

Fichier JS	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
Router.js	39 à 41	_pageNameParser	Retourne une chaîne de caractère : nom du fichier (sans extension) ou une chaîne vide	Utiliser différentes URL et utiliser un console.log de la valeur retournée	Si l'URL appelée n'existe pas, le serveur renvoie une erreur 404
Router.js	24 à 33	_apiRequestUrl	Retourne une chaîne de caractère en concaténant la chaîne passée en paramètre (propriété de la classe parente) et la valeur du paramètre GET « id » si la fonction _pageNameParser retourne « produit »	Utiliser différentes URL avec ou sans paramètre GET « id » et utiliser un console.log de la valeur retournée	Si l'URL appelée n'existe pas, le serveur renvoie une erreur 404. Si l'URL est bonne mais que le paramètre GET « id » ne correspond à aucune entrée sur le serveur, peut renvoyer un message d'erreur ou ne rien renvoyer
HtmlFactory.js	14 à 57	showProductFromList	Affiche un élément HTML dépendant de la réponse du serveur passée en paramètre	Lancer la fonction depuis le script	L'élément pourrait ne pas s'afficher correctement, ou pas avec les bonnes informations
HtmlFactory.js	64 à 170	showOneProduct	Affiche un élément HTML dépendant de la réponse du serveur passée en paramètre	Lancer la fonction depuis le script	L'élément pourrait ne pas s'afficher correctement, ou pas avec les bonnes informations
HtmlFactory.js	177 à 234	showCart	Affiche le contenu du panier si il contient des produits ainsi qu'un formulaire pour valider la commande, sinon affiche un message proposant de consulter le catalogue	Lancer la fonction avec LocalStorage video, et avec la 'cart' de LocalStorage renseignée	Peut générer plusieurs erreurs si la valeur de la clé 'cart' est mal renseignée
HtmlFactory.js	239 à 256	showEmptyCart	Affiche le contenu de la page panier lorsque la clé 'cart' dans LocalStorage est vide	Lancer la fonction depuis le script	L'élément pourrait ne pas s'afficher correctement, ou pas avec les bonnes informations
HtmlFactory.js	261 à 285	showOrderConfirmation	Affiche les éléments HTML contenant la réponse du serveur après la validation de la commande (récupéré depuis LocalStorage). Si la clé correspondante n'existe pas, on doit être redirigé vers la page d'accueil	Ouvrir la page confirmation.html, lancer la fonction depuis le script avec et sans clé 'order' dans LocalStorage	Peut générer plusieurs erreurs d'affichage si la valeur de la clé 'order' est mal renseignée
HtmlFactory.js	287 à 300	loader	Affiche un loader en HTML	Lancer la fonction depuis le script	Les éléments pourraient ne pas s'afficher au bon endroit
HtmlFactory.js	309 à 369	showModal	Affiche une fenêtre modale pouvant être de plusieurs types	Lancer la fonction depuis le script en utilisant tous les paramètres suggérés dans les commentaires de la fonction.	Les boutons de la modale pourraient ne pas avoir le comportement attendu, notamment lors de l'utilisation d'une fonction callback dans la modale de confirmation
HtmlFactory.js	374 à 380	_hideModal	Cache la fenêtre modale	Lancer la fonction depuis le script lorsque la modale est affichée.	Effets de bord (par exemple : éléments devenus non interactifs...)
HtmlFactory.js	388 à 422	_createCartTable	Crée un tableau HTML devant regrouper les produits contenus dans le panier	Lancer la fonction depuis le script, faire un console.log de l'element HTML retourné	Le tableau pourrait ne pas être affiché correctement, ou au bon endroit
HtmlFactory.js	431 à 483	_createCartTableRow	Crée les lignes du tableau contenant les informations de chaque produit du panier	Lancer la fonction depuis le script, faire un console.log de l'element HTML retourné	L'élément pourrait ne pas s'afficher correctement, ou pas avec les bonnes informations
HtmlFactory.js	489 à 530	_createDetailsForm	Crée les champs de formulaire pour la validation de la commande		Les champs de formulaire pourraient ne pas être affichés correctement ou avec les bons attributs.
