Bilan de campagne de test pour 724Events

Sommaire

Introduction

Ce document présente le bilan des tests effectués pour l'application 724Events.

Nous y aborderons les tests fonctionnels décrits dans le cahier de recette ainsi que les tests exploratoires.

Pour ces derniers, nous avons testé avec plusieurs outils différents bien connus du monde du testing. Nous les aborderons au moment opportun.

Puis, nous comparerons les résultats obtenus avec ceux attendus avant de finir sur le mot de la fin, le Go ou No Go.

Présentation des objectifs

L'objectif de cette campagne de test est d'évaluer la qualité de la V1 de l'application 724Events.

Pour rentrer dans les détails, cela nous donne :

- Valider le cahier de recette
- Exécuter des tests exploratoires divers
- Valider le fonctionnement global de l'application dans sa globalité (frontend et backend à la fois)
- Valider les fonctionnalités principales qui sont la présentation des événements et l'inscription à un événement
- Valider les fonctionnalités secondaires comme par exemple, la prise de contact par un utilisateur via un formulaire, la pagination des événements etc...
- Valider le fonctionnement de l'application sur un mobile (responsive).
- Valider le fonctionnement de l'API en isolée
- Vérifier l'absence de failles de sécurité critiques
- Déterminer les anomalies et les répertorier avec leurs origines afin de faciliter leurs résolutions si possibles
- Rapport des observations

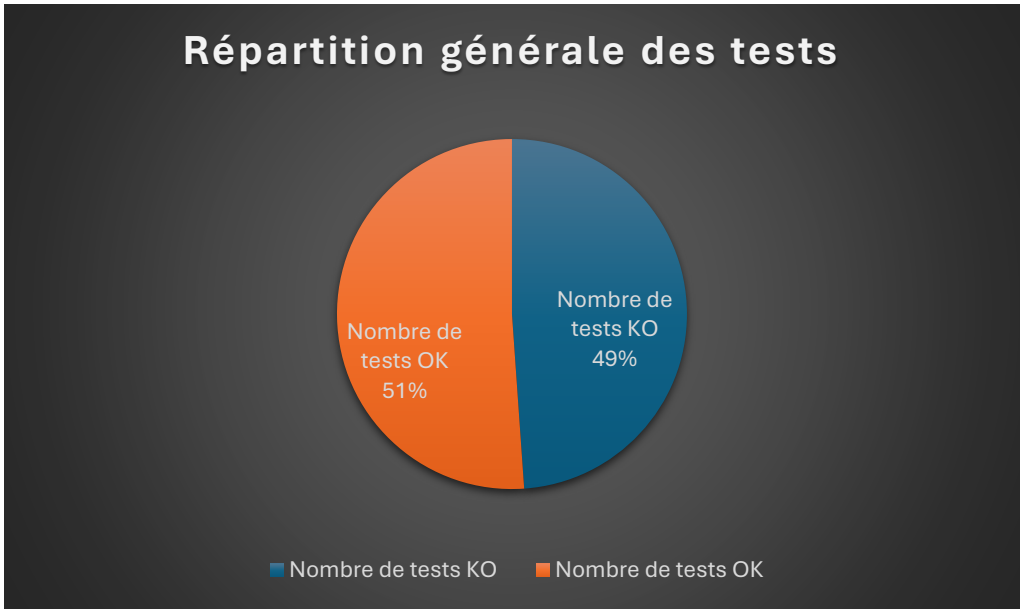
Outils utilisés

Le tableau ci-dessous récapitule les outils utilisés.

Outil	Utilisation
Navigateur internet	Utilisé pour la navigation, la visualisation des résultats mais aussi pour la recherche d'informations non visibles par l'utilisateur lambda.
Postman	Utilisé pour tester l'API de manière plus poussée
Zaproxy	Utilisé pour tester la sécurité globale de l'api
Cypress	Utilisé pour automatiser les tests
Swagger	Utilisé en premier lieu pour tester simplement l'API
SonarQube	Utilisé pour tester la qualité globale du code
Jira	Utilisé pour décrire et répertorier les anomalies rencontrées.

