Planification des tests unitaires

Suite de tests unitaires pour couvrir au minimum 80% de la base de code pour le frontend, soit sur 4 pages html et 4 fichiers Javascript associés.

Globalement à tester :

- Vérification de la disponibilité des APIs et gestion des erreurs,
- Passage d'une page à l'autre avec l'affichage demandé,
- Article mis au panier (ou caddy),
- Suppression de l'article du panier,
- Modification de la quantité de l'article (en plus ou moins),
- Calcul du prix total,
- Validation des champs du formulaire et du formulaire lui-même,
- Page de confirmation après interrogation de l'API.

NOM DU TEST	ACTION A EFFECTUER	RESULTAT ATTENDU	QUE VERIFIER ?	CODE A CONTROLER
Connexion à la page index	Entrez l'adresse du site dans le navigateur: https://rvdub.github.io/Orinoco/ Ne pas oublier de lancer les API en localohost pour se servir de l'application	La page d'accueil s'ouvre avec tous les choix thématiques	A fin 2020, trois thèmes sont disponibles: OriTeddies (peluche), OriCam(app. Photo), OriFurniture(meubles en chêne)	Page index.html: disponible et en ligne
Barre de recherche	Cherchez un mot ou un élément d'un mot pour trouver son thème plus facilement dans la liste	A chaque saisie d'un caractère le résultat s'affine	Majuscule ou minuscule indépendemment: tapez C (Les 3 thèmes disponibles), puis H (teddies et Cam), puis E (teddies uniquement).	index.js // fonction barre de recherche, lignes 1 à 9
Connexion à la page catalogue (suivant le choix client)	Au clic sur un bouton du catalogue choisi, nous arrivons bien au bon catalogue	La page catalogue thématique s'ouvre avec la liste de tous les articles disponibles sur l'API	Clic sur bouton 'Ours en peluche', ouverture d'une nouvelle page avec 5 articles à fin 2020: Norbert, Arnold, Lenny, Gustav, Garfunkel	index.js // fonction d'écoute d'événement sur bouton et utilisation de localStorage, lignes 12 à 31
Requête GET à l'API (suivant thème)	Gestion de la requète avec des promesses et vérification du résultat et gestion de l'erreur	Retour d'un fichier objet avec le contenu des éléments à afficher. Affichage de vérification sur la console	Dans la console, un tableau "data" de 5 objets donne toutes les données renvoyées par l'API. A l'écran, seules s'affichent la photo, un bouton avec le nom de l'article, un descriptif et un prix	product.js // fonction global askApis(theme), lignes 30 à 40 environ
Test panier vide	La barre de navigation autorise l'accès au panier: vérifiez que le panier est bien vide	Message d'alerte indiquant que la panier est vide	Absence d'affichage d'un prix nul ou autre. Dans la console, tableau Array vide	Gestion direct sur la page: my_caddy.html <main> , lignes 59 à 64
Test panier vide (mais formulaire rempli)	Bloquez l'envoi du formulaire sans avoir passer une commande d'article(s)	Affichage d'une alerte indiquant l'absence de commande	prix total(totalPrice) à 0 et ouvre une boite d'alerte: Vous ne pouvez pas commandez, il n'y a rien dans le panier. Dans la console, Array vide	my_caddy.js // fonction d'écoute du bouton Envoyez (environ lignes 135 à 143)
Affichage de tous les produits du thème	Affichez les articles quelque soit leur nombre en gérant de façon dynamique la page html et en ajoutant l'identifiant unique de l'article	Affichage de chaque produit dans une carte avec photo, nom dans un bouton, texte descriptif et prix. L'identifiant unique de l'article est entré dans l'argument id du bouton	5 articles à fin 2020: Norbert, Arnold, Lenny, Gustav, Garfunkel. Dans la console, un tableau "data" de 5 objets donne toutes les données renvoyées par l'API (en plus de l'affichag écran, son id et les options possibles qui sont des couleurs pour les peluches)	products.js // fonction displayProduct (environ lignes 47 à 62)

