Planification de tests unitaires

Front-end du site Orinoco

|  |  |  |
| --- | --- | --- |
| Partie du code testée | Test à réaliser | Résultat attendu |
| Fichier main.js -> fonction updatePanier() (ligne 2) | Ajouter n’importe quel produit au panier | Le nombre de produit dans le panier (indiqué à côté du lien panier dans le header) s’incrémente de 1 |
| Vider le panier | Le nombre de produit revient à 0 |
| Fichier main.js -> fonction getProduit(idTeddy) (ligne 12) | Appeler la fonction avec un id valide | Renvoie l’objet du l’ours en peluche correspondant |
| Appeler la fonction sans argument | Renvoie le texte du statut de la requête en erreur |
| Appeler la fonction avec un id invalide | Renvoie le texte du statut de la requête en erreur |
| Fichier main.js -> fonction getProduits() (ligne 27) | Appeler la fonction | Renvoie les objets des ours en peluche présents dans la base de donnée |
| Fichier index.js -> fonction printProduits() (ligne 1) | Appeler la fonction avec une liste d’objets d’ours en peluche | Chaque ours en peluche s’affiche dans une carte sur la page web |
| Appeler la fonction sans arguments | Rien ne s’affiche sur la page web |
| Appeler la fonction avec un argument qui n’est pas un objet ours en peluche valide | « Erreur lors de l’affichage du produit » s’affiche dans la carte du produit |
| Fichier produit.js -> fonction printTeddy(teddy) (ligne 4) | Appeler la fonction avec un objet ours en peluche | Affichage du produit sur la page |
| Appeler la fonction sans argument | Affichage de « Erreur lors de l’affichage du produit » |
| Appeler la fonction avec un argument qui n’est pas un objet ours en peluche valide | Affichage de « Erreur lors de l’affichage du produit » |
| Fichier produit.js -> fonction addToPanier(teddy) (ligne 30) | Appel de la fonction lorsque le produit n’a pas déjà été ajouté au panier | Création d’un item local storage dont la clé est l’id du produit affiché sur la page et dont la valeur est 1 |
| Appel de la fonction lorsque le produit est déjà présent dans le panier | Récupération de l’item local storage dans la clé est l’id du produit affiché sur la page et augmentation de sa valeur de 1 |
| Fichier panier.js -> fonction printPanier(teddy) (ligne 5) | Appel de la fonction avec un objet ours en peluche | Affichage de la ligne du produit dans le panier (image, nom, prix unitaire, quantité, prix total), Augmentation de la variable totalPrice de la valeur du prix total de cette ligne |
| Appel de la fonction sans argument | Affichage « Erreur, impossible d’afficher ce produit » |
| Appel de la fonction avec un objet qui n’est pas un objet ours en peluche valide | Affichage « Erreur, impossible d’afficher ce produit » |
| Fichier panier.js -> fonction getPanier() (ligne 18) | Appel de la fonction si local storage contient des items valides (clé : id de produit, valeur : quantité) | Appel de la fonction getProduit(idTeddy), Appel de la fonction printPanier(teddy) en cas de réussite et affichage du prix total du panier, en cas d’erreur affichage de l’erreur dans la console et affichage « erreur, impossible d’afficher ce produit » dans le panier. |
| Appel de la fonction si local storage est vide | Rien de ne passe |
| Appel de la fonction si local storage contient des items non valides | affichage de l’erreur dans la console et affichage « erreur, impossible d’afficher ce produit » dans le panier |
| Fichier panier.js -> fonction getProductArray() (ligne 54) | Appel de la fonction si local storage contient des items valides (clé : id de produit, valeur : quantité) | Retourne un tableau composé d’id produits, chaque id produit apparait autant de fois que la valeur correspondant à sa clé dans local storage |
| Appel de la fonction si local storage est vide | Retourne un tableau vide |
| Appel de la fonction si local storage contient des items dont la valeur n’est pas un chiffre supérieur à 0 | Les items concernés ne sont pas ajoutés au tableau |
| Appel de la fonction si local storage contient des items dont la clé n’est pas un id de produit valide | Les items sont tout de même ajoutés au tableau, l’erreur est traitée plus tard (fonction envoiFormulaire()) |
| Fichier panier.js -> fonction postOrder(requete) (ligne 68) | Appel de la fonction avec une requete au bon format | Retourne la réponse envoyée par le serveur |
| Appel de la fonction avec une mauvaise requete ou appel de la fonction sans argument | Renvoie le texte du statut de la requête en erreur |
| Fichier panier.js -> fonction envoiFormulaire() (ligne 84) | Appel de la fonction | * Création de l’objet contact composé des valeurs utilisateur préalablement validées en HTML * Appel a la fonction getProductArray() pour créer le tableau de produit * Création de l’objet requete à partir de l’objet et du tableau précédent * Transformation de l’objet requete en objet JSON * Appel de la foction postOrder(requete) avec l’objet requete en argument * Réussite : Création de deux items session storage avec l’id de la commande retournée par le serveur et le prix total de la commande. Puis redirection vers la page de confirmation * Echec : affichage de l’erreur dans la console et affichage du message d’erreur sur la page |