

Rapport de testing

I. Recommandations

A. Tests à automatiser

Après avoir testé manuellement les fonctionnalités décrites par Marie dans son bilan de campagne de tests de validation initial, je recommande d'automatiser les tests sur les fonctionnalités suivantes :

● Connexion

La connexion est une fonctionnalité fréquemment utilisée et n'est pas amenée à évoluer. Elle est peu complexe à tester et l'écriture des scripts prendra donc peu de temps.

● Panier

Détection d'anomalies dont certaines sont particulièrement impactantes et nécessitent une correction :

- possibilité de dépasser la limite autorisée en ajoutant au panier une quantité supérieure à 20 pour un produit et commande validée dans l'API,
- possibilité d'ajouter au panier une quantité d'articles supérieure au stock indiqué (il prend alors une valeur négative dans la description produit) et commande validée dans l'API ;
- possibilité d'ajouter au panier un article dont le stock est nul ou négatif et commande validée dans l'API.
- une requête sur le contenu du panier utilisateur sans être connecté renvoie un code 401 au lieu d'un code 403.
- la requête en POST pour l'ajout d'un produit au panier génère un code 405 (le serveur ne prend pas en charge cette méthode de requête HTTP).

L'automatisation des tests pour cette fonctionnalité permettra de garantir que les erreurs détectées ne seront pas réintroduites dans le code dans les prochaines versions ou pour l'ajout de nouveaux produits.

L'enjeu de ces deux fonctionnalités a un impact considérable pour le fonctionnement du site :

- Si un utilisateur ne peut se connecter, il ne peut réaliser d'achat ;

- Si le client peut ajouter au panier des produits dont le stock est épuisé, la commande ne pourra aboutir.

Cela peut engendrer une perte financière pour Eco Bliss Bath. L'image de la marque pourrait également être impactée en occasionnant un manque de crédibilité auprès du client qui a passé sa commande.

B. Préconisations pour la suite

Si le budget et le temps le permettent, je préconise par la suite :

- de réaliser des tests sur **la fonctionnalité « Produit »** afin de vérifier côté API la conformité des données aux valeurs attendues ainsi que côté Front end pour vérifier l'affichage des informations. Étant donné qu'il s'agit d'un modèle répétitif, les tests pourront être automatisés afin de gagner du temps et de limiter la probabilité d'erreur (oubli d'un produit, d'un champs).
- de réaliser des tests sur **la fonctionnalité « Créer un compte »** avec des données de test valides et non valides, vérifier le fonctionnement des requêtes API afin de s'assurer qu'un compte a bien été créé, ou être averti de l'échec de la tentative de création de compte. L'automatisation des tests est à privilégier ici car il s'agit d'un scénario fixe et cela permettra de couvrir largement et rapidement les cas de test en utilisant un jeu de données varié.
- de poursuivre dans les prochaines versions les tests sur **la fonctionnalité « Panier »** en vérifiant le calcul de la somme à payer pour chaque produit au panier selon la quantité ajoutée et pour le total du panier.
- Étant donné les bugs rencontrés dans les tests de la fonctionnalité « Panier », je préconise par la suite de réaliser des **tests manuels** pour un **scénario d'achat produit complet** qui va jusqu'à la validation de commande. Il est indispensable de s'assurer que l'utilisateur puisse voir sa commande honorée par l'entreprise.

II. Bilan de campagne de validation : tests automatisés

Document revu par :

Nom	Fonction	Version	Signature
Hélène	Testeur	1.0.0	

Contexte

Eco Bliss Bath est une start-up de 20 personnes, spécialisée dans la vente de produits de beauté écoresponsables dont le produit principal est un savon solide. La boutique prépare un site de vente en ligne. Une première version a été développée par l'équipe des développeurs et vient d'être testée manuellement par la QA testeuse. En tant que QA Engineer, je réponds à la demande du CTO pour la définition de l'automatisation des tests. Pour cette campagne, je me suis basée sur le bilan de campagne des tests manuels ainsi que sur les contraintes budgétaires évoquées par le PO en charge du budget. Les tests ont été réalisés sur une semaine. L'environnement des tests est le suivant :

système d'exploitation : Windows 11 Famille version 24H2

navigateurs : Firefox version 131 et Chrome version Version 130.0.6723.60

logiciels de support aux tests : VS Code v1.94.2, Docker v4.34.3 et Cypress v13.15.0

