

Jour 4 - Fetch

Ce sujet est un hommage au grand Lucas Fetch !

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, aussi 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 **<button>** 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 <p>** 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 : [pokemon.json](#)

Créez un **formulaire** permettant de **trier** ces données.

Il doit contenir les **champs** suivants :

- **id** (input type text),
- **nom** (input type text),
- **type** (liste déroulante <select>)
- **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 directement dans **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 <table>** permettant de contenir ces informations ainsi qu'un **<button> "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 **deeptoughtlaplateforme** !

Compétences visées

- Utiliser la méthode fetch pour communiquer avec la base de donnée
- Comprendre et manipuler un JSON

Base de connaissances

[JavaScript Object Notation – Wikipédia](#)

Qu'est-ce qu'un JSON : définition

[Présentation de JSON](#)

[JSON Introduction](#)

[PHP and JSON](#)

[Présentation et utilisation de l'API Fetch en Javascript - Pierre Giraud](#)

[Utiliser Fetch - Référence Web API | MDN](#)