Introduction

**Description** : Le but de ce plan de test est de tester la route user du backend de l’application. La création d’un utilisateur, supprimer un utilisateur, modifier les valeurs de l’utilisateur, obtenir l’utilisateur, se connecter avec l’utilisateur (non fonctionnel dans la version 1 car la route authentification le fait déjà, utile pour tester la couverture de test).

Des tests unitaires sont effectués pour tester les méthodes « … » et des tests end to end sont effectués pour tester l’interaction entre le frontend et le backend.

**Version du logiciel** : 1.0.0

**Domaine fonctionnel** : Saisies

**Contexte du plan de test** : Ce plan de test permet de vérifier le bon fonctionnement des saisies pour créer un utilisateur et pour le modifier. Il test aussi la suppression de l’utilisateur et la récupération de ses informations.

Scope

**Sous domaines testé** : backend

**Features testé** : Créer un utilisateur, Modifier l’utilisateur, Supprimer l’utilisateur, Récupérer les utilisateurs, Se connecter avec un utilisateur

Hors scope

**Cas limite non testé** :

1. UI/DOM, pas de test de design
2. Pas de test avec une base de données sur un serveur (production ou préproduction) mais avec Mongo Memory Server.

**Cas non-accessible** :

1. …

**Limites du périmètre de test** :

1. Tester si le token d’authentification est invalide, non existant ou valide.
2. Tester si le body est valide, manquant ou invalide.
3. Lors de la création d’un utilisateur, paramètre de body incorrect :
   1. Email valide, invalide, en doublon
   2. Mot de passe trop court, trop long, haché
4. Lors de la suppression d’un utilisateur on vérifie que l’utilisateur a bien été supprimé de la base de données.
5. Lors de la modification d’un utilisateur on vérifie que le mot de passe modifié est bien haché et que les modifications ont bien été changé dans la base de données.
6. Lors de la récupération des utilisateurs on vérifie les cookies, et que l’id retourné est le même que le cookie.
7. Lors du login on vérifie si la route existe et si cela retourne un token valide.

Environnement

Environnement d’intégration pour les tests End to End.

Environnement local pour les tests unitaires.

Données de test

Jeux de donnée de test fourni. (Comprend des valeurs valides, trop longue, vide, attaque d’injection).

Jeux de donnée arbitraire choisi par le testeur.

Risques

Ne pas tester tous les types d’entrée possibles.

Ressources, calendrier

Tests à réaliser avant chaque release.