Objectifs

Cette campagne de tests concerne les tests fonctionnels. L'objectif est de s'assurer que le site respecte les spécifications définies dans les exigences, qu'il fonctionne correctement et répond aux besoins de l'utilisateur. Les tests permettront d'évaluer cette première version du produit, prévenir des défauts, déceler des anomalies et les corriger avant la mise en production de façon à réduire les coûts et s'assurer de la satisfaction du client.

Les 3 scénarios suivants seront testés indépendamment pour cette campagne :

- Les **tests API** seront entièrement automatisés afin de valider le travail de l'équipe de conception et garantir le bon fonctionnement de cette première version. Ainsi nous chercherons à déceler d'éventuels bugs au niveau des endpoints, des données retournées et du traitement des requêtes.
- Parmi les principales fonctionnalités (connexion, affichage produits et panier), les deux plus critiques seront soumises à des tests automatisés. Ainsi nous nous assurerons que la fonctionnalité de **connexion** qui permet de réaliser des achats n'est pas en échec, et que le **panier** fonctionne correctement afin de garantir la réalisation d'une commande valide.
- Les **smoke tests** permettront de vérifier que la VI est fonctionnelle à travers des tests automatisés rapides qui valideront les fonctionnalités de base du site et permettront de s'assurer de leur stabilité lors de l'ajout de nouvelles fonctionnalités.

Selon le temps restant, des tests seront effectués pour vérifier la sécurité en identifiant d'éventuelles failles XSS dans le formulaire pour déposer un avis.

Tests effectués

Tests API

CONNEXION

Se connecter via l'API : <http://localhost:8081/login>

Méthode **POST**

1) Vérification de la **connexion avec identifiant et mot de passe CORRECTS** :

email : test2@test.fr

mot de passe : **testtest**

→ doit renvoyer un code 200

2) Vérification de la **connexion avec identifiant et mot de passe INCORRECTS** :

email : test@test.fr

mot de passe : **testtest**

→ doit renvoyer un code 401 (non authentifié)

PANIER

1) Requête sur les données confidentielles d'un utilisateur avant connexion :

<http://localhost:8081/orders>

Méthode **GET**

→ selon les tests manuels, renvoie un code 401 (accès à la ressource non autorisé)

2) Requête sur les données confidentielles d'un utilisateur après connexion :

!! Pour les test suivants, il faut être connecté !!

- Se connecter via l'API : <http://localhost:8081/login>

Méthode **POST**

email : test2@test.fr

mot de passe : **testtest**

→ doit renvoyer un code 200

Récupérer le token pour s'authentifier dans l'API.

- Requête de la liste des produits du panier : <http://localhost:8081/orders>

Méthode GET

→ doit renvoyer un code 200

→ doit retourner la liste des produits mis au panier

- Ajouter un produit disponible au panier : <http://localhost:8081/orders/add>

Méthode PUT

Indiquer dans le corps de la requête :

l'id du produit (stock > 0) à ajouter

la quantité à ajouter : 1

→ doit renvoyer un code 200

Méthode POST

Indiquer dans le corps de la requête :

l'id du produit (stock > 0) à ajouter

la quantité à ajouter : 1

→ selon les tests manuels, renvoie un code 405

- Ajouter un produit indisponible au panier : <http://localhost:8081/orders/add>

Méthode PUT

Indiquer dans le corps de la requête :

l'id du produit (stock < 0) à ajouter

la quantité à ajouter : 1

→ doit renvoyer un code 400

Méthode POST

Indiquer dans le corps de la requête :

