esprit Se former autrement HONORIS UNITED UNIVERSITIES	Année Universitaire : 2022-2023 

Atelier n° 3/4:

Consommation des web services

Objectifs:

- Consommer des services web en utilisant json-server et le client HTTP **Axios**.

<u>Travail à faire :</u>

Dans le but de terminer la gestion du stock dans notre boutique, nous allons implémenter la récupération des appels d'API en utilisant le serveur JSON.

1) Installer json-server en utilisant la commande :

npm install -g json-server ou yarn add -g json-server

2) Mettre le fichier **db.json** dans un dossier appelé **api** sous **src**.
3) Démarrer le serveur JSON en exécutant la commande suivante sous le répertoire **src/api**:

json-server --watch db.json --port 3001

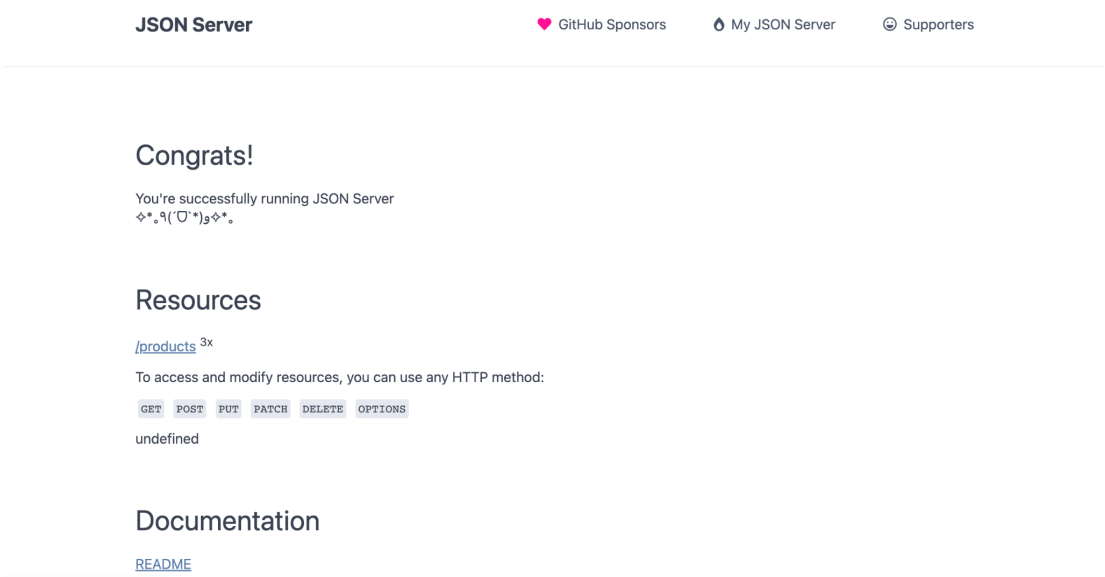
```
\{^_^}/ hi!

Loading db.json
Done

Resources
http://localhost:3001/products

Home
http://localhost:3001

Type s + enter at any time to create a snapshot of the database
Watching...
```



4) Créer un nouveau fichier appelé **api.js** sous un nouveau dossier **service** sous **src**.

   Dans le fichier **api.js**, vous allez mettre ce contenu :

```
import axios from 'axios';

const url = "http://localhost:3001/products";

export const getallProducts = async (id) => {

   id = id || '';

   return await axios.get(`${url}/${id}`);

}

export const addProduct = async (product) => {

   return await axios.post(url,product);

}

export const editProduct = async (id, product) => {

   return await axios.put(`${url}/${id}`,product);

}

export const deleteProduct = async (id) => {

   return await axios.delete(`${url}/${id}`);

}
```

Remarque : vous devez installer le client HTTP Axios :

