# Tests unitaires

## Accueil.js

Fonction getApi pour voir si on récupère bien les objets de l’api -> si nous n’arrivons pas à récupérer les produits, retourne une erreur *« Une erreur est survenue dans la récupération des données. »*

La fonction recupObjet itère les objets de l’api pour les afficher -> Si l’affichage des produits ne fonctionne pas, affiche une erreur *"Nous n'avons pas pu afficher les produits."*

## Produits.js

La fonction getApi récupère un produit selon son id -> Si ne fonctionne pas affiche *"Une erreur est survenu dans la récupération des données."*

La fonction getId récupère le produit d’après l’id passé en paramètres dans l’url -> Si l’id ne peut pas être récupéré, affiche *"Impossible de récupérer l'id"*

La fonction getColor permet de sélectionner une couleur pour l’article -> Si ne fonctionne pas, affiche *"Une erreur est survenue lors du choix de la couleur."*

La fonction addPanier vérifie si le panier est vide et en fonction, créé un array et ajoute ce produit dans un array (son Id) ou récupère l’array existant, ajoute l’id du produit puis enregistre cet array dans le localStorage -> Si ne fonctionne pas, affiche *"Un problème est survenu lors de la sauvegarde du panier."*

## Panier.js

La fonction getPanier récupère le array créé dans la page produit, puis transforme les articles en objets json.

La fonction getPanierLongueur calcule la longueur du panier -> Si le panier est null la longueur = 0, sinon, renvoi la longueur du panier.

La fonction affPanier itère de 0 jusqu’à « longueur » puis exécute la fonction getApi qui permet de récupérer un objet selon son Id -> Si ne fonctionne pas, *affiche "Une erreur est survenu dans la récupération des données."*

Si longueur = 0 car panier = null, affiche un message disant que le panier est vide.

La fonction deleteItem reçoit en paramètre l’id de l’objet à supprimer, puis supprime cet id du panier et recharge la page.

La fonction affichageTotal affiche le prix total du panier.

La fonction sendCommand vérifie que les données entrées par l’utilisateur soient correctes puis POST le panier vers la page de remerciements.

La fonction getApi est une promise qui récupère les produits.

La fonction getTeddies affiche chaque produit dans le panier.

La fonction calcTotal prend chaque prix en paramètres et les additionne.

## Remerciements.js

La fonction checkPanier vérifie si le panier n’est pas vide. Si non vide, longueur = longueur du panier -> Si panier vide, renvoie une fatalError *« Le panier est introuvable ! »*

La fonction mainRemerciements récupère la commande avec le prénom de l’utilisateur puis l’id de la commande si la longueur du panier est supérieur à 0, puis affiche « Merci ‘nom d’utilisateur’ » et « Commande numéro : ‘id commande ‘ »

La fonction recapPanier exécute la fonction getApi qui récupère les produits du panier un par un selon leur id -> Si la récupération ne fonctionne pas, retourne l’erreur *"Une erreur est survenu dans la récupération des données."*  Puis affiche les éléments via la fonction getTeddies -> si ça ne fonctionne pas, affiche l’erreur *"Une erreur est survenue lors de l'affichage des produits dans le panier"*

La fonction deletePanier supprime le panier dès que la commande est passée pour pouvoir faire une nouvelle commande -> si la fonction ne marche pas, affiche *"Une erreur est survenue lors de la suppression du panier"*

La fonction getApi récupère les produits via l’api.

La fonction getTeddies affiche les produits dans le panier.

La fonction affichageTotal récupère le prix total de la commande via le localstorage, enregistré dans la page panier, puis l’affiche.