Présentations des résultats

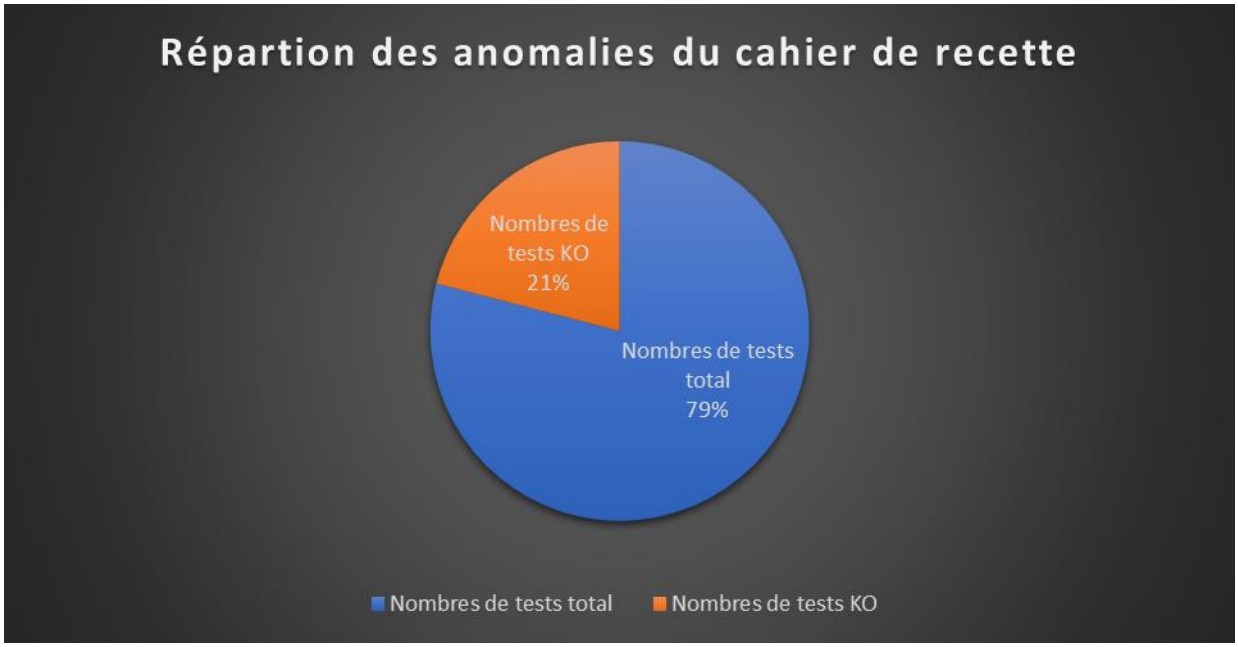
Nos tests ont révélé un total de 44 anomalies sur un total de 90 tests



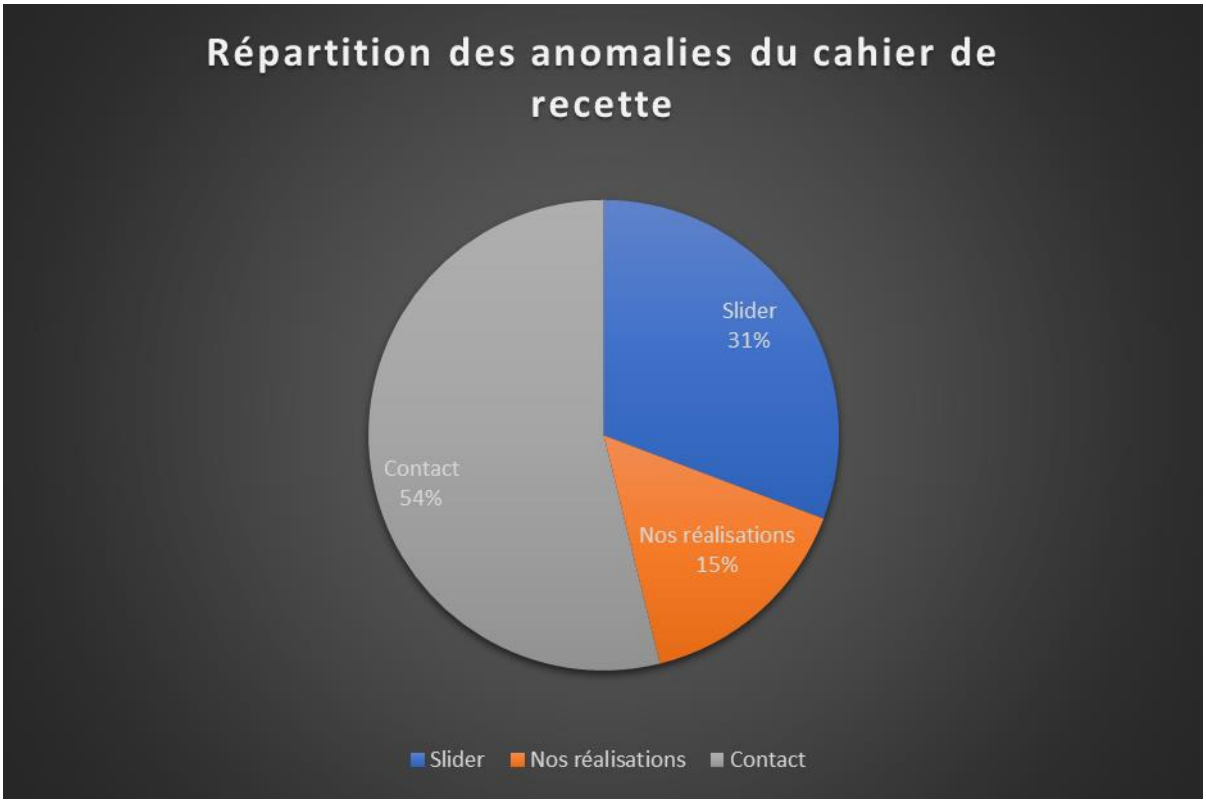
Voyons maintenant les détails.

