

Exercice : Myuka

Instructions

A l'aide des différents exercices que vous avez réalisé sur le JSON, réalisez le TP sur l'application Myuka ci-dessous.

- Fournissez la version de votre projet.
- Commentez et indentez votre code.
- Nommez vos variables de façon parlante.
- Vos méthodes (fonctions) ne doivent réaliser qu'une action à la fois.

Cahier des charges

MYUKA est une start-up de la région parisienne. Elle vous demande de lui concevoir une fiche produit qui sera traduite en page Web par la suite.

Elle souhaite donc une page Web contenant tout en haut un menu avec le titre «**Myuka**» ainsi qu'un logo.

Juste en dessous, un premier bloc prenant toute la largeur de la page et contenant un champ de formulaire, ainsi qu'un bouton.

Les **éléments** doivent être **centrés**.

Un autre bloc en dessous, prenant toute la largeur également, qui contiendra le **titre du produit**, en dessous son **code barre** et **information sur la marque**.

Toujours en dessous, deux séries de deux blocs :

La première série contient sur la gauche :la **photo**.

Sur le bloc de droite : les **ingrédients** contenus dans le produit.

La deuxième série de blocs contient des **valeurs nutritives**, sous forme de tableau et dans l'autre, des infos sur le **conditionnement**, **fabriquant**, etc...

En dessous, une série de trois blocs contenant des informations sous forme d'image.

Ces images concernent le **nutriscore**, **code nova**, **avec** ou **sans huile de palme**.

La start-up vous invite à ajouter autant de blocs que vous juger utile d'utiliser tant que cela reste harmonieux.

Elle vous autorise également à ajouter des données que vous trouvez pertinentes sur le produit.

Pour finir, deux séries de blocs faisant toute la largeur de la page, ainsi qu'un dernier bloc (Footer) pour les **crédits** et **copyright** de la page (pensez à la RGPD....).

Maquette

Le client vous fournit un semblant de maquette pour vous aider dans votre réalisation :

MY YUKA

Obtenez les informations sur les produits alimentaires

Scanner

Photo

Ingrédient

Allergènes

NOVA

NUTRISCORE

Huile de Palme

Info distributeur / fabricant / pays

Footer

Maquette Myuka

Charte graphique

Le client demande également à ce que les limites des blocs soient visibles.

La page doit contenir maximum trois couleurs pour le thème.

Un arrière plan peut être ajouté sur le thème de la cuisine.

La start-up insiste sur le fait que la page doit s'adapter à tous les écrans.

Néanmoins, la start-up vous impose une des trois couleurs : #87bb34.

Sources des données

Bien que la société ne dispose pas de base de données à proprement parler, elle vous fournit une source de données.

L'application Web est basée sur une Api REST diffusée par le site [OpenFoodFact](#).

<https://fr.openfoodfacts.org/>

Le site recense des données sur les produits alimentaires commercialisés principalement en France.

Votre application consommera donc un Webservice.

Voici à titre d'exemple une URL type :

<https://world.openfoodfacts.org/api/v2/product/737628064502.json>

Le code barre étant : 737628064502

[illegible]

Vous avez donc accès aux données en appelant cette Api.

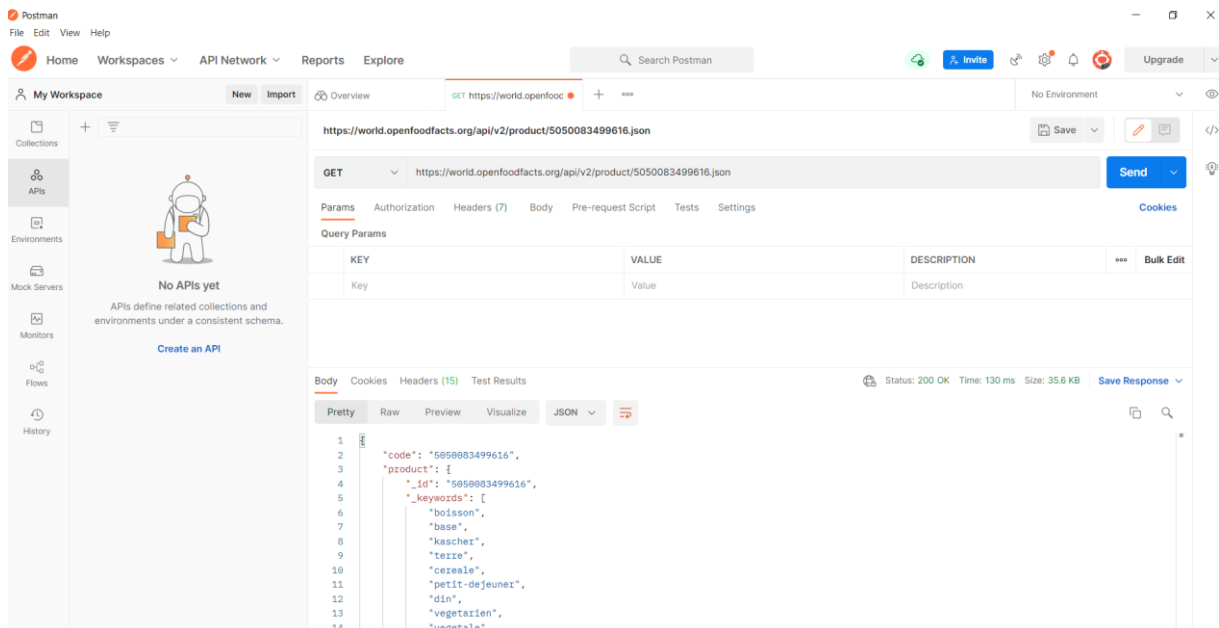
Pour vos tests, n'hésitez pas à appeler l'API en fournissant un autre code barre.

Exemple : <https://world.openfoodfacts.org/api/v2/product/5050083499616.json>

POSTMAN

Installez Postman pour avoir un premier rendu du json reçu lors de votre appel à l'API (<https://www.postman.com/downloads/>)

Saisissez votre URL dans le champ prévu à cet effet et appuyez sur Send :



Vous aurez ainsi un aperçu de votre JSON

Fonctions asynchrones

Le webservice étant hébergé sur un serveur distant, il est préférable de faire de notre fonction fetch une fonction asynchrone (voir cours optionnel sur les promesses) en utilisant les mots clés **async** / **await**.

```
strISBN = "5050083499616";

fetchAPI("https://world.openfoodfacts.org/api/v2/product/" + strISBN);

async function fetchAPI(url) {
  await fetch(url)
    .then((response) => {
      return response.json();
    })
    .then((_data) => {
      console.log(_data);
      //on appelle ensuite notre fonction qui construit notre DOM
    })
    .catch(function (error) {
      console.log(error);
    });
}
```

Si le cours transmis sur les promesses n'est pas clair, se référer à cette page : <https://www.pierre-giraud.com/javascript-apprendre-coder-cours/async-await/>

Optionnel

Vous êtes libres d'ajouter toute page ou toute option, effet qui donnera du « cachet » à votre production (une page de contact, une page de recrutement, la présentation de l'entreprise etc....) dans la mesure où cela respecte la charte du site).