

# Plan de test Orinoco :

## INTRODUCTION :

AUTHOR		CONTACT DETAILS		FINAL DATE FOR COMMENTS	
Christine Endres		//		//	
<b>PRODUCT UNDER TEST</b> What's being tested? What are the business and experience goals of the product? MVP du site Orinoco. Ce site a vocation a devenir un site e-commerce.	<b>TEST OBJECTIVES</b> What are the goals of the usability test? What specific questions will be answered? What hypotheses will be tested? L'objectif est de vérifier que les différentes fonctions permettant au site d'être interactif et dynamique fonctionnent: affichage des produits, du panier et process de commande notamment. Cela va permettre de tester les cas les plus courants qui seront rencontrés en se questionnant sur les usages que feront les clients du site. Les hypothèses d'usage "normal" et de quelques cas extrêmes d'utilisation seront testés (dysfonctionnement réseau, tentative d'envoi de commande sans remplir le formulaire...).	<b>PARTICIPANTS</b> How many participants will be recruited? What are their key characteristics? L'idéal serait que le plan de test soit écrit par une autre personne que celle qui a rédigé le code; voir qu'une troisième personne effectue ces tests.	<b>TEST TASKS</b> What are the test tasks? Seuls des tests unitaires, voir quelques tests d'intégration, ou tout au moins "multi-fonctions" ont été prévus ici; pas de tests système. Chaque tests unitaires vise à vérifier le fonctionnement d'une fonction développée. Toutes les fonctions sont testée au moins dans leur cas simple; jusqu'à 4 tests ont été effectués par fonction pour atteindre un test coverage de 80%. Certains cas extrêmes n'ont peut-être pas été testés (notamment sur les requêtes serveur); un point de vue extérieur serait à ce sujet le bienvenu. Le plan de test ne couvre pas Internet Explorer.	<b>RESPONSIBILITIES</b> Who is involved in the test and what are their responsibilities? Le développeur front est responsable de ces tests mais un plan de test sur le backend est aussi à concevoir, pour combler les failles de sécurité liées à la possibilité de modifier le html notamment.	<b>LOCATION &amp; DATES</b> Where and when will the test take place? When and how will the results be shared? Les premiers tests ont été faits; il faudra les relancer en cas de modification du code; les résultats sont consultables sur la page de test. D'autres tests pourraient être effectués avant la suite du développement .
<b>PROCEDURE</b> What are the main steps in the test procedure?					
<div><div>Définir le périmètre du test.</div><div>Ecrire le plan de test</div><div>Rédiger les test unitaires</div><div>Tester</div><div>Corriger les éventuelles anomalies détectées</div><div>Retester et valider le MVP</div></div>					

## TESTS UNITAIRES :

1- Tests de la méthode `add` permettant d'ajouter des produits au panier prenant 4 paramètres : `id` , identifiant du produit ; `name`, nom du produit ; `qty`, nombre d'unités ajoutées. S'appuie sur la méthode `save` qui sauvegarde ce panier ; ces deux fonctions seront testées conjointement.

Intitulé	Test ajout objet complet
Code	testAdd1
Description	Objectif vérifier qu'on peut ajouter un produit dans le panier
Mode opératoire	Vider panier en début de test Ajouter un produit au panier
Résultat attendu	Le panier contient un unique produit dont les caractéristiques correspondent à ce qui a été ajouté

Intitulé	Test ajout objet incomplet
Code	testAdd2
Description	Objectif vérifier qu'on ne peut ajouter un produit dans le panier s'il est incomplet
Mode opératoire	Vider panier en début de test Ajouter un produit incomplet au panier
Résultat attendu	Le panier est vide

Intitulé	Test ajout objet vide
Code	testAdd3
Description	Objectif vérifier qu'on ne peut ajouter un produit vide dans le panier
Mode opératoire	Vider panier en début de test Ajouter un produit vide au panier
Résultat attendu	Le panier est vide

Intitulé	Test ajout objet avec trop de paramètres
Code	testAdd4
Description	Objectif vérifier qu'on ne peut ajouter un produit avec trop de paramètres dans le panier
Mode opératoire	Vider panier en début de test Ajouter un produit avec trop d'arguments au panier
Résultat attendu	Le panier est vide

2- Test de la méthode count comptant le nbe de produits différents dans le panier et retournant la longueur du tableau [produits] :

Intitulé	Test comptage panier de 1 élément
Code	testCount1
Description	Objectif vérifier qu'il y a bien un produit dans le panier
Mode opératoire	Vider panier en début de test Ajouter un produit au panier Compter le nombre de produits différents dans le panier
Résultat attendu	La longueur est de 1