Cahier de recettes

Le cahier de recette présente un total de 59 tests réparties sur plusieurs éléments du site. Nous avons décelé 13 anomalies ce qui représente un total de 21%.

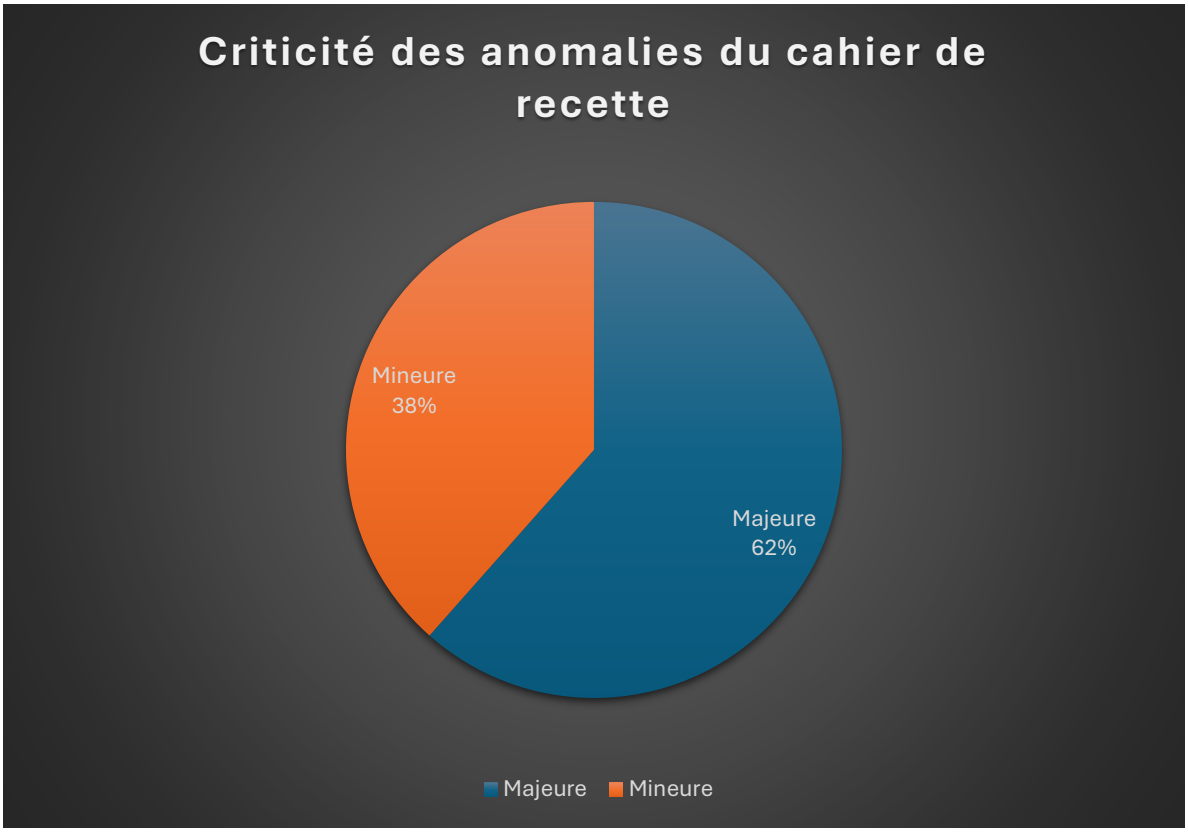


Ces anomalies sont réparties de la façon suivante :



Nous pouvons voir clairement que le formulaire ainsi que le slider posent problèmes.