l'id du produit (stock < 0) à ajouter

la quantité à ajouter : 1

→ selon les tests manuels, renvoie un code 405

PRODUIT

Requête d'une fiche produit spécifique : <http://localhost:8081/products/{id}>

Méthode GET

Choisir un produit et indiquer son ID dans l'URL

→ doit renvoyer un code 200

→ doit retourner la fiche du produit correspondant à l'ID spécifié dans l'URL avec une valeur renseignée pour les champs « id, name, availableStock, skin, aromas, ingredients, description, price, picture, varieties ».

```
{
  "id": 5,
  "name": "Poussière de lune",
  "availableStock": 2,
  "skin": "Peau grasse",
  "aromas": "Musc",
  "ingredients": "Huiles végétales",
  "description": "Essayez notre savon aujourd'hui pour une expérience de bain rafraîchissante et revitalisante.",
  "price": 9.99,
  "picture": "https://cdn.pixabay.com/photo/2016/07/11/15/45/soap-1509963_960_720.jpg",
  "varieties": 6
},
```

AVIS

1) Se connecter via l'API : <http://localhost:8081/login>

Méthode POST

email : test2@test.fr

mot de passe : [testtest](#)

→ doit renvoyer un code 200

Récupérer le token pour s'authentifier dans l'API.

2) Ajouter un avis : <http://localhost:8081/reviews>

Méthode POST

Indiquer dans le corps de la requête :

title : ["Test : Produit validé!"](#),

comment : ["Test : odeur très agréable"](#),

rating : [5](#)

→ doit renvoyer un code 200

→ le corps de la réponse doit renvoyer un id pour l'avis posté, ainsi que le contenu indiqué dans la requête pour les champs « title / comment / rating ».

3) Vérifier que le formulaire ne contient pas de faille XSS :

- Se connecter :

Aller sur la page d'accueil

Dans la barre de navigation, cliquer sur le lien « Connexion »

Remplir les champs avec les données suivantes :

Email : « test2@test.fr »

Mot de passe : « [testtest](#) »

Cliquer sur le bouton « Se connecter »

- Déposer un avis :

Cliquer sur le lien « Avis »

Sélectionner une étoile

Remplir les champs avec les données suivantes :

Titre : « [Test vulnérabilité](#) »

Commentaire : « [<script>alert\("XSS"\)</script>](#) »

Cliquer sur « publier »

→ Dans l'avis publié le commentaire ne doit pas apparaître.

→ Aucune fenêtre d'alerte ne doit s'afficher.

CONNEXION

Se connecter avec des identifiants corrects

- 1) Aller sur la page d'accueil
- 2) Dans la barre de navigation, cliquer sur le lien « Connexion »
→ La page de connexion avec le formulaire s'affiche.
- 2) Dans le champ Email, entrer « test2@test.fr »
- 3) Dans le champs Mot de passe, entrer « testtest »
- 4) Cliquer sur le bouton « Se connecter »
→ En tant qu'utilisateur connecté, je suis redirigé vers la page d'accueil
→ Le lien vers le panier doit être visible dans la barre de navigation.

Se connecter avec des identifiants incorrects

- 1) Aller sur la page d'accueil
- 2) Dans la barre de navigation, cliquer sur le lien « Déconnexion »
- 3) Dans la barre de navigation, cliquer sur le lien « Connexion »
→ La page de connexion avec le formulaire s'affiche.
- 4) Dans le champ Email, entrer « test@test.fr »
- 5) Dans le champs Mot de passe, entrer « testtest »
- 6) Cliquer sur le bouton Se connecter
→ Un message doit alerter : « Identifiants incorrects ».
→ L'utilisateur reste sur la page de connexion.
→ Le lien vers le panier n'est pas visible dans la barre de navigation.

