

Plan de tests

Les tests seront effectués avec le framework de test Jest qui inclue une fonctionnalité de couverture de code, ce qui va nous permettre de savoir quelle pourcentage du code total est testé.

Tests des fonctionnalités majeures de l'application :

Les fonctionnalités principales de l'application à tester sont :

- Afficher la liste des articles disponibles à la vente.
- Permettre à l'utilisateur d'ajouter le produit sélectionné au panier.
- Une fois dans le panier, permettre de modifier la quantité de chaque produit.
- Permettre de supprimer un produit du panier.
- Récupérer les informations de l'utilisateur et de vérifier qu'elle respectent bien le format attendu.
- Envoyer ces informations à l'application qui retourne un numéro de commande.

Tests unitaires des fonctions de l'application :

- `getData()` :
 - Fait une requête HTTP
 - Si la requête est correcte, la fonction récupère les données d'un fichier JSON
 - Sinon, affiche une erreur

Tests :

- Faire une requête valide pour tester la récupération des données.
- Faire une requête non valide pour tester l'erreur.

- `PostData()` :
 - Fait un requête HTTP
 - Si la requête est valide, envoie un objet avec un format particulier à l'API et retourne un numéro de commande
 - Sinon, renvoie une erreur

Tests :

- Envoyer un objet avec un format invalide pour tester l'erreur
- Envoyer un objet avec un format valide pour tester le retour du numéro de commande

- `buildIndex()` :
 - Appelle la fonction `getData()` et stocke les données dans une variable.
 - Pour chaque élément récupéré, crée une instance de la classe `Product`

Tests :

- Tester la variable pour s'assurer qu'elle a bien récupéré les objets grâce à `getData()`

- Vérifier que le nombre d'instance créées est bien égal au nombre d'objets dans la variable
- buildProductPage() :
 - Appelle la fonction getData() et stocke les données dans une variable.
 - Affiche les informations de l'objet récupéré en créant un instance de la classe Product
 - Écoute le clic sur le bouton « ajouter au panier » et ajoute l'élément au panier

Tests :

- Tester la variable pour s'assurer qu'elle a bien récupéré les objets grâce à getData()
- Envoyer un faux objet pour tester l'affichage en cas d'erreur.

- SubmitOrder() :
 - Appelle la fonction postData() avec l'objet à envoyé à l'API comme argument
 - Modifie le LocalStorage
 - Charge une nouvelle page

Test :

- Vérifier le contenu du localStorage avant et après.

- AddToCart() :
 - Si le panier est vide, ajoute l'article à un objet « panier » stocké dans localStorage
 - Sinon, récupère l'objet « panier » du localStorage et :
 - Si l'article est déjà dans le panier, ajoute un à la quantité
 - Sinon, ajout l'article au panier

Test :

- Tester le contenu du localStorage avant et après dans chaque configuration (vide, non vide et article non présent, non vide et article présent)

- addArticle() / subtractArticle():
 - Modifie la propriété « quantité » d'un article du panier
 - Modifie la valeur de la notification
 - Modifie la valeur d'un élément du DOM

Tests :

- Vérifier l'état avant et après de la propriété quantité
- Vérifier l'état avant et après de la notification
- Vérifier l'état avant et après de l'élément du DOM
- Faire les tests avec différentes valeur pour s'assurer que les conditions sont correctes

- deleteArticle() :
 - Supprime un élément de l'objet « panier »
 - Modifie la valeur de la notification
 - Mise à jour du localStorage

Tests :

- Vérifier l'état avant et après de l'objet « panier »
- Vérifier l'état avant et après du localStorage

- cartNotifications() :
 - Affichage ou masquage d'un élément du DOM
 - Modification d'un élément du DOM

Tests :

- Tester la valeur affichée
- Tester une valeur inférieure à zéro pour s'assurer du résultat si la condition n'est pas remplie

- displayButton() :
 - Applique un style à un élément du DOM en fonction de la condition remplieRetourne 'True' si la condition est remplie et 'False' si elle ne l'est pas.

Tests :

- Tester les conditions et vérifier que la fonction retourne bien ce qu'elle est censé faire.

- IsValid() :
 - Compare un input à une regex.
 - Renvoie 'true' si l'input matche la regex

Tests :

- Tester avec un texte ne matchant pas la regex et un autre qui matche