Enfin, concernant la répartition de la criticité, nous voyons clairement une majorité de criticité majeure et mineure :

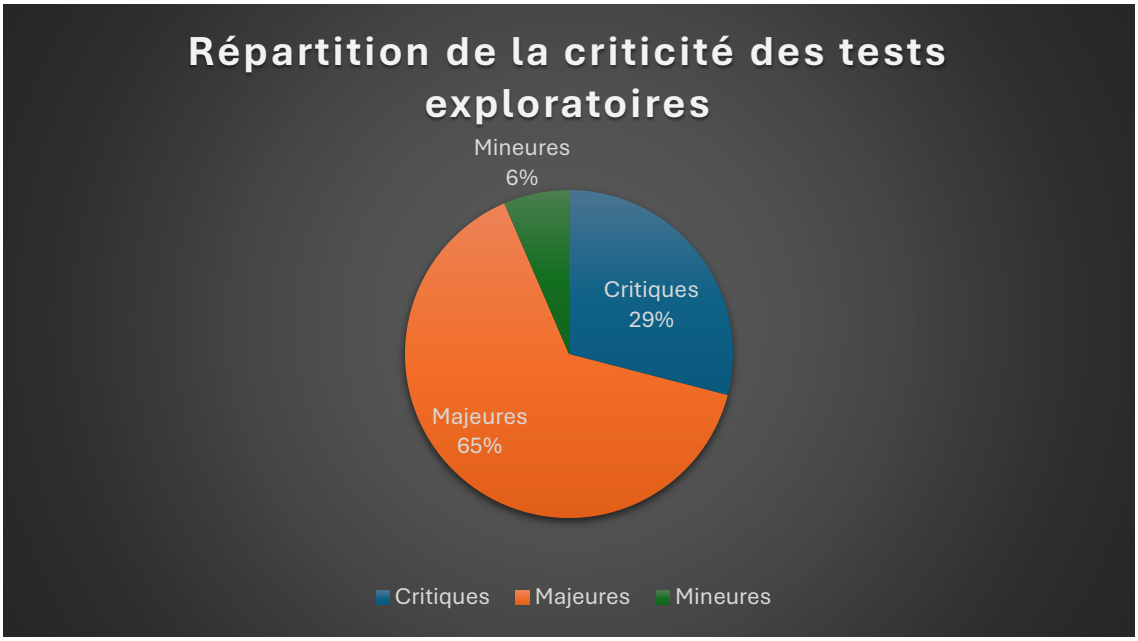


Ces anomalies majeures concernent presque entièrement le formulaire de contact.

Tests exploratoires

Les tests exploratoires ont porté leurs fruits et ont révélé un total de 31 anomalies. En effet, nous avons testé l'API, navigué sur l'application, testé la sécurité, testé la qualité générale du code. Nous en reparlerons dans les observations ci-dessous.

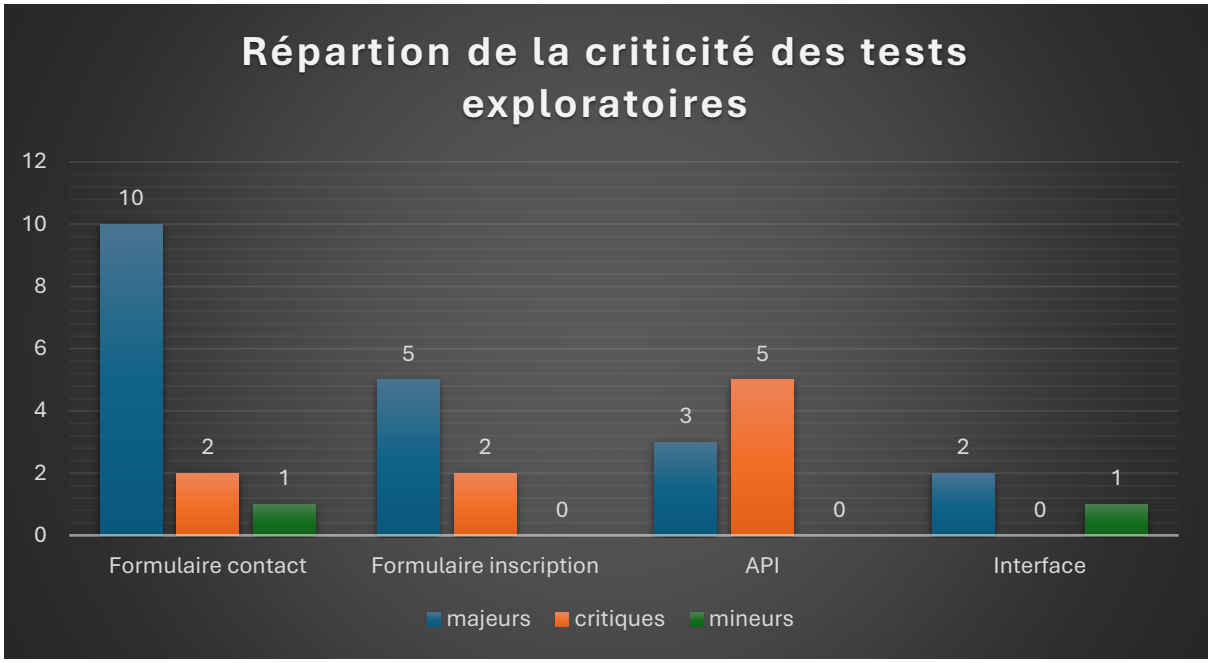
Le graphique ci-dessous nous montre clairement une dominance des anomalies majeures et critiques :