PANIER

Ajouter un produit disponible au panier et vérifier la mise à jour du stock

- 1) Se connecter au site web avec les identifiants de connexion utilisés précédemment.
- 2) Accéder au panier en cliquant sur le lien « Panier » dans la barre de navigation.
→ La page /commande/ s'ouvre avec les articles contenus dans le panier s'il n'est pas vide.
- 3) Vider le panier s'il ne l'est pas, en cliquant sur l'icône poubelle pour chaque article du panier.
→ Le message « Votre panier est vide. Consultez nos produits. » s'affiche dans la page /commande/.
- 4) Accéder à la page /Produits / en cliquant sur le lien « Produits » dans la barre de navigation.
→ La page /Nos produits/ s'affiche avec l'ensemble des produits.
- 5) Sélectionner un produit dont le stock est > 1 et cliquer sur « Consulter ».
→ La page du produit sélectionné s'affiche avec une description.
- 6) Cliquer sur « Ajouter au panier»
- 7) Cliquer sur le lien « Mon panier »
→ l'article et la quantité sélectionnée doivent apparaître
- 8) Retourner sur la page du produit en cliquant sur la flèche retour du navigateur
→ le stock doit être mis à jour (la quantité ajoutée au panier a été soustraite du stock précédent)

Vérifier le contenu du panier avec l'API :

- 1) Se connecter à l'API : <http://localhost:8081/login>
email : test2@test.fr
mot de passe : testtest
Récupérer le token pour s'authentifier dans l'API.
- 2) Faire une requête de la liste des produits du panier : <http://localhost:8081/orders>
Méthode **GET**
→ doit renvoyer un code 200
→ doit retourner l'article ajouté au panier (vérifier l'id) et la quantité sélectionnée

Vérification des limites :

- ajouter une quantité négative

- 1) Vider le panier s'il ne l'est pas, en cliquant sur l'icône poubelle pour chaque article du panier.
→ Le message « [Votre panier est vide. Consultez nos produits.](#) » s'affiche dans la page /commande/.
- 2) Accéder à la page /Produits / en cliquant sur le lien « Produits » dans la barre de navigation.
- 3) Sélectionner le produit « Poussière de lune » en cliquant sur « Consulter »
→ La page produit s'affiche
- 4) Entrer une quantité négative à ajouter au panier égale à -2
- 5) Cliquer sur « Ajouter au panier »
- 6) Cliquer sur le lien « Mon panier »
→ Le message « [Votre panier est vide. Consultez nos produits.](#) » s'affiche. L'article n'a pas été ajouté.

- ajouter une quantité > 20

- 1) Vider le panier s'il ne l'est pas, en cliquant sur l'icône poubelle pour chaque article du panier.
→ Le message « [Votre panier est vide. Consultez nos produits.](#) » s'affiche dans la page /commande/.
- 2) Accéder à la page /Produits / en cliquant sur le lien « Produits » dans la barre de navigation.
- 3) Sélectionner le produit « Extrait de nature » en cliquant sur « Consulter »
→ La page produit s'affiche
- 4) Effacer la quantité affichée dans le champs et entrer un nombre > 20 (21)
- 5) Cliquer sur « Ajouter au panier »
- 6) Cliquer sur le lien « Mon panier »
→ Le message « [Votre panier est vide. Consultez nos produits.](#) » s'affiche. L'article n'a pas été ajouté.

SMOKE TESTS

Vérifier la présence des champs et boutons de connexion

- 1) Aller sur la page d'accueil
- 2) Cliquer sur le lien « Connexion » dans la barre de navigation.
 - La page de connexion s'affiche avec le formulaire
 - Le champ « Email » doit être présent
 - Le champ « Mot de passe » doit être présent
 - Le bouton « Se connecter » doit être présent

Vérifier la présence des boutons d'ajout au panier pour un utilisateur connecté

- 1) Aller sur la page d'accueil
- 2) Cliquer sur le lien « Connexion » dans la barre de navigation.
- 3) Se connecter avec les données suivantes :
 - Email : « **test2@test.fr** »
 - Mot de passe : « **testtest** »Cliquer sur le bouton « Se connecter »
 - L'utilisateur est redirigé vers la page d'accueil
 - Le lien « Se déconnecter » doit être présent dans la barre de navigation
- 4) Cliquer sur le lien « Produits » dans la barre de navigation
- 5) Choisir un produit et cliquer sur « Consulter » pour afficher la fiche produit.
 - Le bouton « Ajouter au panier » doit être présent