NOM DU TEST	ACTION A EFFECTUER	RESULTAT ATTENDU	QUE VERIFIER ?	CODE A CONTROLER
Affichage du prix avec 2 décimales	Transforme le prix entier donné par l'API et ajoute la monnaie euros.	La modification n'agit que sur l'affichage. Le prix reste en centimes et les aditions ultérieurs se font sur des nombres entiers	Norbert 29.00 €uros. A fin 2020, tous les prix terminent par 00, quelque soit le thème	products.js // fonction priceDisplay (environ lignes 42 à 45)
Sélection de l'article grâce à son bouton	Dispose une écoute sur chaque bouton de l'article, quelque soit le nombre d'article. Récupère son identifiant et le numéro d'ordre de l'article dans l'affichage. Gère le choix de l'option en fonction du thème, car l'APIs ne renvoient pas le même nom du tableau d'option (colors, lenses ou varnish) et, quelque soit le nombre d'options disponibles	Préparez l'affichage de la sélection avec des identifiants, la bon intitulé de l'option et son contenu spécifique	Affichage de contôle sur la console des 2 identifiants: identifiant unique renvoyé par l'API et numéro d'ordre. Norbert id=5be9c8541c9d440000665243 et numéro d'odre dans l'affichage à l'écran=0, soit le premier	products.js // fonction selectOneProduct (environ ligne 64 à 91)
Affichage de l'article selectionné	Gestion dynamique de l'affichage de l'article. Trois boutons utilisateur sur écoute: retour à la liste des articles (fonction backToTheList), menu déroulant choix d'option (et son enregistrement immédiat), page du panier	Affichage de l'article avec deux informations supplémentaires: les options disponibles (choix obligatoire avec test) et un bouton pour ajouter au panier. Modification du titre h1 de la page. Activation d'un bouton du menu	Retour à la liste des articles Teddies possibles. Titre de la page: Votre judicieux choix. 4 options de couleurs pour Norbert: Tan, Chocolate, Black, White	products.js // fonction displayOneProduct (environ ligne 93 à 127)
Enregistement au panier	Utilisation de la mémoire du navigateur côté client pour l'enregistrement d'un objet contenant tous les champs de la sélection client (et quantité à 1). Gestion des erreurs avec une promesse et validation de l'information complète. Ouverture de la base indexée sur l'id article, puis transactions	Affichage d'une alerte validant la saisie utilisateur. Enregistrement de tous les champs connu de l'article selectionné dans un objet. Affichage sur la console de la bonne réalisation de la promesse	Alerte: Enregistré au panier. Sur la console: choix ajouté avec identifiant Norbert idem ci-dessus. Transaction terminée indiquant l'absence d'erreur. Si l'article est déjà au panier, pas de nouvel enregistrement sur la console. Variable choice contenant l'Id unique, l'option choisi, adresse image, le nom Norbert, sa description, son prix, quantité à 1 et la date du choix pour le fun ;)	products.js // fonction putInMyCaddy (environ lignes 129 à 153)

NOM DU TEST	ACTION A EFFECTUER	RESULTAT ATTENDU	QUE VERIFIER ?	CODE A CONTROLER
Retour à la liste thématique (depuis l'article unique ou depuis le panier)	Avant de revenir à la fonction displayProduct, effacement de toutes les variables ayant servi à pointer l'article selectionné et effacement de l'affichage de la sélection	Changement du titre h1. Mise à zéro de l'affichage et des variables de sélection d'un article. Le retour a la liste maintient le panier	Retour à la liste des Teddies depuis le menu (après Norbert mis au panier ou bien en consultation libre du panier). L'affichage ressemble en tous points à ce que nous avons vu précédement, sinon les variables n'ont pas été remises à 0. Idem l'affichage sur la console risque de ne pas valider la nouvelle sélection du client	products.js // fonction backToTheList (environ lignes 155 à 165)
Affichage du contenu du panier	Interrogation de la mémoire du navigateur (requête et promesse). Utilisation d'une boucle pour affichage de tous les articles, quelque soit leur nombre. Introduction de l'id article dans le code html pour bien imputer la modification du nombre ou la suppression de l'article	Résumé de l'article avec photo réduite, nom et son option, quantité modifiable et suppression possible. Affichages de vérification sur la console	Photo miniature de Norbert Tan (1ère option du menu déroulant). Sur une deuxième ligne: Prix: 29.00 €uros; Quantité: 1, dans un menu déroulant autorisant un choix entre 1 et 10, puis bouton supprimer	my_caddy.js // fonction displayData (environ lignes 35 à 60)
Calcul du prix et tableau des id (en même temps)	A chaque affichage, deux compteurs (mis préalablement à 0) calculent le prix total et enregistrent un tableau des id. Pour diminuer le risque d'erreur, je ne fais que des additions. A chaque modification du client, l'affichage est mis à 0 et on recalcule dans le nouvel affichage	Le prix total apparaît sur la page my_caddy.html et un tableau des id dans la console	Norbert 29.00 €uros et Arnold 39.00 €uros donne un total affiché de 68 euros. Dans la console, un tableau contient les id de Norbert et Arnold: "5be9c8541c9d440000665243" et "5beaa8bf1c9d440000a57d94". 10 Norbert coûtent 290 euros.	my_caddy.js // fonction displayData (environ lignes 54 et 55)
Gestion de l'Alert-info et de l'affichage d'un prix non nul	1er) affichage dynamique du bandeau alerte: modification du titre si panier plein ET si non fermé par l'utilisateur 2nd) affichage du prix uniquement différent de 0	Affichage de l'alerte info modifiée si l'utilisateur ne l'a pas fermée et, le prix n'apparaît que si le panier contient un article	Votre panier est vide !, remplacé par Votre sélection d'article(s). Prix total = 29.00 euros pour Norbert. Aucun prix si aucun article. Prix total = 0 euros interdit.	my_caddy.js // fonction displayData (environ lignes 65 à 76)
Choix à l'intérieur du panier: quantité ou supprimer	Tous les boutons sur écoute, quelque soit leur nombre. Interrogation du cache navigateur à l'aide d'une transaction: suppression simple grâce à l'id; modfication de la quantité de l'id puis mise à jour de l'objet choice	Modification de la quantité ou suppresion d'un article: nouvel affichage et totaux (prix et tableau des id) mis à jour. La navigation est toujours possible vers la liste pour remettre l'article. Affichages de vérification dans la console	Quantité 2 Norbert, Total 58.00 euros. "Quantité modifiée" apparaît dans la console. Suppression: l'article Norbert et la ligne Total disparaîssent; "Choix supprimé" dans la console assure la validation de la modification	my_caddy.js // fonction modifyChoice (environ lignes 79 à 100 fonction quantité, 104 à 128 fonction supprimer)