Certaines anomalies critiques ont même fait planter l'application et cela a nécessité un redémarrage à la main.

Nous avons été le plus précis possible ce qui explique ce grand nombre d'anomalies.

Nous avons principalement testé 4 catégories. Une fois de plus, ce graphique montre que le formulaire de contact n'est pas au point :

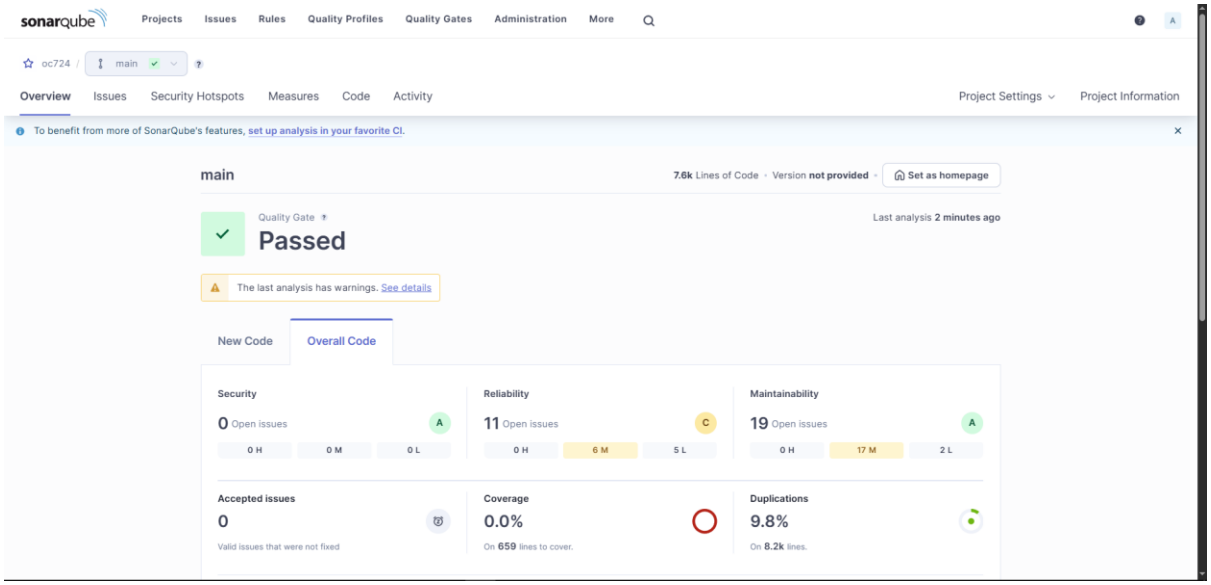


Suivi de l'API qui présente des anomalies critiques et majeures.