HtmlFactory.js	550 à 572	_showHideErrorElement	Affiche un élément « span » sous un champ de formulaire contenant une erreur en indiquant l'erreur en question, et encadre le champ en rouge	Créer volontairement une erreur dans un champ de formulaire	L'élément « span » pourrait ne pas s'afficher correctement ou déplacer certains éléments de son parent
HtmlFactory.js	586 à 595	_lockScroll	Doit empêcher le scrolling de la page	Vérifier le comportement du scroll de la page	Possibles effets de bord; L'affichage des éléments déjà existants peut être altéré
HtmlFactory.js	600 à 606	_unlockScroll	Doit rétablir le scrolling de la page web lorsque la fonction _lockScroll a déjà été appelée	Vérifier le comportement du scroll de la page	Possibles effets de bord; L'affichage des éléments déjà existants peut être altéré
Cart.js	14 à 24	addToCart	Doit ajouter un produit dans un array stocké avec la clé 'cart' dans LocalStorage	Faire un console.log de la clé 'cart' contenue dans LocalStorage	Le produit stocké dans LocalStorage pourrait ne pas être inséré avec les bonnes valeurs et/ou ne pas incrémenter le produit similaire pré-existant
Cart.js	31 à 70	deleteItemFromCart	Doit supprimer un produit dans l'array stocké avec la clé 'cart' dans LocalStorage; doit également supprimer la ligne correspondante du tableau HTML et recalculer les index des lignes suivantes	Vérifier les lignes du tableau HTML; faire une console.log de la valeur de l'attribut 'data-index' pour le lien 'supprimer' de chaque ligne; faire un console.log de la clé 'cart' contenue dans LocalStorage	La ligne du tableau pourrait ne pas être supprimée; les index des liens pourraient ne pas être calculés correctement; le contenu de LocalStorage pourrait ne pas être mis à jour correctement
Cart.js	76 à 92	headerNotification	Affiche une info bulle sur l'icone « panier » du header si la clé 'cart' du LocalStorage contient un array de produits. Masque cette info bulle si le clé ne contient rien ou n'existe pas	Vérifier l'affichage de l'info bulle	L'info bulle pourrait ne pas être affichée au bon endroit ou afficher un nombre de produits erroné
Cart.js	115 à 150	buyCartContent	Si le contenu du formulaire de commande est valide, doit soumettre l'ensemble des données (contenu du panier et informations client) au backend, et rediriger vers la page confirmation.html après avoir retiré la clé 'cart' du LocalStorage et créé une clé 'order'	Effectuer un console.log du LocalStorage au début et à la fin de la méthode; vérifier que l'on est bien redirigé vers la page confirmation.html	
Cart.js	158 à 169	totalPrice	Calcule le prix total (en centimes) des produits contenus dans le panier	Lancer la fonction depuis le script en lui passant l'URL de l'api et la le contenu de la clé 'cart' du LocalStorage en paramètre; effectuer un console.log de la valeur retournée	Le calcul pourrait être faux
Cart.js	177 à 187	_setNewCartItem	Crée un nouvel objet à ajouter au panier, à partir du formulaire permettant d'ajouter un produit au panier	Effectuer un console.log de l'objet retourné	L'objet créé pourrait ne pas avoir les bonnes valeurs
Cart.js	194 à 208	_updateCartItem	Ajoute un produit supplémentaire au panier; si ce produit existe déjà, incrémente sa quantité	Effectuer un console.log de la valeur retournée	Le produit pourrait ne pas être ajouté au LocalStorage, remplacer un produit déjà existant, ou mal incrémenter la quantité
Validator.js	46 à 55	validate	Compare les valeurs entrées en paramètre avec une regex en fonction du type de valeur à analyser	Lancer la fonction depuis le script en utilisant des valeurs différentes (notamment des valeurs limites) pour chaque type de valeur; faire un console.log de l'objet retourné	Certaines valeurs pourraient passer la validation alors qu'elles ne devraient pas; l'erreur renvoyée pourrait ne pas être la bonne