



Rapport de tests automatisés E2E

Module : Qualité et sureté de fonctionnement logiciel

Réalisé par :

Othman BEKHOUCHE
Wassim JAAFRI
Hamza AZOUAGH

Supervisé par :

Mr. Oussama BELLAHDAR
Mr. Ilyes MAZOUZI

Table des matières

1	Introduction générale	2
2	Présentation du projet	3
2.1	Site testé : saucedemo.com	3
2.2	Outil de test : cypress	4
2.3	Objectif	4
3	Fonctionnalités testées	5
3.1	Affichage du formulaire de connexion	5
3.2	Connexion réussie avec un utilisateur valide	5
3.3	Validation des champs obligatoires	5
3.4	Gestion du cas d'un utilisateur bloqué	5
4	Résultats	6
4.1	Bug 1 – Erreurs réseau (POST 401) pendant la saisie des champs	6
4.2	Bug 2 – Erreur affichée mais validation UI incomplète	7
4.3	Bug 3 – Utilisateur bloqué : message peu explicite	8
4.4	Scénarios de connexion Swag Labs - Page login	9
5	Recommandations et conclusion	10
5.1	Recommandations	10
5.2	Conclusion	10

Chapitre 1

Introduction générale

Dans un écosystème de développement web en constante évolution, la fiabilité des applications est devenue un enjeu majeur pour garantir la satisfaction des utilisateurs et la pérennité des produits. L'Assurance Qualité (QA), et plus spécifiquement l'automatisation des tests, joue un rôle central pour sécuriser les mises en production et réduire le temps de cycle de développement.

Ce projet s'inscrit dans cette démarche d'ingénierie de la qualité. Il a pour objectif la mise en œuvre d'une suite de tests automatisés de bout en bout (End-to-End) sur l'application SauceDemo (Swag Labs), une plateforme e-commerce de référence utilisée pour l'entraînement aux pratiques de testing.

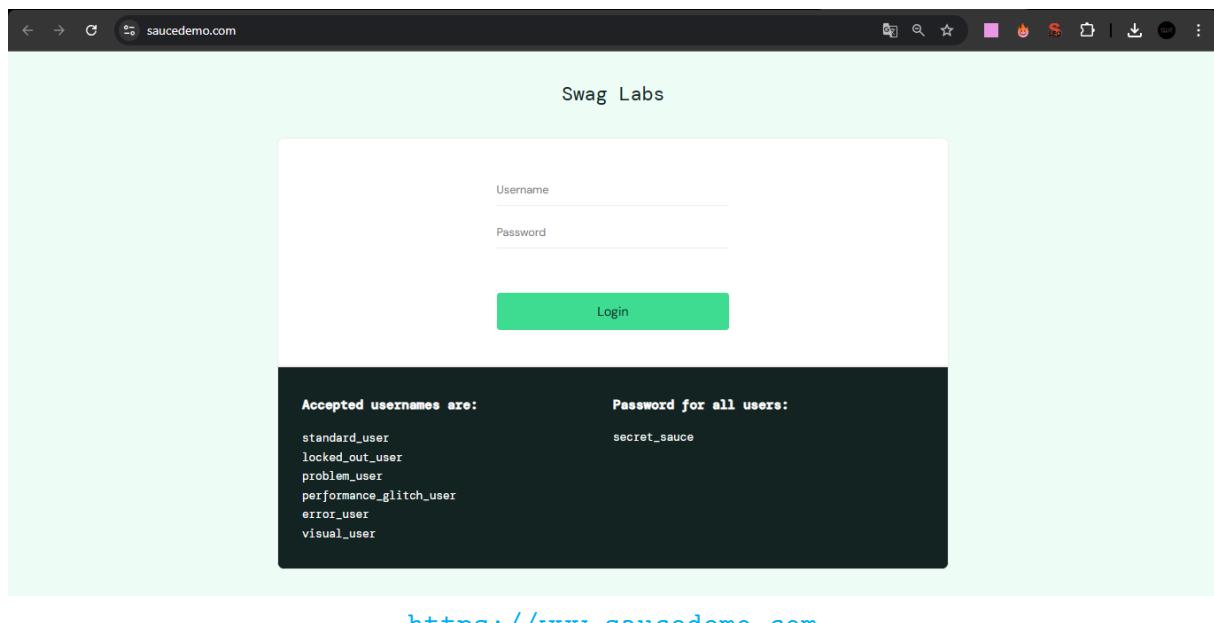
Le choix technologique s'est porté sur le framework Cypress. Contrairement aux solutions basées sur Selenium, Cypress s'exécute directement dans le navigateur, offrant ainsi une vitesse d'exécution accrue, une meilleure stabilité et des capacités de débogage avancées.