Vérifier la présence du champ de disponibilité du produit

- 1) Cliquer sur le lien « Produits » dans la barre de navigation
- 2) Choisir un produit et cliquer sur « Consulter » pour afficher la fiche produit.
 - Un champ indiquant la quantité en stock doit apparaître

Résultats de tests

TESTS API

Connexion

- ✓ Vérification de la connexion avec identifiant et mot de passe CORRECTS
- ✓ Vérification de la connexion avec identifiant et mot de passe INCORRECTS

Panier

- ✓ Requête sur les données confidentielles d'un utilisateur avant connexion

⚠ → Attendu : erreur 401
→ Reçu : erreur 401 => devrait renvoyer 403 (accès à la ressource non autorisé)

Requête sur les données confidentielles d'un utilisateur après connexion :

- ✓ - Se connecter via l'API
- ✓ - Requête de la liste des produits du panier
- Ajouter un produit disponible au panier :

- ✓ Méthode PUT

- ✓ Méthode POST

- ⚠ → Attendu : code 405
→ Reçu : erreur 405

- Ajouter un produit indisponible au panier :

- ✗ Méthode PUT
→ Attendu : erreur 400
→ Reçu : code 200

- ✓ Méthode POST
→ Attendu : erreur 405

- ⚠ → Reçu : erreur 405

Produits

- ✓ Requête d'une fiche produit spécifique

⚠ Les données récupérées montrent une anomalie sur la valeur du stock pour certains produits (id 3, 4 et 9) : le stock est négatif.

Avis

- ✓ Se connecter via l'API
- ✓ Ajouter un avis
- ✓ Vérifier que le formulaire ne contient pas de faille XSS

TESTS DE FONCTIONNALITÉS

Connexion

- ✓ Se connecter avec des identifiants corrects
- ✓ Se connecter avec des identifiants incorrects

Panier

- ✓ Ajouter un produit disponible au panier et vérifier la mise à jour du stock
- ✓ Vérifier le contenu du panier avec l'API

Vérification des limites :

- ✓ - ajouter une quantité négative
- ✗ - ajouter une quantité > 20

→ Attendu : Le message « Votre panier est vide. [Consultez nos produits.](#) » s'affiche. Les articles n'ont pas été ajoutés.

→ Obtenu : Les articles sélectionnés sont ajoutés au panier.

SMOKE TESTS

- ✓ Vérifier la présence des champs et boutons de connexion
- ✓ Vérifier la présence des boutons d'ajout au panier pour un utilisateur connecté
- ✓ Vérifier la présence du champ de disponibilité du produit

=====

(Run Finished)