npm install axios / yarn add axios

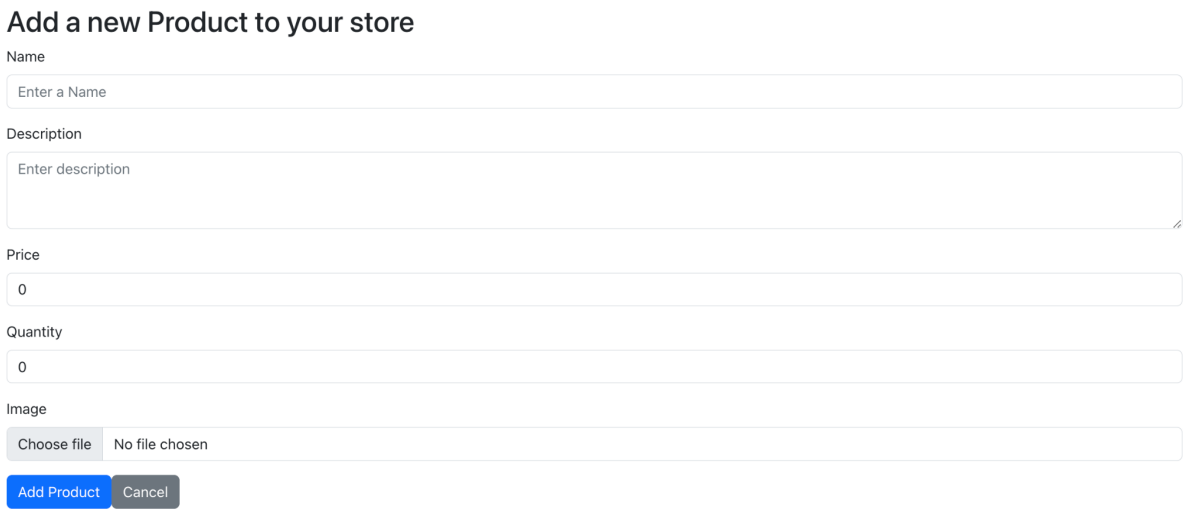
5) Faire les changements nécessaires dans le fichier **Products.js** pour faire l'appel de la liste des produits.
6) Faire les changements nécessaires dans le fichier **ProductDetails.js** pour faire l'appel d'un produit selon l'id.

Remarque : Si le produit n'existe pas , un message **"Product does not exist"** va être affiché.

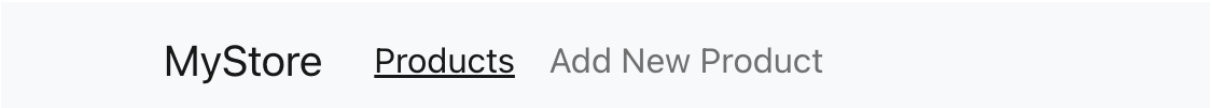
7) Créer un nouveau composant appelé **AddProduct** dans le dossier **Component** où vous allez implémenter un formulaire d'ajout d'un produit avec l'invocation du service web d'ajout et ajouter la route nécessaire.

Remarque :

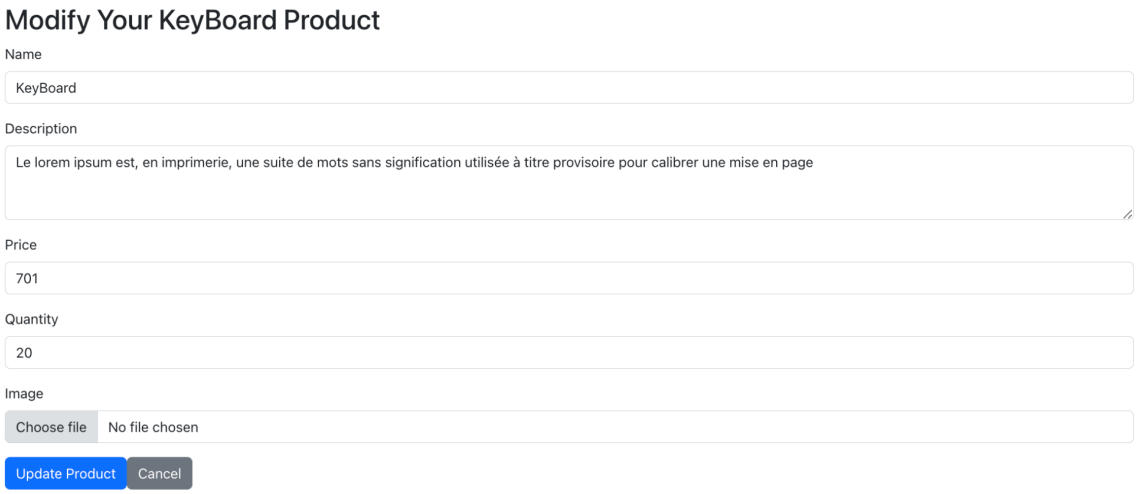
- En cliquant sur le bouton **Add Product**,vous allez rediriger l'utilisateur vers la page de la liste des produits **(utiliser le hook useNavigate)**.