Le périmètre de ce rapport se concentre sur le module critique de l'authentification. Nous avons cherché à valider non seulement les scénarios nominaux (connexion réussie), mais également la robustesse de l'application face aux erreurs (champs vides, utilisateurs bloqués) et aux comportements imprévus (erreurs réseaux, incohérences d'interface).

Chapitre 2

Présentation du projet

2.1 Site testé : saucedemo.com



Saucedemo.com est un faux site e-commerce créé spécifiquement pour les développeurs et les testeurs logiciels. Il n'a aucune vocation commerciale réelle : vous ne pouvez rien y acheter. Son unique but est de servir de terrain d'entraînement pour tester des programmes informatiques. Il permet de s'exercer à l'automatisation de tests. Quand une entreprise crée un site web, elle doit vérifier qu'il fonctionne (que le bouton "panier" marche, que la connexion est sécurisée, etc.). Au lieu de faire ces clics à la main, on écrit des scripts (des robots) pour le faire.

Il est stable : Contrairement à un vrai site qui change souvent (nouvelles pubs, mises à jour), celui-ci reste identique, ce qui évite de casser les scripts d'entraînement.

Il simule des bugs : Il fournit des identifiants exprès pour voir votre robot réagit face à une erreur (exemple : un utilisateur bloqué ou un glitch d'affichage).

2.2 Outil de test : cypress



Si Saucedemo est la cible d'entraînement, alors Cypress est l'arme. **Cypress** est un outil open source JavaScript moderne pour les tests automatisés, se concentrant principalement sur les tests d'« End-to-End » (E2E) et de composants, permettant aux développeurs de simuler des actions utilisateur complètes dans un navigateur pour vérifier le fonctionnement d'une application web du début à la fin, avec un débogage visuel et des snapshots à chaque étape pour faciliter l'identification des problèmes. C'est une solution complète pour les développeurs souhaitant automatiser les tests de leur application web, en offrant un environnement rapide et convivial pour écrire, exécuter et déboguer des tests fonctionnels et de composants, garantissant ainsi la qualité de l'expérience utilisateur et des fonctionnalités.

2.3 Objectif

L'objectif principal de cypress est de simplifier et fiabiliser la mise en place de tests pour les applications web.

Contrairement à d'autres outils (comme Selenium) qui pilotent le navigateur de l'extérieur via le réseau, Cypress s'exécute directement à l'intérieur du navigateur, dans la même boucle d'exécution que votre application.

Notre objectif est de vérifier les fonctionnalités principales de la page de connexion du site SauceDemo à l'aide de tests automatisés E2E (End-to-End).

Chapitre 3

Fonctionnalités testées

Dans le cadre de ce projet, des tests automatisés ont été conçus et exécutés avec succès à l'aide de l'outil Cypress sur la plateforme de démonstration SauceDemo. Ces tests ont été utilisés pour valider les fonctionnalités d'authentification, avec pour objectif principal de vérifier la fiabilité et la conformité du processus de connexion proposé par l'application.

3.1 Affichage du formulaire de connexion

Ce scénario consiste à vérifier que la page de connexion de SauceDemo s'affiche correctement. À l'aide de Cypress, la présence des champs Username et Password, ainsi que du bouton Login, est contrôlée afin de s'assurer que l'interface utilisateur est accessible et fonctionnelle.

3.2 Connexion réussie avec un utilisateur valide

Ce test vise à valider le bon fonctionnement de l'authentification avec un utilisateur autorisé. Cypress simule la saisie des identifiants du compte standarduser et déclenche l'action de connexion. Le succès du test est confirmé par la redirection vers la page des produits, indiquant que l'utilisateur a été correctement authentifié.

3.3 Validation des champs obligatoires

Ce scénario permet de vérifier le comportement de l'application lorsque les champs de connexion sont laissés vides. Cypress tente de soumettre le formulaire sans renseigner les identifiants et vérifie l'affichage d'un message d'erreur informant l'utilisateur que les champs requis doivent être remplis.

3.4 Gestion du cas d'un utilisateur bloqué

Ce test valide la gestion des comptes verrouillés sur SauceDemo. Cypress simule une tentative de connexion avec les identifiants du compte lockedoutuser. Le test vérifie que l'accès est refusé et qu'un message d'erreur explicite est affiché, indiquant que le compte est bloqué.

