

## PLAN DE TESTS – ONORICO

Plan de test portant sur 4 pages de code JS, une page de code CSS et 4 pages de code HTML conformément aux spécifications fonctionnelles et techniques

Fichier Javascript	Lignes de code testées	Test unitaire	Description du test	Résultat attendu	Résultat
views\onorico.js	Ligne 3 à 46 (appel de l'API via fetch)	Récupération de la liste des ours depuis l'API	Vérification de la bonne récupération de tous les ours depuis l'API au moyen d'un GET, comparaison de la liste affichée avec la liste que l'on retrouve sur le document teddy.js dans le dossier serveur ; vérification de la bonne correspondance nom ours/photo/prix.	Affichage de tous les ours disponibles sous forme de liste sur la page d'accueil	OK
views\onorico.js	Ligne 3 à 5 ; ligne 44 à 46	Vérification de la disponibilité de l'API	Déconnection du serveur local.	Affichage d'un message d'erreur en cas de problème d'accès à l'API ; affichage de la liste des ours sinon	OK
views\produit.js	Lignes 33, 34	Récupération des données grâce à un ID	Vérification de la récupération du bon ours avec caractéristiques correspondantes (nom, photo, prix, couleurs) en comparant le numéro d'ID présent dans l'URL lors de l'affichage de la page produit.	Affichage de la fiche produit ours correspondante lors du click sur un ours depuis la page d'accueil	OK
views\panier.js	Lignes 28 à 85	Affichage adapté du contenu du panier	Test au click depuis les pages d'accueil et pages produits : click sur panier avant d'avoir ajouté des produits ; puis click sur panier une fois plusieurs produits ajoutés	Affichage du message « Votre panier est vide » ou du contenu du panier dans un tableau	OK
views\produit.js	Lignes 60 à 73	Fonctionnement du local Storage pour le stock des produits ajoutés au panier	Ajout de plusieurs ours au panier ; ajout d'ours similaires (noms et couleurs) afin de s'assurer que la quantité de l'ours en question est bien indentée dans le tableau « products » du local storage	Récupération de la liste des produits ajoutés au panier dans le local storage et de la quantité de chaque	OK
views\produit.js	Lignes 60 à 73	Récupération des données depuis le local storage pour mise à jour du tableau « votre panier »	Comparaison du contenu du local storage avec le tableau des produits afin de s'assurer qu'il est similaire ; suppression du contenu du local storage afin de s'assurer que le tableau disparaît et que le message « votre panier est vide » apparaît	Affichage dans le tableau des produits, sur la page panier, de tous les ours ajoutés au panier avec les quantités correspondantes ; affichage du message si panier vide	OK
views\produit.js views\panier.js	Lignes 75 à 79 Lignes 69 à 78	Calcul automatique des quantités et du prix total du panier	Ajout de plusieurs ours dans le panier, à la fois identique, mais aussi différents	Affichage du bon montant pour chaque ligne et pour le « TOTAL » en dessous du local panier	OK

views\panier.js	Ligne 96	Fonctionnement du formulaire de contact – remplissage des champs	Click sur « valider » en omettant le remplissage d'un champ (test avec chacun des champs) ; click sur « valider » après avoir rempli tous les champs	Message d'erreur en cas de non-remplissage d'une case, affichage de la page commande sinon	OK
utils\fonctions.js views\panier.js	Lignes 2 à 9 ; Ligne 98	Fonctionnement du formulaire de contact – format correct de l'email	Click sur « valider » en écrivant mal l'adresse email : pas @ ou pas de .fr/com... à la fin ; click sur valider après avoir correctement renseigné l'email	Message d'erreur en cas de mauvais format d'email, affichage de la page commande sinon	OK
utils\fonctions.js views\panier.js	Lignes 12 à 16 ; Ligne 97 ;	Fonctionnement du formulaire de contact – format correct du nom, prénom, ville	Seuls les lettres, - et espaces sont acceptés dans ces champs : Click sur « valider » en ajoutant des caractères spéciaux ou des lettres ; click sur « valider » en indiquant un nom de ville avec un espace, et un nom contenant un espace	Message d'erreur en cas de mauvais format (contient chiffres ou caractères spéciaux), affichage page commande si lettres, - ou espace	OK
views\ commande.js + entity\ commande.js	Lignes 7, 8 et 16, 17 du fichier views	Récupération des données depuis le local storage pour affichage du montant total des achats	Ajout de plusieurs éléments au panier afin d'avoir un montant total calculé automatiquement sur la page panier, remplissage correct du formulaire puis click sur « valider commande »	Affichage de la confirmation de commande avec montant total du panier correct	OK
views\ commande.js + entity\ commande.js	Lignes 7, 8 et 13, 14 du fichier views	Récupération du numéro de commande depuis l'API	Click sur « valider commande » après avoir ajouté un ou plusieurs éléments au panier et remplissage correct du formulaire	Affichage de la confirmation de commande avec numéro de commande envoyé par l'API	OK