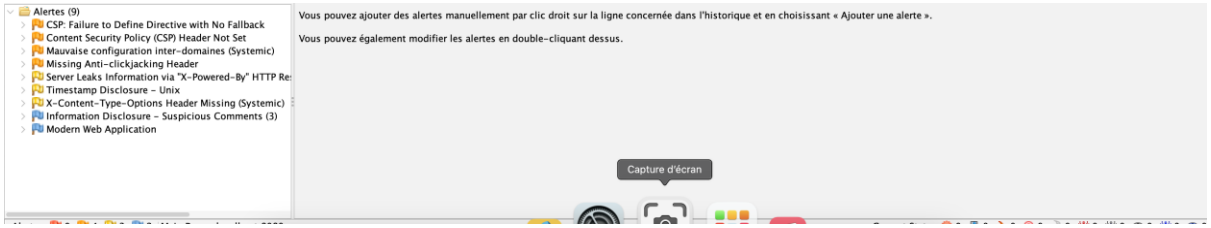
Observations

Lors des tests exploratoires et du cahier de recette, nous avons pu faire quelques observations :

- "Oeuvre" est mal orthographiée dans la page de la description du WEF.
    - La correction est "œuvre"
- Favicon
    - La favicon est celle du framework React et non celle de l'application
- Responsive
    - Le site "commence à tirer la tête" dès que la largeur descend en dessous de 436px.
- Image section
    - L'image de la section "Slider" est trop large, elle déborde ce qui "casse" le layout.
- Qualité générale du code
    - La qualité générale du code est "Passed" comme le montre la capture ci-dessous



- Sécurité
    - L’application ne présente pas de failles de sécurité critique comme le montre la capture ci-dessous.



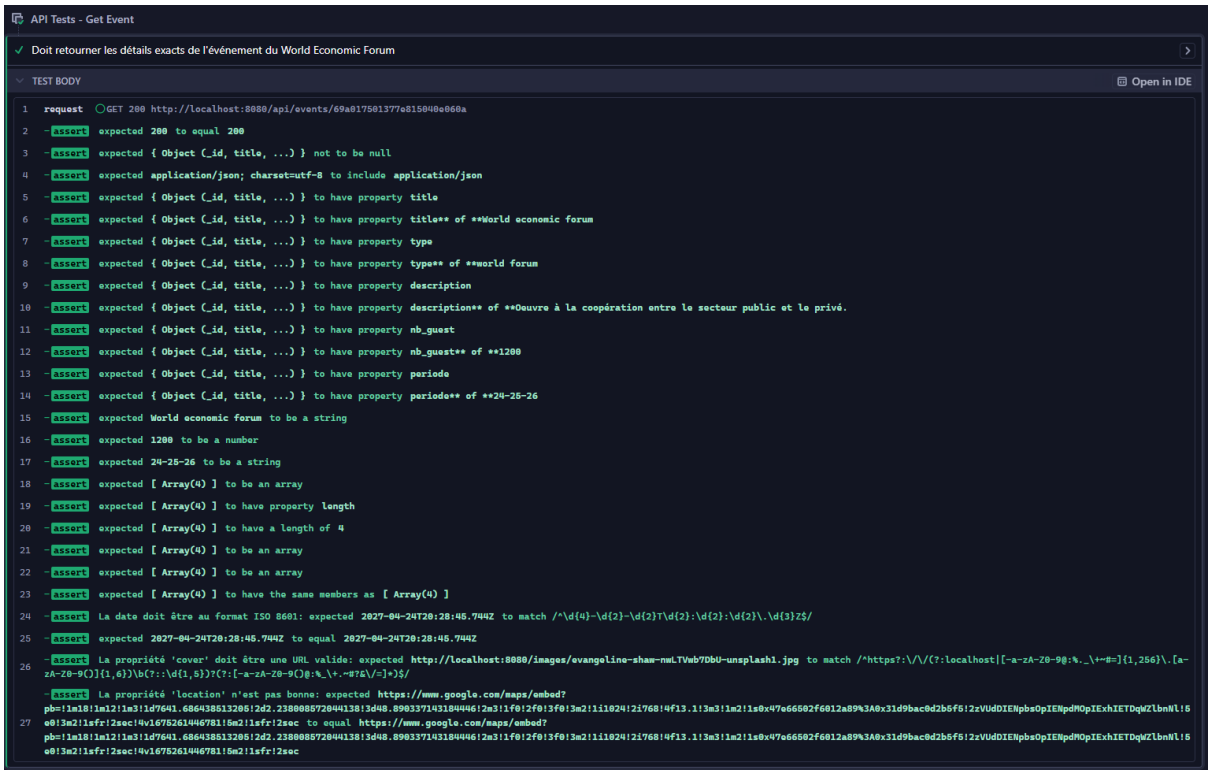
Tests automatisés

Nous avons choisi d'automatiser le formulaire d'inscription et la récupération d'un événement car ces derniers sont la pierre angulaire de l'application. L'utilisateur intéressé par ce que propose l'application doit pouvoir accéder aux informations des événements et s'y inscrire. Il est vrai que le formulaire d'inscription est plus intéressant côté utilisateur que celui du contact, cependant le nombre d'anomalies rencontrées dans ce dernier nous montrent clairement qu'il est à tester de manière automatique.

