

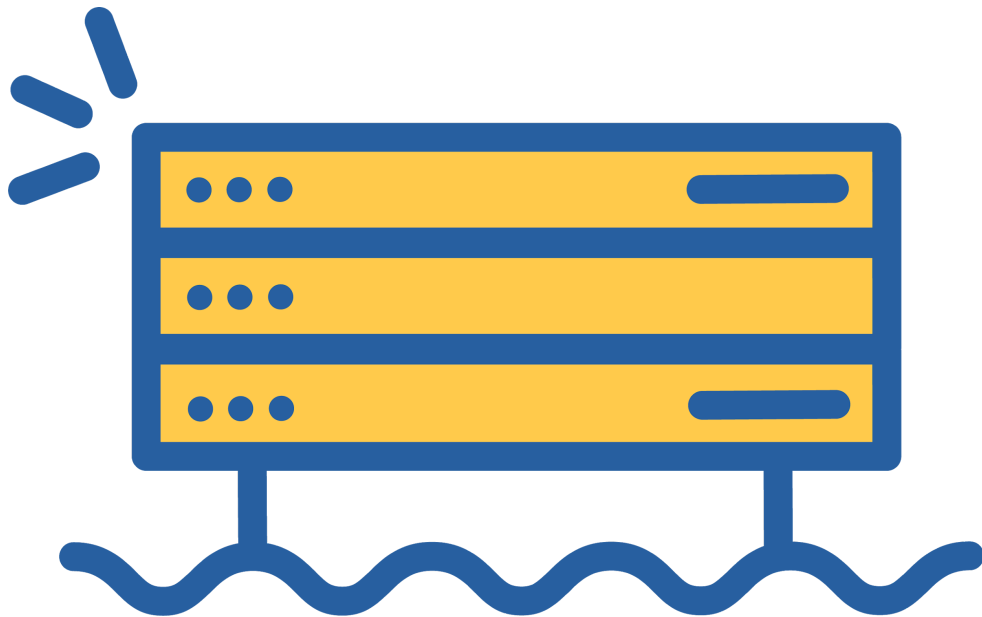
---

La Plateforme\_

# Jour 4 - ~~AJAX~~ Fetch

OLD Asynchronous JavaScript and XML

---



**La Plateforme\_**

A decorative graphic at the bottom right of the page, consisting of several blue geometric shapes: a triangle, a square, and another triangle, all in different shades of blue, arranged in a way that suggests a modern, architectural design.

---

## Job 00

Dans le répertoire de rendu “runtrack3”, créez un dossier “jour04”. Pour chaque job, créez un dossier “jobXX” où XX est le numéro du job. N’oubliez pas d’envoyer vos modifications dès qu’un job est avancé ou terminé et mettez des commentaires explicites.

Pour chacun des jobs, vous devez rendre un fichier index.php et un fichier script.js. Votre fichier index.php doit contenir les balises html de base et inclure votre script.js. Vos fichiers doivent contenir uniquement ce qui est spécifié dans les jobs.

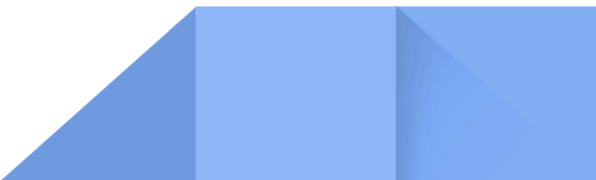
Vous voilà maintenant informés des différences entre javascript et jquery, si bien dans l’utilisation que dans les fonctionnalités. Il y a cependant une facette très importante que vous n’avez pas vu et que nous allons aborder aujourd’hui : l’asynchrone et plus précisément ~~AJAX~~ Fetch.

## Job 01

Créez un bouton ayant comme id “button”. Créez un fichier expression.txt contenant votre expression favorite. Lorsqu’un utilisateur clique sur le bouton, à l’aide de Fetch, récupérez le contenu du fichier expression.txt, placez le dans un paragraphe et insérez le dans le corps de votre page.

## Job 02

Créez une fonction javascript “jsonValueKey()” qui prend en paramètre une chaîne de caractères au format json et une clé. Cette fonction retourne la valeur liée à cette clé dans la chaîne de caractères. Par exemple si la string passée en paramètre est “{name: “La Plateforme\_”, address: “8 rue



---

d'hozier", city: "Marseille", nb\_staff: "11", creation:"2019"}" et la clé est "city", la fonction retourne "Marseille".

## Job 03

Téléchargez le fichier suivant :

<https://drive.google.com/open?id=1hiOzUjgzH9iurgYPpiZ-3gyMAzLiy7W>

Créez un formulaire permettant de trier ces données. Il doit contenir les champs suivant : id (input type text), nom (input type text), type (liste déroulante) et filtrer (input type button). Lorsque l'on clique sur "filtrer", le script doit à l'aide de Fetch, récupérer le contenu du fichier et lister les éléments répondant aux critères sélectionnés en les affichant sur une page HTML.

## Job 04

Créez une base de données "utilisateurs" contenant une table "utilisateurs" et ayant comme champs "id", "nom", "prenom" et "email". Ajoutez des utilisateurs à l'aide de phpmyadmin. Créez une page users.php qui se connecte à la base de données, récupère l'ensemble des utilisateurs et affiche ces informations au format json.

Dans votre page index.php, créez un tableau permettant de contenir ces informations ainsi qu'un bouton "update". Lorsque l'on clique sur ce bouton, le tableau doit se mettre à jour et contenir l'ensemble des informations des utilisateurs présents dans la base de données.

Vous pouvez tester votre code en ajoutant/supprimant des utilisateurs à l'aide de phpmyadmin entre deux clics.

---

## Rendu

Le projet est à rendre sur <https://github.com/prenom-nom/runtrack3>.

Dossiers "jourXX" -> "jobXX". Fichiers nommés index.php et script.js

Pensez à donner les droits sur le répertoire à deepthoughtlaplateforme !

## Base de connaissances

[https://fr.wikipedia.org/wiki/JavaScript\\_Object\\_Notation](https://fr.wikipedia.org/wiki/JavaScript_Object_Notation)

<https://json.org/json-fr.html>

[https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)

[https://www.w3schools.com/php/php\\_json.asp](https://www.w3schools.com/php/php_json.asp)

<https://www.pierre-giraud.com/javascript-apprendre-coder-cours/api-fetch/>

[https://developer.mozilla.org/fr/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/fr/docs/Web/API/Fetch_API/Using_Fetch)

