**Stratégie de Test pour l’API « Porduct »**

Afin de tester l’API « product », je vous propose 3 niveau différentes :

**Tests unitaire (Niveau développeur) :**

Création des tests unitaires pour l’ensemble des scénarios possibles de chaque méthode de l’API (GET, POST, PATCH …)

Afin de vérifier le bon fonctionnement de l’API techniquement et fonctionnellement, il faut créer un maximum de scénarios afin de traiter tous les cas possibles.

Il est important de valider toutes les méthodes de façon unitaire, en utilisant des jeux des données moquées fournis par le métier ou les testeurs.

**Tests d’acceptation (Niveau Testeur) :**

Sur un environnement de recette, un QA peut vérifier le bon fonctionnement de l’API, et si elle est conforme aux spécifications (scénarios métiers, règle de gestion, caractère obligatoire d’un champ…),

**Tests de performance :**

C’est tests ont pour objectif de mesurer les temps de réponses d’une ressource d’API en fonction de sa sollicitation. Pour que le fonctionnement du cache soit bien testé, les jeux de données doivent être en nombre suffisant et représentatifs.

**Types de Tests :**

* **Tests de syntaxes** : format, orthographe, caractère obligatoire ou non d’un champ…
* **Tests fonctionnels** : règle de gestion, calcul…
* Construction et exécution de **scénarios de tests**, cas de tests métiers.

Pour des tests d’API, il faut également mettre en place des scénarios complets de tests. Ces scénarios doivent correspondre à des cas métiers.

Une stratégie de test nécessite que tous les acteurs du projet (développeur, testeur, chef de projet …) soit inclus dans la création de cette stratégie.

Les types de tests ne sont pas les mêmes selon l’environnements de test et selon celui qui conçoit et exécute les tests