenant de localhost.

```
6
7 function Security() { Show usages Hamza Bahlouane
8   return 
3 <head>
4   <meta charset="UTF-8" />
5   <meta http-equiv="Content-Security-Policy" content="default-src 'self'; img-src 'self' ">
6   <link rel="icon" type="image/svg+xml" href="/vite.svg" />
7   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8   <title>Vite + React</title>
9 </head>
10 <body>
11   <div id="root"></div>
12   <script type="module" src="/src/main.jsx"></script>
13 </body>
14 </html>
15
```

Console cleared at 23:35:26

[vite] connecting...

[vite] connected.

Refused to load https://images.unsplash.com/photo-1711950901044-fa6215a9c59b?q=80&w=2970&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVudDB8fHx8fA%3D%3D because it does not appear in the img-src directive of the Content Security Policy.

Refused to apply a stylesheet because its hash, its nonce, or 'unsafe-inline' appears in neither the style-src directive nor the default-src directive of the Content Security Policy.

Module Code — client.ts:19
client.ts:175
security:1

[Home](#) [Fetch](#) [Users](#) [Security](#)

TANSTACK
React Router v1
Router8 items
state10 items
isLoading: false
isTransitioning: false
status: 'pending'
resolvedLocation6 items
location6 items
matches2 items
pendingMatches:
cachedMatches: []
statusCode: 200
redirect:

Elements Console Sources Network Timelines Storage Graphics Layers Audit

Search < > All Evaluations Errors Warnings Logs Debugs

Emulate User Gesture

Console cleared at 23:35:26

[vite] connecting...

[vite] connected.

Refused to load https://images.unsplash.com/photo-1711950901044-fa6215a9c59b?q=80&w=2970&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVudDB8fHx8fA%3D%3D because it does not appear in the img-src directive of the Content Security Policy.

Refused to apply a stylesheet because its hash, its nonce, or 'unsafe-inline' appears in neither the style-src directive nor the default-src directive of the Content Security Policy.

Module Code — client.ts:19
client.ts:175
security:1