Chapitre 4

Résultats

4.1 Bug 1 – Erreurs réseau (POST 401) pendant la saisie des champs

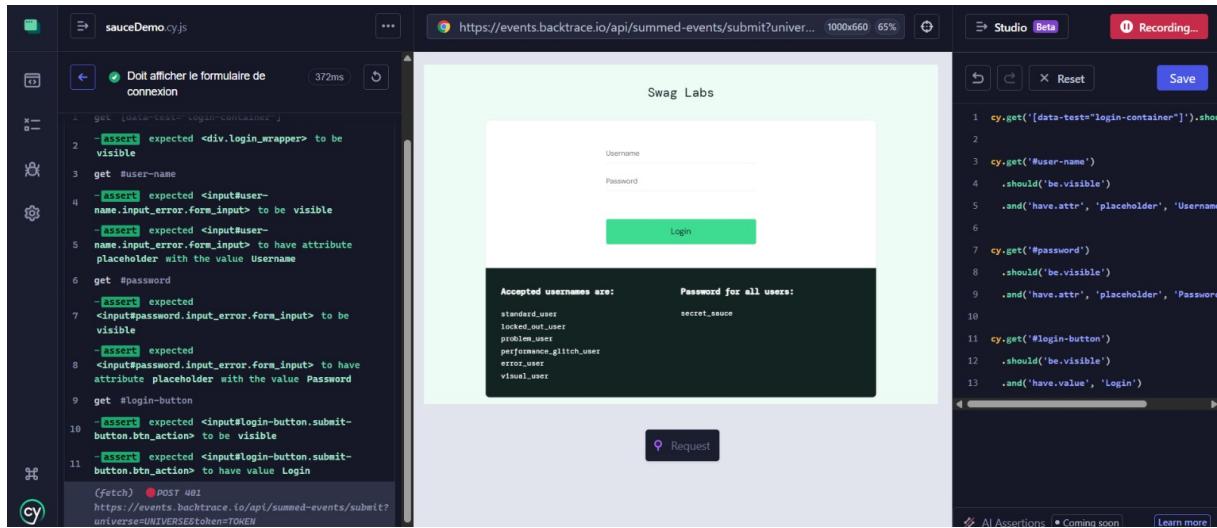


FIGURE 4.1 – Erreur 401 visible dans les logs Cypress

Description : Lors de la saisie du *username* et du *password*, des requêtes réseau automatiques sont envoyées vers l'API [https://events.backtrace.io/api/....](https://events.backtrace.io/api/). Ces requêtes retournent une erreur HTTP 401 (Unauthorized).

Comportement observé :

- L'erreur apparaît dans le log Cypress.
- Elle se produit même lorsque l'utilisateur saisit des informations correctes.
- Le test fonctionnel continue et passe correctement.

Comportement attendu :

- Aucune erreur réseau ne devrait apparaître lors d'une simple saisie utilisateur.
- Les outils de tracking ne devraient pas impacter les tests.

Impact : Peut créer de la confusion lors de l'analyse des résultats et engendre un risque de faux positifs lors des tests automatisés.

Cause probable : Service externe de monitoring (Backtrace) mal configuré ou token d'authentification manquant/invalide.