Intitulé	Test comptage panier de 1000 éléments
Code	testCount2
Description	Objectif vérifier qu'il y a bien 1000 produits dans le panier
Mode opératoire	Vider panier en début de test Ajouter 100 produits au panier Compter le nombre de produits différents dans le panier
Résultat attendu	La longueur est de 1000

Intitulé	Test comptage panier vide
Code	testCount3
Description	Objectif vérifier qu'il y a bien 0 produits dans le panier
Mode opératoire	Vider panier en début de test Compter le nombre de produits différents dans le panier
Résultat attendu	La longueur est de 0

3- Test de la méthode findId recherchant un produit dans le panier à partir de son id et en renvoyant la position :

Intitulé	Test recherche d'un produit dans le panier à partir de son id
Code	testFindId1
Description	Objectif vérifier que la position est bien retournée avec un produit unique
Mode opératoire	Vider panier en début de test Ajout d'un produit
Résultat attendu	La position 0 est bien retournée

Intitulé	Test recherche d'un produit dans le panier à partir de son id
Code	testFindId2
Description	Objectif vérifier que la position est bien retournée avec plusieurs produits
Mode opératoire	Vider panier en début de test Ajout de 3 produits dont le produit recherché au milieu
Résultat attendu	La position 1 est bien retournée

Intitulé	Test recherche d'un produit dans le panier à partir de son id
Code	testFindId3
Description	Objectif vérifier que la position n'est pas retournée quand le produit n'y est pas
Mode opératoire	Vider panier en début de test Ajout de 3 produits sans le produit recherché
Résultat attendu	La position 1 est bien retournée

4- Test de la méthode get recherchant un produit dans le panier à partir de son id et le retournant

Intitulé	Test consultation d'un produit du panier à partir de sa position avec un seul produit
Code	testGet1
Description	Objectif vérifier que le bon produit est retourné
Mode opératoire	Vider panier en début de test Ajouter un produit Appeler le produit par son id
Résultat attendu	Le produit demandé par son id est bien renvoyé

Intitulé	Test consultation d'un produit du panier à partir de sa position avec plusieurs produits
Code	testGet2
Description	Objectif vérifier que le bon produit est retourné
Mode opératoire	Vider panier en début de test Ajouter 3 produits Appeler un produit présent par son id
Résultat attendu	Le produit demandé par son id est bien renvoyé

Intitulé	Test consultation d'un produit absent du panier à partir de sa position
Code	testGet3
Description	Objectif vérifier qu'aucun produit n'est retourné
Mode opératoire	Vider panier en début de test Ajouter 3 produits Appeler un produit absent par son id
Résultat attendu	Le produit demandé par son id est bien « null »

5- test de la méthode `setQty` modifiant la quantité si le produit est déjà présent dans panier, à partir de sa position ; un paramètre : `newQty`

Intitulé	Test modification quantité dans le panier
Code	testsetQty1
Description	Objectif vérifier que la quantité est bien modifiée pour le produit présent
Mode opératoire	Vider panier en début de test Ajouter un produit En modifier la quantité Récupérer ce produit
Résultat attendu	Vérifier qu'on a bien la nouvelle quantité dans le panier

Intitulé	Test modification quantité dans le panier
Code	testsetQty2
Description	Objectif vérifier que la quantité est bien modifiée pour le produit choisi
Mode opératoire	Vider panier en début de test Ajouter 3 produits Modifier la quantité d'un des produits à partir de sa position
Résultat attendu	Vérifier qu'on a bien la nouvelle quantité dans le panier pour le bon produit

#### 6- test de la méthode load récupérant les produits dans le panier

Intitulé	Test récupération d'un produit du panier
Code	TestLoad1
Description	Objectif vérifier qu'un produit est bien chargé
Mode opératoire	Vider panier en début de test Ajouter 1 produit Sauvegarder le panier Vider le panier Vérifier la présence du produit dans le panier après load
Résultat attendu	Vérifier qu'on a le produit ajouté dans le panier

Intitulé	Test récupération des produits du panier
Code	TestLoad2
Description	Objectif vérifier que les produits sont bien chargés
Mode opératoire	Vider panier en début de test Ajouter 3 produits Sauvegarder le panier Vider le panier Vérifier la présence des produits dans le panier après load
Résultat attendu	Vérifier qu'on a les produits ajoutés dans le panier

