

Plan de test unitaire

Avant la mise en place des tests unitaire

Pour que les tests se passent bien, je vous conseille de suivre les étapes comme indiqué ci-dessous pour éviter la casse. Télécharger le logiciel postman pour récupérer des portions d'informations. Et utiliser un convertisseur de JSON en ligne pour la rapidité et la praticité (<https://json-to-js.com/>). Ensuite créer votre fichier js et le lier à la page où vous souhaitez réaliser vos tests.

1 Script.js

Test de la fonction `accueilProducts(urlCall)`, `urlCall` est un argument de type `String` qui vaut ("<http://localhost:3000/api/cameras>"). Vous avez qu'à l'appeler pour la tester. objectif de la fonction (afficher produit à la page d'accueil: récupérer données json côté backend pour les traiter avec un fonction enfant)

Test de la fonction `recupInserer(reponse)` est une fonction appelé dans la fonction `accueilProducts`, son argument `reponse` est un array d'objet javascript de type `String` fourni par la fonction parent `accueilProducts(urlCall)`. objectif de la fonction (afficher produit à la page d'accueil: ajouter les éléments recuperer coté backend sur le DOM de la page)

Donc pour le tester vous aurez besoin de l'array d'objet qui se nomme `reponse` (pour le récupérer il suffit d'utiliser le logiciel postman et faire un appel get sur <http://localhost:3000/api/cameras>, pour récupérer les données, et enregistrer les données recuperer dans une variable `reponse` exemple : `const reponse = [{ }, { }]`). Et vous aurez qu'à l'appeler.

2 QueryRecup.js

Test de la fonction `recupArticleUrl(idSansHash)`, pour mettre en place le test unitaire de cette fonction. On aura besoin de récupérer l'id du produit transmit avec l'url comme ceci sur la page ([produit.html#5be1ed3f1c9d44000030b061](#)). si l'id n'est pas transmise automatiquement à cause des tests veuillez l'insérer manuellement pour le bon fonctionnement. Et vous aurez juste à l'appeler pour la

mise en marche. objectif de la fonction (afficher produit à la page de personnalisation appeler produit.html : ajouter un produit recuperer côté backend sur le DOM de la page)

Test de la fonction `ajouterPanier()` , on aura besoin juste de l'appeler pour quelle se met en marche.

3 PanierAffiche.js

Test de la fonction `afficherDansLaPagePanier()` , on aura juste besoin de l'appeler dans votre fichier de test.

Test de la fonction `ajoutAuDom(reponsPanier)`, on aura besoin d'un objet javascript nommé `reponsePanier` qui contient les informations du produit (récupérer les données sur le logiciel postMan en faisant une requête get avec <http://localhost:3000/api/cameras/idDuProduit>). objectif de la fonction (afficher produit unique à la page de panier appeler panier.html : afficher produit unique, appelé dans une boucle permet d'afficher tous les produits ajouter au panier)

4 QueryPost.js

Test de la fonction `totalAfficher()` on aura juste besoin d'avoir un `localStorage` avec la clé `total` et une valeur à l'intérieur.

Test de la fonction `recupererTotalEtPost()` on aura juste besoin de l'appeler vu que c'est le test parent.