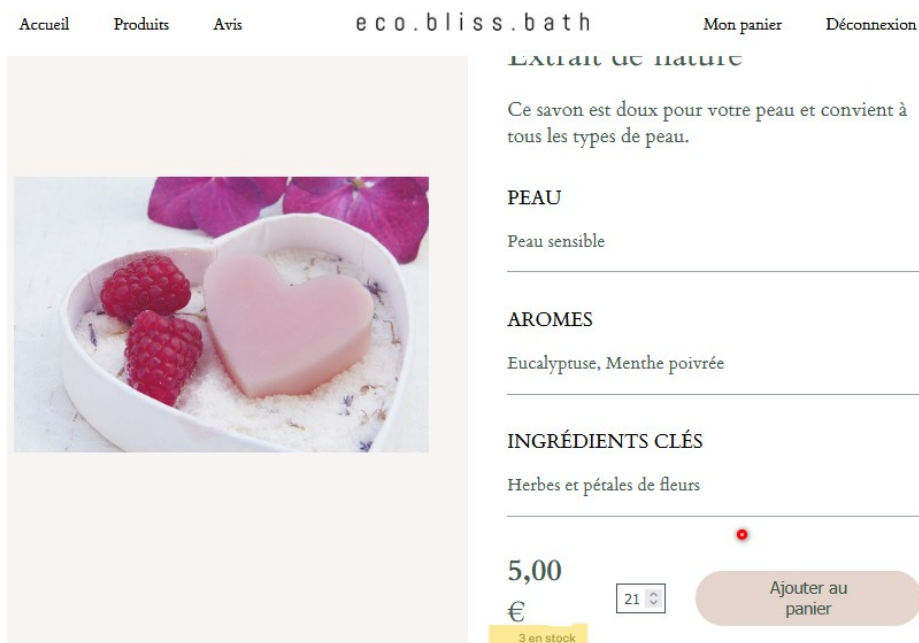
Spec		Tests	Passing	Failing	Pending	Skipped
✗ ShoppingCart/ShoppingCart.cy.js	00:18	5	4	1	-	-
✓ connexion/connexion.cy.js	00:04	2	2	-	-	-
✓ API/apiLogin.cy.js	00:01	2	2	-	-	-
✗ API/apiOrders.cy.js	00:04	6	5	1	-	-
✓ API/apiProducts.cy.js	233ms	1	1	-	-	-
✓ API/apiReviews.cy.js	00:01	2	2	-	-	-
✓ SmokeTests/smokeTests.cy.js	00:04	3	3	-	-	-
✗ 2 of 7 failed (29%)	00:33	21	19	2	-	-

Rapport d'incident d'anomalies

Anomalies majeures

* **FRONT END - Fonctionnalité PANIER** : ajouter une quantité > 20 pour un produit
→ 21 exemplaires du produit « Extrait de nature » dont le stock disponible était de 3 articles ont été ajoutés au panier.

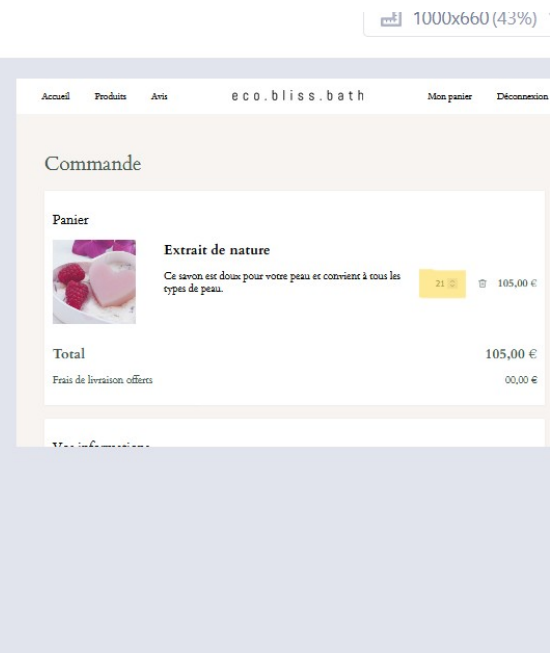
Il s'agit d'une anomalie critique car la commande ne pourra aboutir.



```
ShoppingCart cy.js 00:17

(new url) http://localhost:8080/#/cart
(xhr) OPTIONS 200 http://localhost:8081/me
(xhr) OPTIONS 200 http://localhost:8081/orders
16 wait @waitOrders
(xhr) GET 200 http://localhost:8081/me
(xhr) GET 200 http://localhost:8081/orders waitOrders
17 get [data-cy=cart-empty] 0
-assert expected [data-cy=cart-empty] to be visible
! AssertionError
Timed out retrying after 4000ms: Expected to find element: [data-cy=cart-empty] , but never found it.
cypress/e2e/ShoppingCart/ShoppingCart.cy.js:142:39

140 |   cy.intercept("GET", "http://localhost:8081/orders").as("waitOrders")
141 |   cy.wait("@waitOrders").then(()=>{
> 142 |     cy.getBySel("cart-empty").should("be.visible") // Vérifier que le panier
      |                                     ^
143 |   })
```



