# Projet de Tests avec Robot Framework

## 1. Introduction

Ce projet a pour objectif de tester l’API et la page web du site StackOverflow à l’aide de Robot Framework et de bibliothèques comme Playwright. L’idée est de valider la cohérence entre les données renvoyées par l’API et celles affichées sur l’interface web.

## 2. Cahier des charges

Le test technique repose sur deux axes principaux :

- \*\*API\*\* : Vérifier que l’API `https://api.stackexchange.com/2.3/questions` permet de récupérer correctement les questions et proposer différents scénarios de test.

- \*\*Page Web\*\* : Vérifier que la première question affichée sur la page `https://stackoverflow.com/questions?tab=Newest` correspond à celle renvoyée par l’API.

## 3. Organisation du projet

Le projet est structuré comme suit :

- \*\*api/\*\* : Contient les tests spécifiques à l’API  
- \*\*pages/\*\* : Contient les pages objets utilisées pour les tests Web  
- \*\*resources/\*\* : Variables, configuration et fichiers utilitaires  
- \*\*tests/\*\* : Scripts de test Robot Framework (API, Web, multi-utilisateurs, performance)  
- \*\*test\_data/\*\* : Données externes utilisées pour les tests  
- \*\*results/\*\* : Rapports générés après exécution des tests (Allure, HTML, XML)  
- \*\*requirements.txt\*\* : Liste des dépendances Python  
- \*\*setup.bat / run\_tests.bat\*\* : Scripts pour l’installation et l’exécution rapide

## 4. Scénarios de test proposés

Quelques idées de scénarios automatisés :

- Vérifier que l’API renvoie bien une réponse avec code 200  
- Vérifier que le format des données est bien en JSON  
- Vérifier la présence des champs attendus (ex. `title`, `link`, `owner`)  
- Vérifier que la liste renvoyée n’est pas vide  
- Comparer le titre de la première question API avec celui de la page web  
- Exécuter un test multi-utilisateurs simulant plusieurs appels simultanés  
- Vérifier les performances de l’API (temps de réponse)

## 5. Exécution des tests

Pour exécuter les tests, il suffit de suivre les étapes suivantes :

1. Lancer les tests via Robot Framework :  
 run\_tests.bat

2. Consulter les rapports générés dans le dossier `results/`.

**Points d'amélioration**

* Ajouter davantage de cas de test couvrant des scénarios négatifs (ex. API hors service, paramètres invalides).
* Mettre en place des tests de sécurité pour vérifier que les données sensibles ne sont pas exposées.
* Automatiser l’intégration des tests dans un pipeline CI/CD (ex. GitHub Actions, GitLab CI).
* Ajouter un monitoring continu de l’API avec alertes en cas de lenteur ou d’erreurs.
* Étendre les tests de performance avec des charges plus importantes.
* Générer automatiquement des jeux de données pour enrichir les tests.
* Améliorer la documentation technique avec des schémas d’architecture .

## 6. Conclusion

Ce projet fournit une base pour tester à la fois une API et une interface web. Il démontre l’utilisation de Robot Framework pour des tests fonctionnels, multi-utilisateurs et de performance.