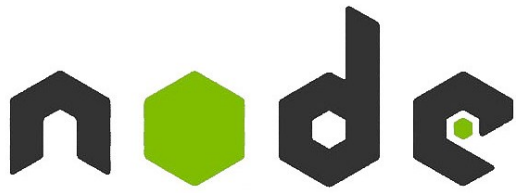


# PLAN DE TESTS

[Lien du projet sur GitHub](#)

Appuyez ici



JavaScript™



Orinoco

Préparé par  
M. EL-KHABOU. E

## Plan de Tests

Fichiers	Lignes	Fonctions Testées	Résultat attendu	Comment vérifier le résultat attendu	Problème possible	Validé
orinoco.js	Ligne 04 à 21	loadCameras()	La fonction asynchrone <b>fetch()</b> lance une requête et renvoie une promesse, mais puisqu'il y a <b>await</b> , la fonction asynchrone sera suspendue jusqu'à la finition de la demande pour faire l'affichage	catch(e) { console.log(e); } La fonction doit retourner un tableau au format JSON	En cas de problème de réseau par exemple, il se peut que la liste d'éléments récupérés et a afficher soit nulle.	Voir
orinoco.js	Ligne 24 à 34	displayCameras	Tous les produits récupérés doivent s'afficher convenablement dans la page de listes des produits	Se positionner dans le répertoire back-end et appeler le serveur avec la commande <b>node server</b> , puis démarrer la page <b>products-list.html</b>	Le back ne renvoi rien / surtout si le serveur est déconnecté ou rencontre un problème de communication	Voir
Fichiers	Lignes	Fonctions Testées	Résultat attendu	Comment vérifier le résultat attendu	Problème possible	Validé
product.js	Ligne 05 à 19	GetCamera()	Récupère les données de la caméra sélectionnée avec l'API Fetch, / _id de la caméra : Création du lien pour afficher la caméra sélectionnée ; La caméra sélectionnée apparaît sur la Page	REF. au dossier backend/routes/camera.js -> router.get('/:id', cameraCtrl.getOneCamera); Le lien doit fonctionner et nous envoie vers la caméra sélectionnée	« id » absent du « backend » ou objet Vide : Le lien ne fonctionne pas ou la caméra ne correspond pas avec celle qui a été sélectionnée, Ou bien l'ID n'est pas correct	Voir
product.js	Ligne 21 à 28	buildCameraProduct	Chaque caméra a bien sa page avec ses propres données (nom, prix, image, options de lentilles et sa description)	Dans une page ou il ya une seule camera, toutes les informations doivent étre remplies	Absence « key : value » La valeur retournée pourrait étre indéfini, ou La valeur ajouté pourrait ne pas étre prise en compte	Voir
product.js	Ligne 32 à 53	AddEventListener	Récupération des données API correspondant a l'"ID" dans une variable produit. L'objet CAMERA stocke les valeurs.	Tester le bouton Ajouter au panier ou appeler la fonction en omettant que Le produit ne pourrait pas étre ajouté ou sélectionner une option	Le produit pourrait ne pas étre ajouté ou sélectionner une option localStorage.	Voir
Fichiers	Lignes	Fonctions Testées	Résultat attendu	Comment vérifier le résultat attendu	Problème possible	Validé
basket.js	Ligne 8 à 39	onLoadBasket	La fonction récupère le dernier élément du local storage et regroupe les mêmes cameras par leur ID, puis les affiche dans le Panier.html (basket)	Il faut carrément choisir deux ou trois produits et voir si nous les retrouvons sur la page panier (basket.html)	Les informations sont erronées ou le tableau est vide	Voir
basket.js	Ligne 42 à 57	deleteItem	La fonction permet de vider le panier , afficher un message 'votre panier est vide ' et redirection vers la page d'accueil, et si le panier contient tjrs des items sélectionnés, il fait un reload du panier pour afficher ces derniers.	Il faut Tester l'écouteur d'événement en cliquant sur le bouton supprimer pour vider et supprimer l'article choisi du panier	Le panier pourrait ne pas se vider / ou la redirection vers la page d'accueil peut ne pas se faire	Voir
basket.js	Ligne 60 à 66	getOrderId	Fonction pour obtenir un Id de commande / au clic du bouton, on arrive sur la page de confirmation affichant un numéro de commande aléatoire	il suffit juste de valider sa commande pour atterrir sur la page confirmation qui doit contenir le numéro de la commande	la page ne se charge pas ou le numéro de commande ne soit pas généré	Voir
basket.js	Ligne 69 à 99	handleSubmit(event)	l'envoi du formulaire qui declanche la redirection de la page	il suffit juste de valider sa commande pour atterrir sur la page confirmation qui doit contenir le numéro de la commande ainsi que son total payé	Le panier est vide et pas génération de ligne de commande, et les donnés de formulaire ne sont pas rempli (vide)	Voir
basket.js	Ligne 101 à 116	sendCommandToServer	Redirection vers la page de confirmation	Clic sur le bouton de validation pour faire la redirection de page	la redirection ne se fait pas	Voir
Fichiers	Lignes	Fonctions Testées	Résultat attendu	Comment vérifier le résultat attendu	Problème possible	Validé
confirmation.js	Ligne 02 à 29	onLoadCommand	Affichage du message de confirmation de la commande, avec un numéro de commande, et le total de cette dernière, tout en supprimant le panier, c'est a dire le local storage	c'est simple, il faut juste passer commande	La valeur retournée pourrait ne pas correspondre au résultat attendu	Voir