* BACK END - API méthode PUT : ajouter un produit indisponible

Dans le test API, l'article « Sentiments printaniers » (id 3) dont le stock est négatif a bien été ajouté au panier (code 200).

```
apiOrders cy.js

✗ Ajouter un produit non disponible (method PUT)
  BEFORE EACH
    1 request ○ POST 200 http://localhost:8081/login
  TEST BODY
    1 request ○ PUT 200 http://localhost:8081/orders/add
    2 -assert expected 200 to equal 400

! AssertionError

expected 200 to equal 400

cypress/e2e/API/apiOrders.cy.js:89:36

87 |     })
88 |     .then((response) => {
> 89 |       expect(response.status).to.eq(400)
    |                                     ^
90 |     })
91 |   })
92 |
```

Anomalies mineures

* BACK END – API méthode POST : ajouter un produit disponible / indisponible

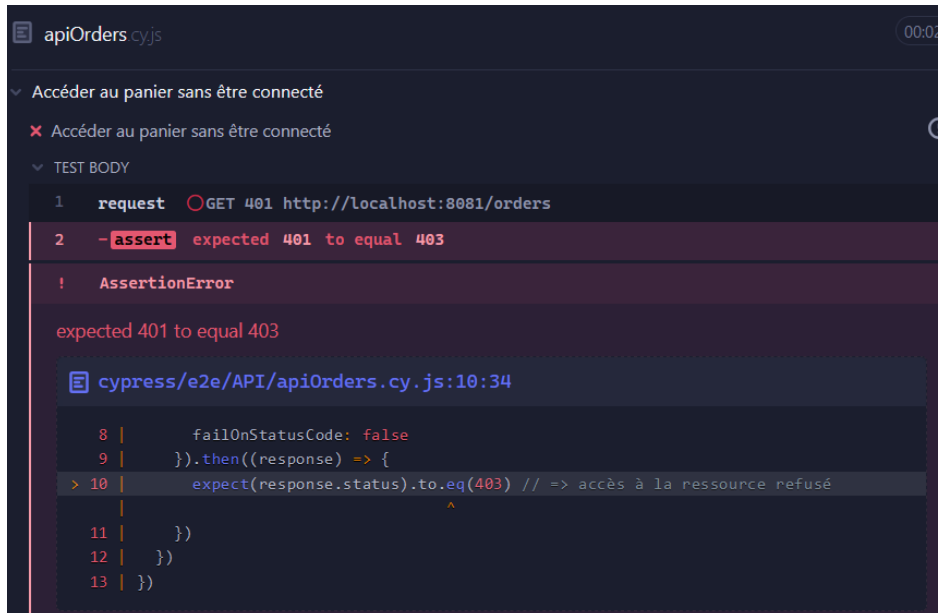
Les tests API sont en échec pour l'ajout d'un produit au panier avec la méthode POST : code 405 (Le serveur ne prend pas en charge la méthode de requête HTTP "POST"). Il s'agit d'une fonctionnalité future qui reste à développer.

```
✓ Ajouter un produit disponible au panier (method POST)
  BEFORE EACH
    1 request ○ POST 200 http://localhost:8081/login
  TEST BODY
    1 request ○ POST 405 http://localhost:8081/orders/add
    2 -assert expected 405 to equal 405
```

```
✓ Ajouter un produit non disponible (method POST)
  BEFORE EACH
    1 request ○ POST 200 http://localhost:8081/login
  TEST BODY
    1 request ○ POST 405 http://localhost:8081/orders/add
    2 -assert expected 405 to equal 405
```


* **BACK END – API méthode GET** : requête sur les données confidentielles d'un utilisateur avant connexion

