

# Jour 4 – Fetch

Description du sujet

## Job 00

---

Dans le répertoire de rendu **“runtrackJs”**, 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.html** et un fichier **script.js**. Votre fichier index.html 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 vue 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 :

```
{
  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 un fichier JSON nommé **utilisateur.json** contenant des informations utilisateurs. Chaque utilisateur possède un **id**, un **nom**, un **prénom** et un **email**.

Ajouter des utilisateurs dans votre fichier.

Dans votre page **index.html**, créez un **tableau <table>** permettant de contenir les informations des utilisateurs 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 le fichier.

Vous pouvez **tester votre code** en ajoutant/supprimant des utilisateurs.



## Rendu

---

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

Dossiers "jourXX" -> "jobXX".

Fichiers nommés **index.html** et **script.js**

Pensez à mettre votre repository en public !

## Compétences visées

---

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

## Base de connaissances

---

- [Qu'est-ce qu'un JSON : définition](#)
- [Présentation de JSON](#)
- [JSON Introduction](#)
- [Présentation et utilisation de l'API Fetch en Javascript - Pierre Giraud](#)
- [Utiliser Fetch - Référence Web API | MDN](#)