Intitulé	Test reconstruction panier vide
Code	testLoad3
Description	Objectif vérifier que le panier est initialisé lorsqu'il n'y a rien dans localStorage
Mode opératoire	Vider localStorage et le panier Vérifier la présence d'un panier vide après load
Résultat attendu	Vérifier qu'on a un panier de longueur 0 ;

#### 7- test de la méthode display affichant le résumé du panier en haut de page

Intitulé	Test affichage résumé panier
Code	TestdDisplay1
Description	Objectif vérifier que la table panier contient bien le résumé du panier
Mode opératoire	Vider le panier Ajouter un produit et sauvegarder le panier Vérifier la construction de la table panier après display
Résultat attendu	Vérifier qu'on a une table affichant nom et quantité du produit ajouté au panier avec les bons éléments et attributs

Intitulé	Test affichage résumé panier
Code	TestdDisplay2
Description	Objectif vérifier que la table panier contient bien le résumé du panier
Mode opératoire	Vider le panier Ajouter 3 produits et sauvegarder le panier Vérifier la construction de la table panier après display
Résultat attendu	Vérifier qu'on a une table affichant nom et quantité des produits ajoutés au panier avec les bons éléments et attributs

Intitulé	Test affichage résumé panier
Code	TestdDisplay3
Description	Objectif vérifier que rien n'est affiché quand le panier est vide
Mode opératoire	Vider le panier Changer son id Ajouter un produit Vérifier que display ne s'exécute pas car panier null (id#)
Résultat attendu	Vérifier qu'on a une table vide

#### 8- Test méthode refresh qui va effacer panier puis le reconstruire

Intitulé	Test affichage panier mis à jour
Code	TestdRefresh1
Description	Objectif vérifier que le panier est bien reconstruit
Mode opératoire	Vider le panier Ajouter un produit Sauvegarder le panier Refresh
Résultat attendu	Vérifier que le panier affiché est bien le bon

Intitulé	Test affichage panier mis à jour
Code	TestdRefresh2
Description	Objectif vérifier que le panier est bien reconstruit
Mode opératoire	Vider le panier Ajouter 3 produits Sauvegarder le panier Refresh
Résultat attendu	Vérifier que le panier affiché est bien le bon

9- Test **méthode save** qui sauvegarde le panier dans localStorage

Intitulé	Test sauvegarde panier
Code	TestSave
Description	Objectif vérifier que le panier est bien sauvegardé dans localStorage
Mode opératoire	Vider le panier Ajouter 1 produit Vider localStorage Sauvegarder le panier Regarder le contenu de localStorage
Résultat attendu	Panier est bien sauvegardé dans localStorage

10- Test **méthode clear** qui vide le panier

Intitulé	Test vidange panier
Code	TestClear
Description	Objectif vérifier que le panier est bien vidé
Mode opératoire	Ajouter 1 produit Vider le panier
Résultat attendu	Panier est bien vide

11- Test **fonction ajout\_panier** qui ajoute des produits dans le panier au clic sur "Ajouter au panier"

Intitulé	Test fonction ajout panier
Code	TestAjoutPanier1
Description	Objectif vérifier que le produit ajouté est bien dans le panier
Mode opératoire	Vider le panier et le localStorage Ajouter 1 produit avec ajout_panier()
Résultat attendu	Panier contient bien le produit ajouté

Intitulé	Test fonction ajout panier
Code	TestAjoutPanier1
Description	Objectif vérifier que le produit ajouté est bien dans le panier
Mode opératoire	Vider le panier et le localStorage Ajouter un produit avec add() Ajouter 1 produit de même id avec ajout_panier()
Résultat attendu	Panier contient bien le produit ajouté avec la quantité modifiée



### 12- Fonction panier\_commande qui construit le panier et calcul le montant de la commande

Intitulé	Test fonction construction le panier et calcul montant commande
Code	TestPanierCommande1
Description	Objectif vérifier la construction de la table de commande
Mode opératoire	Vider le panier et le localStorage Vider la table de produits Ajouter un produit Aller au panier
Résultat attendu	Table de commande avec le bon produit et les bons montants

Intitulé	Test fonction construction le panier et calcul montant commande
Code	TestPanierCommande2
Description	Objectif vérifier la construction de la table de commande
Mode opératoire	Vider le panier et le localStorage Vider la table de produits Ajouter 3 produits Aller au panier
Résultat attendu	Table de commande avec les bons produits et les bons montants

Intitulé	Test fonction construction le panier et calcul montant commande
Code	TestPanierCommande3
Description	Objectif vérifier la non construction de la table de commande
Mode opératoire	Vider le panier et le localStorage Vider la table de produits Aller au panier
Résultat attendu	Table de commande vide