Les tests API pour la requête sur les données confidentielles d'un utilisateur avant connexion retournent effectivement un code 401 (non authentifié) au lieu de 403 (accès à la ressource refusé).



```
apiOrders.cy.js 00:02
✓ Accéder au panier sans être connecté
✗ Accéder au panier sans être connecté
  TEST BODY
  1 request ○ GET 401 http://localhost:8081/orders
  2 -assert expected 401 to equal 403
  ! AssertionError
  expected 401 to equal 403
  Cypress/e2e/API/apiOrders.cy.js:10:34
   8 |     failOnStatusCode: false
   9 |   }).then((response) => {
> 10 |     expect(response.status).to.eq(403) // => accès à la ressource refusé
      |                                     ^
   11 |   })
   12 | })
   13 | })
```

Observations

Certains tests ont permis de mettre en évidence d'autres anomalies qui n'ont pas fait ici l'objet de tests particuliers mais qu'il faudra prévoir de tester dans la prochaine version car elles touchent la fonctionnalité produit, qui est également une fonctionnalité essentielle :

- Le test sur la requête des informations produit montre que certains produits affichent un stock négatif (dans l'API et sur le site web). Le test sur l'ajout d'une quantité supérieure à 20 pour un produit disponible montrent qu'une quantité supérieure au stock disponible peut être ajoutée au panier et que le stock mis à jour prend une valeur négative. Ce défaut a un impact sur l'expérience utilisateur qui ne comprendra pas l'état du stock.

- Les tests menés sur le site web pour saisir une quantité pour un produit donné montrent qu'il est possible à l'aide des flèches ^ et v de sélectionner une quantité > 20 ou < 1. Afin d'être sûr que les contraintes liées à la quantité produit ajouté au panier soient respectées, je recommande :

- de ne pas permettre à l'utilisateur de saisir manuellement la quantité dans le champs
- de limiter les valeurs possibles à sélectionner entre 1 et 20.

- Le formulaire permettant de déposer un avis a été testé dans le but de vérifier l'absence de faille de sécurité XSS. Aucune anomalie n'a été décelée, cependant j'émet une réserve car même si aucune fenêtre d'alerte ne s'affiche, le commentaire n'apparaît pas (voir captures d'écran ci dessous).

Votre avis

Note



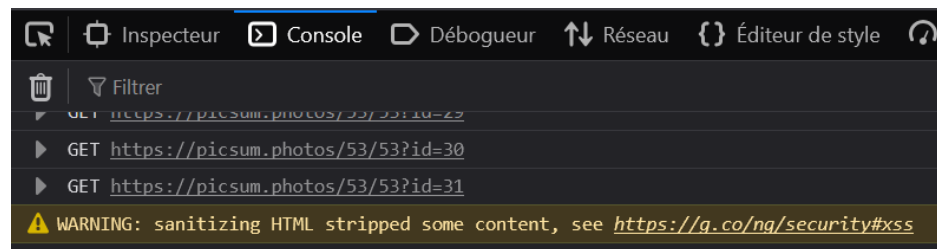
Titre

test faille XSS

Commentaire

<script>alert("XSS")</script>

Publier



Confiance

Les tests effectués montrent que cette première version déployée pour Eco Bliss Bath nécessite des améliorations importantes afin de garantir le bon fonctionnement du site.

En effet, des fonctionnalités essentielles telle que le panier sont en échec.

Les anomalies remontées pour cette fonctionnalité sont critiques.

La possibilité d'ajouter plus de 20 exemplaires d'un produit ou encore d'ajouter un produit indisponible peut entraîner un dysfonctionnement majeur dans la commande. Si la validation est acceptée, le client ne recevra pas les produits ajoutés au panier et ne sera pas satisfait. Si le processus de commande n'aboutit pas, cela représente une perte financière pour l'entreprise et engendre une perte de confiance côté client. Cela risque de porter atteinte à l'image de l'entreprise.

Suite à cette campagne de test, je recommande de corriger prioritairement les anomalies détectées pour la fonctionnalité panier et de mettre en place des tests pour la fonctionnalité produit pour corriger les anomalies mises en évidence concernant l'affichage du stock.