

## I. Objectif

L'objectif de cet environnement de test est de permettre aux développeurs de réaliser des tests d'intégration et d'acceptation d'un service de gestion de tâches en ligne.

## II. Configuration Requise

- Système d'exploitation : Windows, macOS, Linux
- Environnement de développement : IDE (par exemple, Visual Studio Code, IntelliJ IDEA)
- Serveur Web : Apache, Nginx, ou tout autre serveur web compatible avec votre application
- Base de données : MySQL, PostgreSQL, SQLite, ou tout autre système de gestion de base de données compatible avec votre application
- Navigateur Web : Google Chrome, Mozilla Firefox, Safari, etc.
- Outils de test : Postman, JUnit, Jest, Cypress, ou tout autre outil de test adapté à votre service

## III. Installation et Configuration

- 1) Installez et configurez votre serveur web et votre base de données selon les besoins de votre service.
- 2) Téléchargez et configurez votre application de gestion de tâches en ligne dans votre environnement de développement.
- 3) Importez les données de test dans votre base de données, y compris des utilisateurs, des tâches et des configurations, pour simuler un environnement réaliste.
- 4) Installez les outils de test nécessaires dans votre environnement de développement, comme Postman pour les tests d'API ou Jest pour les tests unitaires.

## IV. Scénarios de Test

Voici quelques scénarios de test que vous pouvez réaliser dans cet environnement :

### 1) Test de Connexion :

- Vérifiez que les utilisateurs peuvent se connecter à l'application avec des identifiants valides.
- Vérifiez que les utilisateurs ne peuvent pas se connecter avec des identifiants invalides.

### 2) Test de Création de Tâche :

- Vérifiez que les utilisateurs peuvent créer une nouvelle tâche avec des données valides.
- Vérifiez que les utilisateurs ne peuvent pas créer une tâche sans les informations obligatoires.

### 3) Test d'Intégration API :

- Vérifiez que les différentes parties de votre API interagissent correctement entre elles.
- Vérifiez que les données sont correctement transmises entre les différentes couches de l'application.

### 4) Test de Performance :

- Mesurez le temps de réponse de l'application pour différentes actions, comme la création d'une tâche ou la récupération de données.
- Vérifiez que l'application répond de manière acceptable même sous une charge élevée.

### 5) Test de Validation des Données :

- Vérifiez que les données entrées par les utilisateurs sont validées correctement avant d'être enregistrées dans la base de données.
- Vérifiez que les erreurs de validation sont correctement traitées et renvoyées à l'utilisateur.

## V. Exécution des Tests

- 1) Exécutez les tests unitaires pour vérifier le bon fonctionnement des différentes parties de votre application.
- 2) Exécutez les tests d'intégration pour vérifier que les différentes parties de votre application interagissent correctement entre elles.
- 3) Exécutez les tests d'acceptation pour valider que votre service répond aux exigences fonctionnelles et non fonctionnelles spécifiées.

## VI. Analyse des Résultats

Analysez les résultats des tests pour identifier les éventuels problèmes ou anomalies. Documentez les erreurs rencontrées et travaillez avec votre équipe de développement pour les résoudre. Assurez-vous de mettre à jour les tests en fonction des modifications apportées à l'application.