En effet, il est assez complexe mais avec la mise en place de beaucoup de contraintes pour « contrer les personnes malveillantes » mais il peut devenir très « sûr » et limiter fortement les bugs futurs. De plus, avec Cypress, cela peut devenir rapide et réutilisable pour les 2 formulaires.

Récupération d'un événement (World Economic Forum)

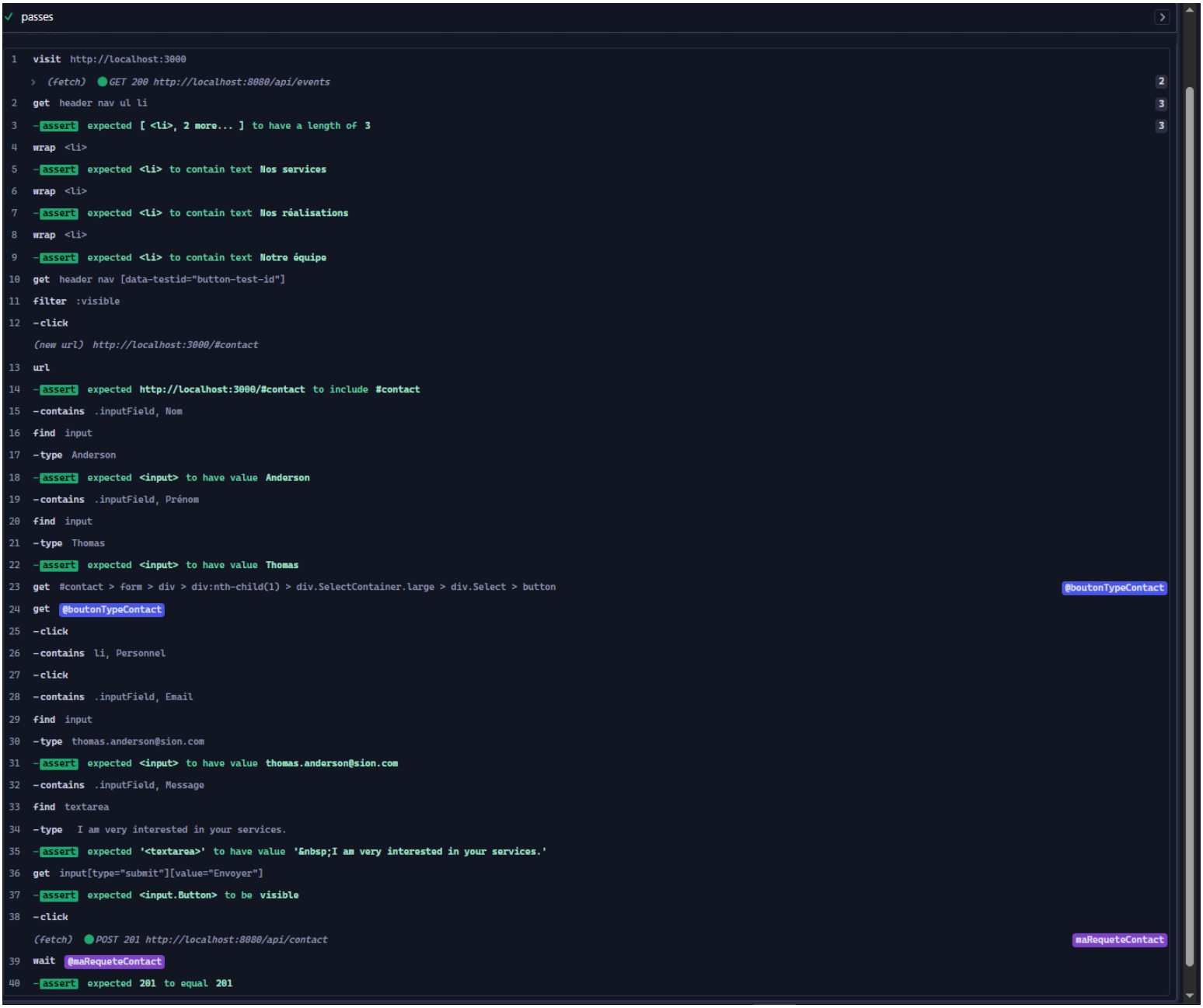
La récupération de l'événement du World Economic Forum en elle-même est un succès :



