

## Introduction

Ce document décrit l'approche, la portée, les ressources des tests pour l'application de librairie Express.js / React.js.

## Objectifs

- Valider les différentes fonctionnalités de l'application.
- Garantir des résultats de calcul précis.
- Vérifier la gestion appropriée des saisies utilisateur, y compris les cas limites et les scénarios d'erreur.
- Garantir une expérience utilisateur agréable.

## Rôles et responsabilité

- Développeurs SW
  - Développer les programmes selon les spécifications.
  - Effectuer des tests unitaires pour garantir la qualité du code.
- Responsable Qualité
  - Développer et maintenir les tests pour vérifier les exigences du projet.
  - Exécuter des tests et documenter les défauts.
  - Assurer le respect des normes de qualité.
- Chef de Projet
  - Servir de point de contact principal avec le client.
  - Surveiller l'avancement et ajuster les plans selon les besoins.

## Stratégie de tests

### Niveaux de Tests

- Unitaires : Composants individuels.
- Intégration : Interaction entre modules.
- Régression : Vérifier que les modifications n'introduisent pas de défauts.
- UAT : Validation par les utilisateurs finaux.

### Approche

À chaque push ou pull request sur la branche « main » du repository :

- Les tests du frontend sont exécutés.
- Les tests de l'API sont exécutés.
- Les tests ESLint (contrôle statique du code) sont exécutés.

## Environnement de tests

- Serveur de test : Un serveur configuré pour héberger l'application Express.js / React.js.
- Base de données de test (MySQL) : Une base de données avec des données d'exemples.
- Outils de test :
  - Jest : Pour les tests unitaires et d'intégration.
  - Supertest : Pour tester les endpoints de l'API Express.js.
  - Testing Library : Pour les tests des composants de l'application React.js.
- Docker : Utilisé pour créer des environnements de test isolés et reproductibles. Les conteneurs Docker garantiront que l'application et ses dépendances sont configurées de manière cohérente à travers différents environnements.

## Cas de Test

### IT001

#### Description

Vérifie que l'ajout d'un sujet dans la base de données fonctionne correctement via l'endpoint PUT /subject.

#### Prérequis

- L'application est installée, fonctionnelle et démarrée.
- Base de données initialisée.

#### Étapes

- Accéder à l'endpoint PUT /subject.
- Envoyer une requête avec le nom du sujet.

#### Données

- Nom : Test Subject

#### Résultats attendus

- Le sujet est enregistré dans la base de données avec le nom spécifié.
- La réponse contient l'objet avec le nom du sujet.

## **IT002**

### Description

Vérifie que l'API renvoie une erreur lorsque le nom du sujet est trop long et qu'il n'est pas enregistré dans la base de données.

### Prérequis

- L'application est installée, fonctionnelle et démarrée.
- Base de données initialisée.

### Étapes

- Accéder à l'endpoint PUT /subject.
- Envoyer une requête avec un nom de sujet trop long.

### Données

- Nom : Test Subject Who Is Very Too Long So The Test Should Fail Because The Name Is Too Long To Be Saved In The Database

### Résultats attendus

- La réponse contient une erreur indiquant que le nom est trop long.
- Le sujet n'est pas enregistré dans la base de données.

## **IT003**

### Description

Vérifie que l'API renvoie une erreur lorsque le nom du sujet est vide et qu'il n'est pas enregistré dans la base de données.

### Prérequis

- L'application est installée, fonctionnelle et démarrée.
- Base de données initialisée.

### Étapes

- Accéder à l'endpoint PUT /subject.
- Envoyer une requête avec un nom de sujet vide.

### Données

- Nom : (vide)

### Résultats attendus

- La réponse contient une erreur indiquant que le nom est requis.
- Le sujet n'est pas enregistré dans la base de données.

## IT004

### Description

Vérifie que l'API renvoie une erreur lorsque le corps de la requête est vide.

### Prérequis

- L'application est installée, fonctionnelle et démarrée.
- Base de données initialisée.

### Étapes

- Accéder à l'endpoint PUT /subject.
- Envoyer une requête avec un corps vide.

### Données

Aucune

### Résultat attendu

- La réponse contient une erreur indiquant que le corps de la requête ne doit pas être vide.

## IT005

### Description

Vérifie que l'API renvoie une erreur lorsque le nom du sujet contient des caractères non ASCII et qu'il n'est pas enregistré dans la base de données.

### Prérequis

- L'application est installée, fonctionnelle et démarrée.
- Base de données initialisée.

### Étapes

- Accéder à l'endpoint PUT /subject.
- Envoyer une requête avec un nom de sujet contenant des caractères non ASCII.

### Données

- Nom : Test Subject 😊

### Résultats attendus

- La réponse contient une erreur indiquant que le nom doit uniquement contenir des caractères ASCII.
- Le sujet n'est pas enregistré dans la base de données.

## **IT006**

### Description

Vérifie que l'API renvoie une erreur lorsque le corps de la requête contient des propriétés supplémentaires et que le sujet n'est pas enregistré dans la base de données.

### Prérequis

- L'application est installée, fonctionnelle et démarrée.
- Base de données initialisée.

### Étapes

- Accéder à l'endpoint PUT /subject.
- Envoyer une requête avec des propriétés supplémentaires.

### Données

- Nom : Test Subject With Additional Properties
- Propriété supplémentaire : Property

### Résultat attendu

- La réponse contient une erreur indiquant que le corps ne doit pas contenir de propriétés supplémentaires.
- Le sujet n'est pas enregistré dans la base de données.

## **IT007**

### Description

Vérifie que l'API renvoie le nom du sujet et que ce nom correspond à celui enregistré dans la base de données.

### Prérequis

- L'application est installée, fonctionnelle.
- Base de données initialisée.

### Étapes

- Accéder à l'endpoint GET /subject.
- Envoyer une requête pour obtenir le sujet.

### Données

- Sujet
  - Nom : Math

### Résultats attendus

- La réponse contient le nom du sujet.

- Le nom du sujet renvoyé correspond à celui enregistré dans la base de données.

## **IT008**

### Description

Vérifie que l'entrée est ajoutée lorsqu'on clique sur le bouton "+" dans l'interface utilisateur.

### Prérequis

- L'application est installée et fonctionnelle.

### Étapes

- Cliquer sur le bouton "+" pour ajouter une nouvelle entrée.

### Données

Aucune

### Résultat attendu

- L'endpoint de l'API *POST /grade* est appelé.

## **UT001**

### Description

Vérifie que la fonction de calcul de la moyenne renvoie correctement la moyenne des notes.

### Prérequis

- L'application est installée et fonctionnelle.

### Étapes

- Appeler la fonction avec une liste de notes.

### Données

- Liste des notes :
  - 1 | Nom: Geometry | Note: 6
  - 2 | Nom: Algebra | Note: 3.5
  - 3 | Nom: Trigonometry | Note: 4.5
  - 4 | Nom: Statistics | Note: 5.5
  - 5 | Nom: Linear Algebra | Note: 4
  - 6 | Nom: Differential Equations | Note: 5

### Résultat attendu

- La fonction renvoie la moyenne des notes : 5.