# Planification de tests unitaires Orinours

# index.js

*Requête GET avec l'API fetch*

- Lignes: 3 à 32.
- Description: envoie une requête pour récupérer tous les produits de l'api si la réponse est positive une boucle est créée pour récupérer tous les produits et leurs caractéristiques (ligne 13), du html est ensuite créer et s'itère en fonction du nombre d'articles (lignes 17 à 23) si une erreur est retournée celle ci sera retournée dans le bloc catch (ligne 30 à 32).
- Déclenchement: au chargement de la page.
- Problème possible: ne reçoit pas les données de l'API; faire un *console.log(response)* pour voir les *array* envoyé par le serveur

# Produit.js

***Requête GET pour l'affichage du produit sélectionné***

- Lignes: 7 à 38.
- Description: Récupération du produit avec l'id associé depuis le serveur et création du html pour afficher le produit sélectionné et ses caractéristiques (ligne 7 à 18).
  La méthode forEach va permettre d'itérer les propriétés du tableau colors ensuite l'événement click lance la fonction d'ajout au panier.
- Déclenchement: au chargement de la page.
- Problème possible: ne reçoit pas les données de l'API; faire un *console.log(response)* pour voir le tableau envoyé par le serveur

*addItemCart(item)*

- Lignes: 42 à 76.
- Description: permet de créer un tableau unique qui sera enregistré dans le localStorage à chaque fois qu'un produit est ajouté au panier, si le *localStorage* est vide elle crée un nouveau tableau *cartItem* et l'enregistre dans le localStorage (lignes 57 à 59) sinon elle récupère le tableau déjà existant, ajoute le nouveau produit et l'enregistre dans le localStorage. (lignes 62 à 73)
- Déclenchement: s'exécute après que la requête *GET* est passé.
- Problème possible: n'enregistre pas les données dans le *localStorage*; vérifier le fichier *localStorage* via l'inspecteur d'élément

# Paniers.js

*displayQuantity()*

- Ligne: 8 à 137.
- Description:  Créer le tableau html concernant le panier avec les items du *localStorage* si ils ont été sélectionnés précédemment dans la page produit  et implémente le formulaire en html. sinon il génère au lien vers la page d'accueil
- Déclenchement:  au chargement de la page.
- Problèmes possible: n'affiche pas les items du *localStorage*.

*Fonction addOneItem*

- Ligne: 142 à 147.
- Description: incrémenter la quantité de produit  via l'écoute du bouton + (ligne 102).
- Déclenchement:  suite à l'appel de la fonction *displayQuantity*()
- Problèmes possible: ne modifie pas les quantités dans le localStorage

***Fonction removeOneItem***

- Ligne: 152 à 167.
- Description: décrémente la quantité de produit  via l'écoute du bouton - (ligne 95). Si la quantité tombe à 0 l'item est retiré du *localStorage*.
- Déclenchement:  suite à l'appel de la fonction *displayQuantity*()
- Problèmes possible: ne modifie pas les quantités dans le localStorage

***Fonction deleteItemSelect***

- Ligne: 174 à 183.
- Description: supprime l'article sélectionné et supprime également dans dans le *localStorage* via l'écoute du bouton (ligne 111)
- Déclenchement:  suite à l'appel de la fonction *displayQuantity*()
- Problèmes possible: ne modifie pas les quantités dans le localStorage

***Fonction cancelMyOrdered***

- Ligne: 152 à 167.
- Description: annulation de tout le panier et vide le *localStorage* via l'écoute du bouton (ligne 116)
- Déclenchement:  suite à l'appel de la fonction *displayQuantity*()
- Problèmes possible: ne modifie pas les quantités dans le *localStorage*

***Fonction updateNumberArticles***

- Ligne: 196 à 200.
- Description: Réinitialise la section "*selectionProduits*" et le nombre d'article dans le panier.
- Déclenchement:  au chargement de la page.
- Problèmes possible: ne modifie pas la quantité de produit dans le panier.

*sendForm()*

- Lignes: 207 à 229.
- Description: récupère les valeurs du formulaire (input) dans l'objet *contact* et les *id* des produits dans le tableau products qui sont envoyé dans la fonction *postOrder*.
- Déclenchement: s'exécute après que la requête GET est passé.
- Problème possible: ne pas récupérer les valeurs des inputs.

*postOrder()*

- Lignes: 235 à 259.
- Description: requête POST envoi au serveur l'objet *contact* et le tableau *product* et enregistré dans le localStorage et remplacé par la page confirmation.
- Déclenchement: s'exécute après avoir cliqué sur le bouton "valider votre commande"
- Problème possible: n'envoie pas les objets au serveur et la page confirmation ne se génère pas.

# confirmation.js

- Lignes: ligne 1 à 23
- Description: Récupération des différents éléments dans le localStorage afin de les afficher sur la page confirmation puis retire les informations du localStorage pour une future nouvelle commande.
- Déclenchement: au chargement de la page.
- Problème possible: ne récupère pas les informations du localStorage.