Cependant, nous avons vu dans les tests exploratoires que les dates remontées ne sont pas toujours les bonnes. Une recherche plus approfondie par les développeurs sera nécessaire pour déterminer la racine de l'anomalie

Formulaire de contact

Le formulaire en lui-même (hors message de succès) fonctionne mais il laisse « tout passer » à l'instant T. Ainsi, avec des données valides, nous voyons le succès du contact par un utilisateur :



Nous voyons clairement le code de succès 201 renvoyé par l'API. Nous voyons aussi dans la capture d'écran qu'aucun filtre n'a été appliqué… Appliquer les filtres à ce test pourra comme vu précédemment réduire grandement les bugs et anomalies potentiels futurs…

Recommandations

Afin de faciliter les tests futurs, ne pas hésiter à privilégier les id uniques dans le code HTML pour les objets manipulables et remplissable par un utilisateur. En effet, la sélection de l'élément est plus simple, plus rapide mais surtout plus robuste car il faudra changer chaque test ou l'élément est compris afin d'être à jour si un développeur change une partie du fonctionnement des éléments destinés à l'utilisateur.

Concernant les inputs des formulaires, privilégier les outils déjà en place des frameworks frontend afin de déterminer la validité ou au minimum utiliser une expression régulière.

Ne pas hésiter à mettre des dispositions de sécurité dans le frontend et le backend. Il ne faut pas avoir confiance dans ce que renvoie le frontend car ce dernier peut être simulé par des outils voire mêmes des requêtes peuvent être interceptées et modifiées à des fins malicieuses.

L'ANSSI fournit un guide des recommandations de sécurités.

Conclusion

L'application en l'état n'est pas déployable à cause des anomalies majeures et critiques rencontrées.

C'est un "No Go" pour la V1.

Le "Go" est atteignable si les corrections se focalisent sur les parties critiques notamment concernant la sécurité mais aussi sur le formulaire de contact...

Annexes

Tableau de tous les tickets Jira

44 anomalies

Numéro de ticket	Titre
CCS-9	Slider / Défilement Images
CCS-10	Slider / Conférence / Mois
CCS-12	Nos réalisations / Pagination
CCS-13	Nos réalisations / Filtre
CCS-14	Tests exploratoires / Test de charge
CCS-19	Formulaire contact / Caractères non latins
CCS-20	Formulaire contact / Grand nombre de caractères
CCS-21	Formulaire contact / 1 seul caractère
CCS-22	Formulaire contact / Email caractères non latin
CCS-23	Formulaire contact / Plusieurs adresses mails valides
CCS-24	Formulaire contact / Caractères spéciaux dans adresse mail
CCS-25	Formulaire contact / Tenter l'injection SQL
CCS-27	Formulaire contact / Champs vides
CCS-28	Formulaire contact / XSS
CCS-29	Formulaire contact / Trim
CCS-30	Formulaire contact / Rage Click
CCS-31	Formulaire contact / Espaces only
CCS-32	Formulaire inscription / Date antérieure événement
CCS-33	Formulaire inscription / Date future événement
CCS-34	Formulaire inscription / Caractères non latins
CCS-35	Formulaire inscription / Très grand nombre de caractères
CCS-36	Formulaire inscription / XSS
CCS-37	Formulaire inscription / Injection SQL
CCS-38	Formulaire inscription / Adresse mail mal formatée
CCS-39	Postman / ID event trop court
CCS-40	Postman / ID Mal formaté mais de bonne longueur
CCS-41	Postman / ID incorrect
CCS-42	Postman / Récupérer dernier événement
CCS-43	Postman / POST Formulaire contact avec mail mal formaté
CCS-44	Postman / POST Injection XSS
CCS-45	Postman / POST Inscription Happy Path
CCS-46	Formulaire contact / Happy path valide
CCS-48	Formulaire contact / Nom only
CCS-49	Formulaire contact / Prénom only
CCS-50	Formulairre contact / Message only
CCS-51	Formulaire contact / Autorisation manquante
CCS-52	Fomulaire contact / Email only
CCS-53	Formulaire contact / Personnel ou Entreprise only
CCS-54	World Economic Forum / Mois Mai
CCS-55	Formulaire Inscription / Happy Path
CCS-57	Slider / Bouton sous image
CCS-58	Fomulaire Contact / Mail Capitalized
CCS-59	Événements / map Google
CCS-60	Événements / Unicité des dates