Plan de tests Orinoco

Architecture générale

L'application web sera composée de 4 pages :

- une page de vue sous forme de liste, montrant tous les articles disponibles à la vente ;
- une page "produit", qui affiche de manière dynamique l'élément sélectionné par l'utilisateur et lui permet de personnaliser le produit et de l'ajouter à son panier ;
- une page "panier" contenant un résumé des produits dans le panier, le prix total et un formulaire permettant de passer une commande. Les données du formulaire doivent être correctes et bien formatées avant d'être renvoyées au back-end. Par exemple, pas de texte dans les champs date ;
- une page de confirmation de commande, remerciant l'utilisateur pour sa commande, et indiquant le prix total et l'identifiant de commande envoyé par le serveur.

Fonctionnalités à tester

1. **Affichage de la liste des produits disponibles via l'API sur la page d'accueil « index.html ».**

   1. Tester que les données de l'API sont bien reçues et correspondent bien au modèle fournit dans les spécifications fonctionnelles.
   index.js : lignes 56 à 71, 120 à 124.

   2. Tester si au clic sur le bouton 'Plus d'infos' d'un des produits, l'utilisateur est bien redirigé vers la page du produit concerné.
   index.js : ligne 68.

3. Tester que l'info-bulle indiquant le nombre d'élément dans le panier soit cachée si le panier est vide et affichée s'il y a des articles.
index.js : lignes : 529 à 540.

2. La page « produit.html » affiche la description d'un produit en particulier ; son identifiant est passé en paramètre de requête de l'URL.

1. Tester si l'id du produit existe bien, sinon renvoyer une erreur.
index.js : lignes 94 à 100.

2.Tester que les données de l'API sont bien affichées à l'écran et que l'utilisateur peut sélectionner une couleur pour personnaliser son produit.
index.js : lignes 103 à 114.

3. Tester que si l'utilisateur choisit moins d'un article, la valeur du nombre d'article choisit est définit sur 1.
index.js : ligne 139.

3. La page « panier.html » affiche un récapitulatif de la commande en cours et permet de modifier le panier et d'effectuer une commande.

1. Tester qu'il y a bien un message indiquant à l'utilisateur que son panier est vide si c'est le cas.
index.js : lignes 197 à 201.

2. Tester que tous les articles présent dans le localStorage soient affichés dans la vue avec toutes les infos nécessaires pour la description d'un article.
index.js : lignes 202 à 240.

3. Tester que le sous-total corresponde bien au prix total des articles et au nombre d'articles dans le panier.
index.js : lignes 242 à 281.

4. Tester si le formulaire ne contient que des lettres, des chiffres, des espaces ou des tirets et qu'il ne commence pas par un espace sinon afficher une erreur et empêche le formulaire de s'envoyer avec des données non valides.
index.js : lignes 327 à 367.

5. Tester si l'envoie des données à l'API ne retourne pas une erreur.
index.js : lignes 370 à 404.

6. Tester que la suppression d'un article dans le panier est prise en compte et que le panier se met à jour en conséquence.
index.js : lignes 483 à 502.

7. Tester que choisir une quantité d'article directement dans le panier est bien pris en compte, que l'utilisateur ne peut pas choisir moins de 1 article et que le panier se met à jour.
index.js : lignes 508 à 516.

8. Tester que lorsque l'on clic sur le bouton : 'Passer la commande' le formulaire s'affiche.
index.js : lignes 303 à 309.

4. La page « confirmation.html » affiche dans la vue un message confirmant la commande et un récapitulatif de toute la commande passée par l'utilisateur.

1. Tester que s'il n'y a pas de commande en cours, un message est affiché dans la vue pour en informer l'utilisateur.
index.js : lignes 417 à 421.

2. Tester que le numéro de la commande a bien été récupéré via l'API et affiché dans la vue.
index.js : lignes 414, 424, 440.

3. Tester que tous les articles présents dans le panier lors du passage de la commande soient bien affichés dans la vue avec les infos de chaque article.
index.js : lignes 413, 415, 444 à 462.

4. Tester que l'on a bien récupéré les infos de contact du formulaire et les affiche dans la vue.
index.js : lignes 414, 426 à 436.

5. Tester que le montant de la commande et le nombre d'articles soient bien affichés dans la vue.
index.js : lignes 441, 466.

6. Tester que le panier est bien vide après que la commande soit confirmée.
index.js : ligne 473.