4.2 Bug 2 – Erreur affichée mais validation UI incomplète

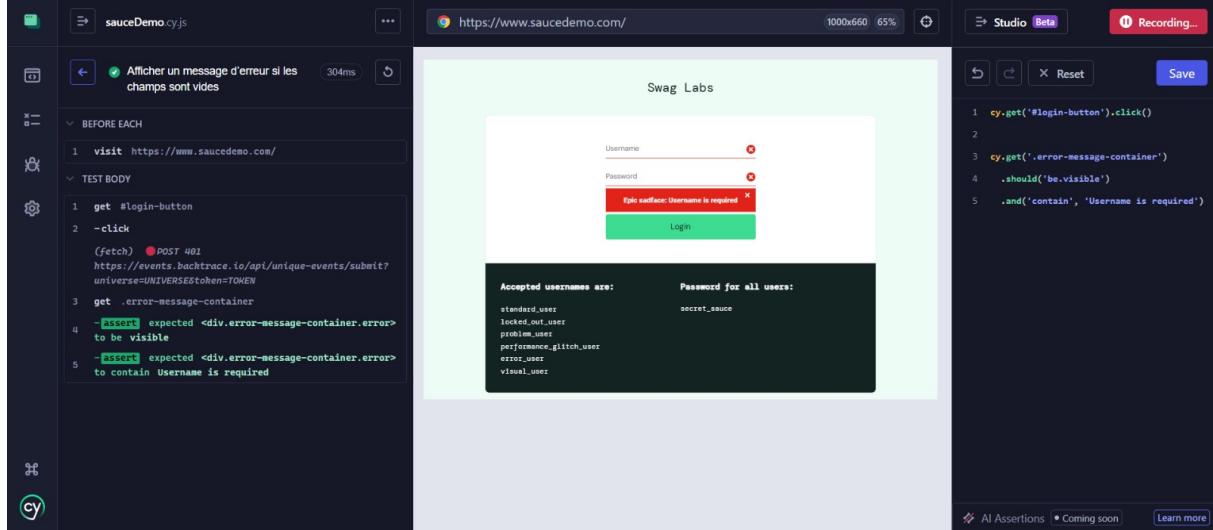


FIGURE 4.2 – Message d’erreur sans focus automatique

Description : Lorsque l’utilisateur clique sur *Login* sans remplir les champs, un message d’erreur s’affiche, mais la validation visuelle n’est pas totalement cohérente.

Comportement observé :

- Message affiché : "Epic sadface : Username is required".
- Icônes d’erreur visibles.
- **Pas de focus automatique** sur le champ concerné.

Comportement attendu :

- Focus automatique sur le champ *Username*.
- Message d’erreur plus précis si plusieurs champs sont vides.

Impact : Expérience utilisateur perfectible, peut gêner les utilisateurs novices.

4.3 Bug 3 – Utilisateur bloqué : message peu explicite

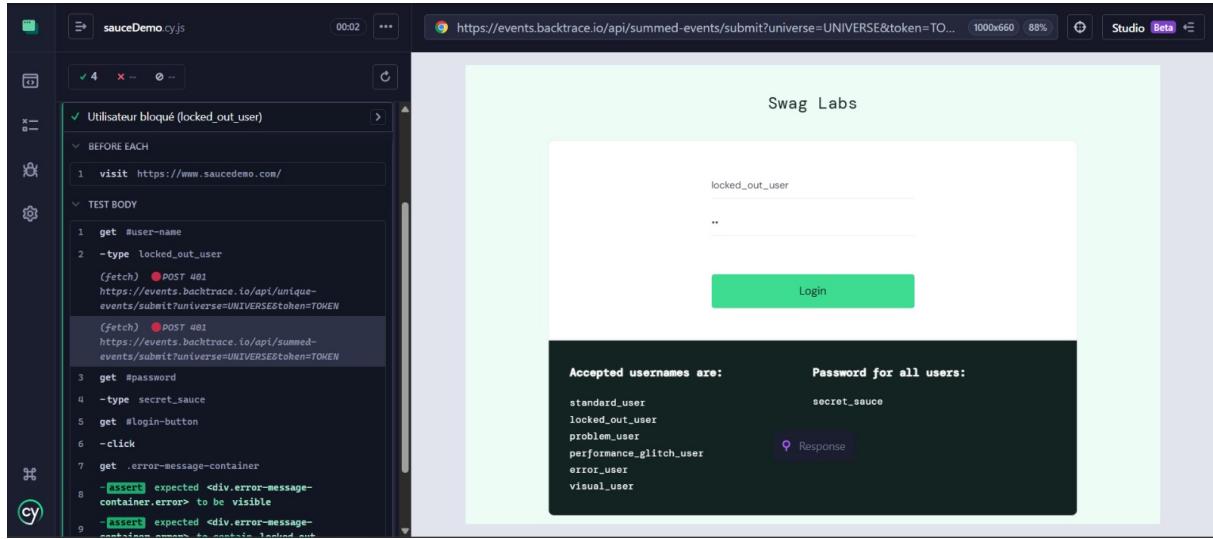


FIGURE 4.3 – Message d’erreur pour utilisateur bloqué

Description : Lors de la connexion avec l’utilisateur *locked_out_user*, le système bloque correctement l’accès.

Comportement observé :

- Message d’erreur affiché.
- L’utilisateur n’est pas redirigé.
- Aucun lien ou action proposée.

Comportement attendu :

- Message plus explicite (ex : contacter l’administrateur).
- Option d’assistance ou de support.

Impact : Bonne sécurité, mais mauvaise communication utilisateur.