NOM DU TEST	ACTION A EFFECTUER	RESULTAT ATTENDU	QUE VERIFIER ?	CODE A CONTROLER
Validation du formulaire	Les données du formulaires doivent être correctes et bien formatées	Si champ oublié, une info-bulle indique qu'il faut le remplir. Si format non respecté, une autre info-bulle le rappelle. Envoi possible uniquement quand les cinq champs sont correctement remplis. Formulaire récupéré en JS et bouton submit désactivé	Tant qu'un cadre du formulaire apparaît en rouge, le formulaire ne peut être soumis. Ces coordonnées valident le formulaire: R.\ D'Arbrissel-Fontevraud \10A rue de la 1ère Armée Fr. \9220 Brain /Longuenée Cedex 9 \r@af.fr	my_caddy.html lignes 76 à 109 environ. My_caddy.js (lignes 135 à 143 environ)
Récupération données du formulaire	Une boite de dialogue informe le client que sa saisie fonctionne. L'objet JSON attendu par l'API teddies apparaît dans la console. Fermez la boite de dialogue pour que le processus continue	Stockage des 5 champs dans 5 variables et construction de l'objet JSON	{"contact":{"firstName":"R.","lastName":" D'Arbrissel-Fontevraud","address":"10A rue de la 1ère Armée Fr.","city":"9220 Brain /Longuenée cedex 9","email":"r@af.fr"},"products":["5be9c8 541c9d440000665243"]}	my_caddy.js // fonction dataRetrievalContactForm (environ lignes 145 à 168)
Requête POST à l'API (quelque soit le thème)	Gestion de la requête POST avec promesse, vérification du résultat et gestion de l'erreur	Retour d'un string orderId. Affichage de vérification sur la console. Sauvegarde sur localStorage du prix total et du numéro de confirmation de commande	Exemple de réponse positive du serveur sur la console: orderId 5b5b22b0-518d-11eb-a0f3-fd31bd6ad61a	my_caddy.js // fonction global askApis(theme) lignes 169 à 187 environ
Levée d'une exception	La navigation sur le site ne permet pas de sortir d'un catalogue thématique sans arriver à la commande, mais la flêche retour du navigateur oui. Il est donc possible de commander dans les trois catalogues thématiques disponibles à ce jour. Cela fonctionne, mais les APIs ne le gèrent pas encore => modification du texte à afficher en évitant de laisser apparaître "undefined".	Affichage sur la page order_confirmation.html du prix total et du numéro de commande ou du message alternatif expliqué ci-avant. Affichage de contrôle sur la console	Affichage indispensable: prix total et id de commande. Exemple "Confirmation pour un prix total de 29.00 €uros. Votre identifiant de commande est le 5b5b22b0-518d-11eb-a0f3-fd31bd6ad61a."	order_confirmation.js// affichage et test, environ lignes 10 à 17
Mise à zéro du panier	Nous arrivons à la fin du processus d'achat. Il faut videz la base de données du navigateur du panier commandé, pour permettre une nouvelle commande	Panier effacé. Affichage de contrôle sur la console	Dans la console: Base de donnée vidée	order_confirmation.js // fonction clearData (lignes 25 à 55 environ)