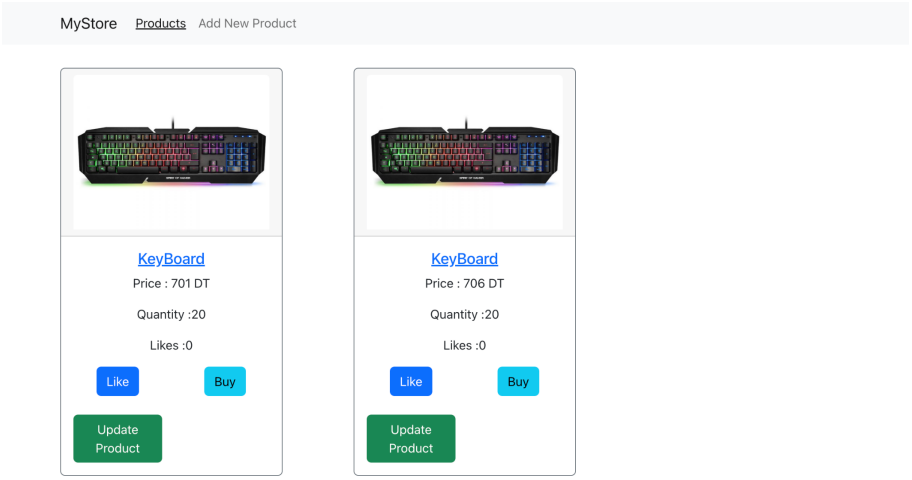
8) Ajouter un bouton **Add new Product** dans le composant **NavigationBar** pour accéder à l'interface d'ajout.



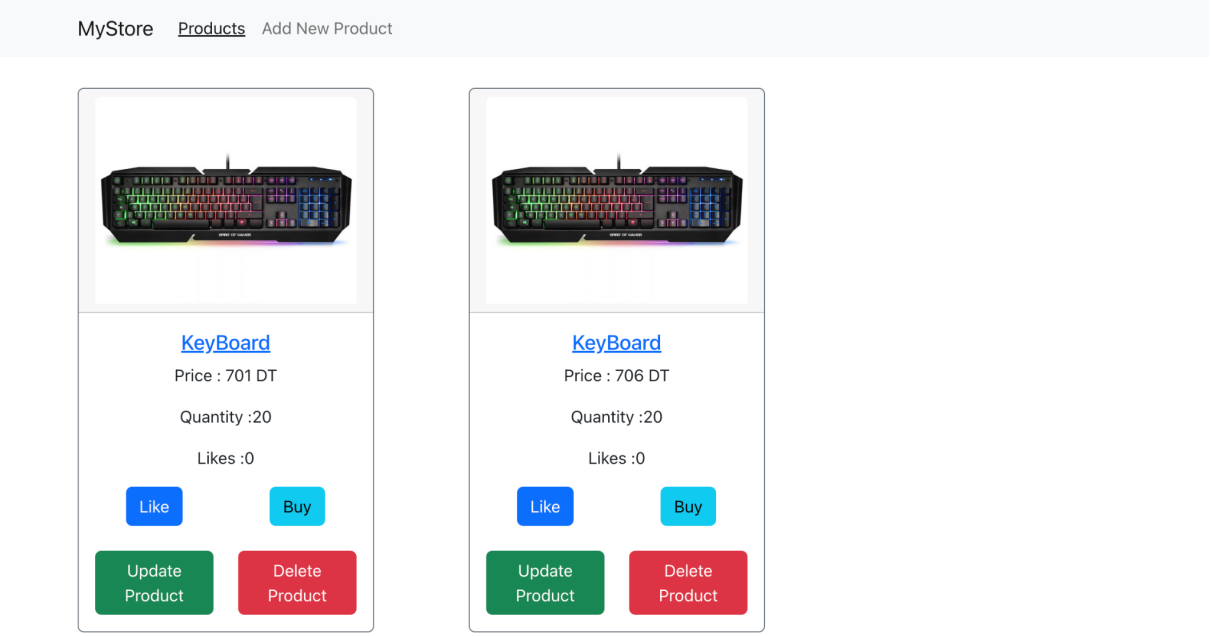
9) Créer un nouveau composant appelé **UpdateProduct** dans le dossier **components** où vous allez implémenter un formulaire de mise à jour d'un produit bien déterminé avec l'invocation du service web de la modification.



10) Ajouter un bouton **Update Product** dans le fichier **Product.js** pour accéder à l'interface de modification et faire les changements nécessaires dans le fichier **App.js** pour ajouter la route du composant de la modification d'un produit..



11) Ajouter un bouton **Delete Product** dans le fichier Product.js.



12) Implémenter la fonction de la suppression d'un produit en invoquant le service web de suppression.