4.4 Scénarios de connexion Swag Labs - Page login

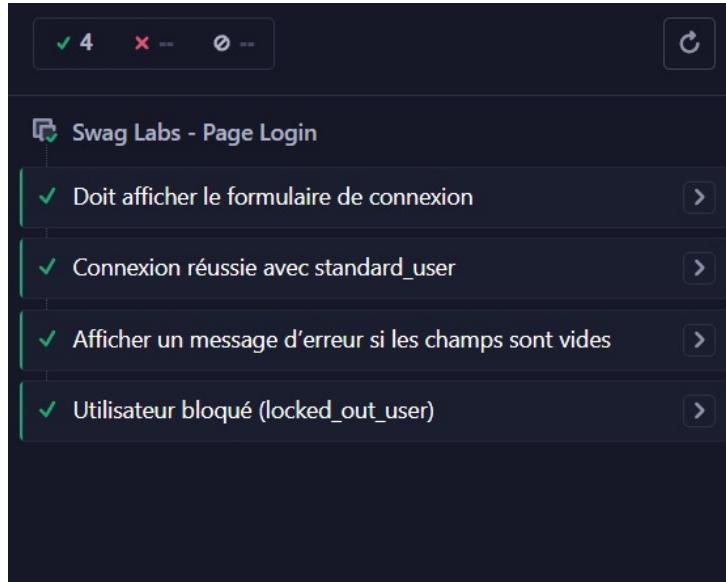


FIGURE 4.4 – Swag Labs - Page login

Description : Tous les indicateurs sont au vert pour la fonctionnalité de connexion. Les tests confirment le bon comportement de l'application face aux scénarios d'accès valides et invalides, assurant ainsi la stabilité du mécanisme d'authentification avant déploiement.

La figure illustre la réussite des 4 suites de tests sur le composant Login :

1. **UI** : Présence du formulaire.
2. **Cas passant** : Connexion avec standardUser.
3. **Gestion d'erreurs** : Validation des champs obligatoires.
4. **Sécurité/Logique métier** : Gestion correcte du statut lockedOutUser.

Scénario	Résultat
Affichage du formulaire	Succès
Connexion <i>standard_user</i>	Succès
Champs vides	Succès
Utilisateur bloqué	Succès
Erreurs réseau parasites	À corriger

TABLE 4.1 – Tableau récapitulatif des exécutions

Chapitre 5

Recommandations et conclusion

5.1 Recommandations

Dans le cadre de l'amélioration continue de la qualité de l'application, plusieurs recommandations peuvent être formulées. Sur le plan technique, il est conseillé de désactiver ou de simuler (mocker) les services de suivi et de reporting des erreurs, tels que Backtrace, en environnement de test afin d'éviter la génération de faux positifs pouvant fausser les résultats des tests automatisés. Du point de vue de l'expérience utilisateur (UX), l'amélioration des messages d'erreur ainsi que la gestion automatique du focus sur les champs obligatoires contribuerait à une meilleure compréhension des erreurs par l'utilisateur et à une navigation plus intuitive. En termes de qualité logicielle, l'ajout d'assertions Cypress spécifiques permettant de vérifier l'absence d'erreurs réseau critiques renforcerait la fiabilité des scénarios de test. Enfin, sur le plan architectural, la centralisation de la gestion des erreurs destinées aux utilisateurs favoriserait une meilleure cohérence globale de l'application, tout en facilitant la maintenance et l'évolution du système.

5.2 Conclusion

Ce travail a permis de mettre en évidence l'importance des tests automatisés End-to-End dans le cadre de l'assurance qualité des applications web modernes. À travers l'utilisation de l'outil Cypress sur la plateforme de démonstration SauceDemo, une suite de scénarios de test ciblant le processus d'authentification a été conçue, automatisée et exécutée avec succès.

Les résultats obtenus confirment la fiabilité fonctionnelle du mécanisme de connexion, aussi bien pour les scénarios nominaux que pour les cas d'erreur courants, tels que la saisie de champs vides ou la gestion des comptes utilisateurs bloqués. Par ailleurs, l'analyse des logs et du comportement applicatif a permis d'identifier certaines anomalies techniques et ergonomiques, notamment liées aux services externes de monitoring et à l'expérience utilisateur, démontrant ainsi la valeur ajoutée des tests automatisés dans la détection précoce des défauts. Ce projet illustre également les avantages de Cypress en matière de rapidité d'exécution, de lisibilité des scénarios et de facilité de débogage, en faisant un outil particulièrement adapté aux environnements de développement modernes.

En conclusion, l'automatisation des tests E2E constitue un levier essentiel pour améliorer la qualité logicielle, réduire les risques de régression et assurer une expérience utilisateur fiable et cohérente.