### 13- Fonctions verifierTexte et isValid qui vérifient le formulaire avant validation

Intitulé	Test fonction vérification du formulaire avant validation
Code	TestVerifier1
Description	Vérifier que le formulaire n'est pas validé si l'une des cellules est vide ou l'email au mauvais format
Mode opératoire	Vider le formulaire Lancer la vérification Vérifier qu'un message d'erreur s'affiche à chaque endroit et que le formulaire n'est pas validé
Résultat attendu	Le bon message d'erreur s'affiche à chaque endroit

Intitulé	Test fonction vérification du formulaire avant validation
Code	TestVerifier2
Description	Vérifier que le formulaire est validé si les cellules sont correctement remplies
Mode opératoire	Vider le formulaire Lancer la vérification Vérifier qu'aucun message d'erreur ne s'affiche et que le formulaire est validé
Résultat attendu	Le bon message d'erreur s'affiche à chaque endroit

14- **Fonction commande\_valide** qui, une fois la commande validée, récupère le n° de commande et le montant à payer

Intitulé	Test fonction qui une fois la commande validée, récupère le n° de commande et le montant à payer
Code	TestCommandeValide1
Description	Vérifier qu'à l'envoi de la commande l'id et le montant sont bien récupérés pour être affichés sur le page commande
Mode opératoire	Lancer la validation Vérifier que la fonction renvoi bien l'id de la commande et son montant (contenus dans localStorage)
Résultat attendu	Affichage id commande et montant commande sur page commande

15 - **Fonction envoi\_commande**, qui envoie la commande si formulaire complet et valide

Intitulé	Test fonction qui envoie la commande si formulaire complet et valide
Code	TestEnvoi_commande1
Description	Vérifier qu'au clic sur « Envoyer la commande » une requête POST est bien envoyée et que fonction commande_valide appelée si réponse 200 reçue
Mode opératoire	Compléter panier et formulaire correctement Vérifier connexion au « serveur » Clic sur « Envoyer la commande » Vérifier que la fonction renvoi bien 200 et la réponse attendue
Résultat attendu	Affichage id commande et montant commande sur page commande

Intitulé	Test fonction qui envoie la commande si formulaire complet et valide
Code	TestEnvoi_commande2
Description	Vérifier qu'au clic sur « Envoyer la commande » une requête POST est bien envoyée et que fonction commande_valide n'est pas appelée si réponse 400 reçue
Mode opératoire	Compléter panier et formulaire correctement Déconnecter le « serveur » Clic sur « Envoyer la commande » Vérifier que la requête POST renvoie bien 400 et un message d'erreur
Résultat attendu	Affichage message d'erreur et pas d'envoi de la commande

16 -Fonction load and co., qui chargent les fiches produit sur la page d'accueil

Intitulé	Test fonction affichage fiches ours
Code	TestAffichage1
Description	Vérifier qu'au chargement de la page index les fiches produits sont bien affichées
Mode opératoire	Vérifier connexion au « serveur » Ouvrir index.html Vérifier que la fonction requête GET renvoie bien 200 et la réponse attendue
Résultat attendu	Affichage fiches produits sur page d'accueil

Intitulé	Test fonction affichage fiches ours
Code	TestAffichage2
Description	Vérifier qu'au chargement de la page index les fiches produits ne sont pas affichées si pas de connexion au serveur
Mode opératoire	Intérompre connexion au « serveur » Ouvrir index.html Vérifier que la fonction requête GET renvoie bien 400 et pas de fiche produit
Résultat attendu	Non affichage fiches produits sur page d'accueil

17 -Fonction getById and co., qui chargent la fiche du produit sélectionné sur la page de personnalisation

Intitulé	Test fonction affichage fiche ours sélectionné
Code	TestAffichage3
Description	Vérifier qu'au chargement de la page perso.html, la fiche produit de l'ours sélectionné est bien affichée
Mode opératoire	Vérifier connexion au « serveur » Cliquer sur un ours dans index.html Vérifier que la fonction requête GET renvoie bien 200 et la réponse attendue
Résultat attendu	Affichage fiche produit sélectionné sur page perso

Intitulé	Test fonction affichage fiche ours sélectionné
Code	TestAffichage4
Description	Vérifier qu'au chargement de la page perso.html, la fiche produit de l'ours sélectionné n'est pas affichée
Mode opératoire	Charger index.html Déconnecter le « serveur » Cliquer sur un ours dans index.html Vérifier que la fonction requête GET renvoie bien 400 et pas de fiche produit
Résultat attendu	Non affichage fiche produit sélectionné sur page perso