

Panier javascript

Introduction

Dans ce TD on va créer une page affichant une liste de produits, similaire à un site e-commerce classique. Pour insérer les éléments visuels dans la page, on peut avoir recours à jQuery ou bien JavaScript seul (au choix). Pour la partie asynchrone, vous pourrez utiliser fetch, ou faire faire appel à un framework (jQuery ou Axios)

Etape 1 : Liste de produits

Avec l'objet `products`, construire la liste des produits dans la page HTML dans le div `#products`. Le fichier `app.js` contient la structure de base du pattern module. On va créer une méthode `panier.modules.actions.construireListeProduits` qui fera ce travail.

Utiliser le fichier `styles.css` pour mettre en forme la liste des produits. Chaque article doit présenter les valeurs suivantes : nom du produit, description, image, prix.

Etape 2 : Asynchrone

Présentation sur Ajax

API

Au lieu d'utiliser une liste de produits locales, nous allons plutôt la récupérer depuis une API Accéder à l'API <https://tools.sopress.net/iut/panier/> et familiarisez vous avec les routes proposées L'API requiert un token pour toutes les actions relatives au panier. On utilisera une adresse mail comme token.

La racine de l'API est <https://tools.sopress.net/iut/panier/api/>

Définir dans votre code les deux constantes suivantes :

```
const TOKEN = 'mon.adresse.mail@mail.com';

const API = 'https://tools.sopress.net/iut/panier/api/';
```

Puis faire un appel à l'API de cette manière (ici avec jQuery)

```
$.get(`${API}products?token=${TOKEN}`).then(data => {  
    console.log(data)  
})
```

Etape 3 : Constuire le panier

Produits

- Ajouter un bouton "Ajouter au panier" pour chaque produit. Via l'API `cart` en POST, faire en sorte que le produit concerné soit ajouté au panier en cliquant sur le bouton.
- A l'aide de l'API `cart` en GET, récupérer la liste des produits qui sont dans le panier et afficher le contenu du panier dans le div `#panier-content`. Afficher aussi le montant total du panier.
- faire en sorte que le panier soit mis à jour quand on ajoute un produit au panier
- Pour chaque produit dans le panier, ajouter un bouton "Retirer" qui va retirer le produit du panier via l'API `cart/{pid}` en DELETE
- Faire en sorte que le panier soit vidé au clic sur le bouton "Vider le panier" via l'API `cart/` en DELETE

Etape 4 : validation asynchrone

Principe

Nous devons commander les produits du panier en passant par l'API `car/{pid}/buy` en PUT. Il faut donc faire un appel de cette route API pour chaque produit dans le panier. L'API met un temps aléatoire pour retourner la réponse disant que la commande du produit est validée.

Commander tous les produits du panier

Au clic sur le bouton "Commander", faites la commande de tous les produits du panier un par un. Dès qu'une réponse positive est reçue pour un des éléments du panier, afficher la ligne du produit correspondant en vert dans le panier. Quand tous les produits sont commandés, afficher une alerte à l'utilisateur "Tous les produits sont commandés".

Informations et consignes

- Respectez le pattern module
- Indenter et commenter le code

Etape 5 : Bonus

Pagination

L'API produits retourne par défaut les 6 premiers produits de la liste. Ajouter une navigation permettant d'aller à la page précédente / suivante en utilisant le paramètre `page` de l'api `products`

Options de tri

Ajouter la possibilité de trier la liste des produits sur le nom du produit (ordre croissant / décroissant) et sur le prix du produit en utilisant les paramètres `sort` et `field` de l'api `products`