

CAHIER DE RECETTE

User Stories & Cas de Test (Guide Xray)

Équipe QA :

Mohamed Dhia Eddine Thabet	<i>Test Manager</i>
Mohamed Aziz Dridi	<i>Tester</i>

Novembre 2025

Table des matières

1	Guide d'Utilisation Jira / Xray	3
2	Module 1 : Gestion des Tâches (Frontend)	3
2.1	US-01 : Affichage de la liste des tâches	3
2.2	US-02 : Création d'une tâche	4
2.3	US-03 : Modification d'une tâche	4
2.4	US-04 : Suppression d'une tâche	5
3	Module 2 : API Backend (Intégration)	5
3.1	US-05 : API CRUD complète	5
4	Module 3 : Performance & Charge	6
4.1	US-06 : Robustesse sous charge	6

1 Guide d'Utilisation Jira / Xray

Ce document sert de référence pour la création des tickets dans Jira.

- **Étape 1** : Créer une **User Story** (Type : Story).
- **Étape 2** : Créer un **Test** (Type : Test Xray) lié à la Story.
- **Étape 3** : Dans le Test, remplir les "Test Details" (Steps).
- **Étape 4** : Créer une **Test Execution** pour lancer la campagne.

2 Module 1 : Gestion des Tâches (Frontend)

2.1 US-01 : Affichage de la liste des tâches

Type Jira : User Story

Résumé : En tant qu'utilisateur, je veux voir la liste de mes tâches pour savoir ce que j'ai à faire.

Critères d'Acceptation :

- La liste affiche toutes les tâches non supprimées.
- Si la liste est vide au premier lancement, une tâche par défaut s'affiche.
- La date du jour est affichée en haut.

Test Associé : TC-FRONT-01 (Affichage Initial)

Type Jira : Test (Manual)

Priorité : High

#	Action (Step)	Résultat Attendu (Expected Result)
1	Ouvrir l'application (http://localhost:3000)	La page charge sans erreur console. Le titre "TODO LIST" est visible.
2	Vérifier le contenu de la liste	Au moins une tâche est présente (ex : "Click to edit task name").
3	Vérifier la date	La date affichée correspond à la date système actuelle.

2.2 US-02 : Création d'une tâche

Type Jira : User Story

Résumé : En tant qu'utilisateur, je veux ajouter une nouvelle tâche pour planifier mon travail.

Test Associé : TC-FRONT-02 (Ajout de Tâche)

Type Jira : Test (Manual / Automated Cypress)

#	Action (Step)	Résultat Attendu (Expected Result)
1	Cliquer sur le bouton "Add task"	Une nouvelle ligne apparaît instantanément dans la liste.
2	Vérifier le texte par défaut	La nouvelle tâche contient le texte "Click to edit task name".
3	Recharger la page (F5)	La nouvelle tâche est toujours présente (Persistance).

2.3 US-03 : Modification d'une tâche

Type Jira : User Story

Résumé : En tant qu'utilisateur, je veux modifier le nom d'une tâche pour corriger ou préciser son contenu.

Test Associé : TC-FRONT-03 (Édition)

Type Jira : Test (Manual)

#	Action (Step)	Résultat Attendu (Expected Result)
1	Cliquer sur le texte d'une tâche existante	Le champ devient éditable (Focus).
2	Saisir "Acheter du lait"	Le texte s'affiche correctement pendant la frappe.
3	Appuyer sur "Entrée" ou cliquer ailleurs	La modification est sauvegardée.

2.4 US-04 : Suppression d'une tâche

Type Jira : User Story

Résumé : En tant qu'utilisateur, je veux supprimer une tâche terminée ou inutile.

Test Associé : TC-FRONT-04 (Suppression Standard)

Type Jira : Test (Manual)

#	Action (Step)	Résultat Attendu (Expected Result)
1	Identifier une tâche à supprimer	Le bouton "Poubelle" est visible à droite de la tâche.
2	Cliquer sur l'icône "Poubelle"	La tâche disparaît de la liste.

Test Associé : TC-FRONT-05 (Suppression Totale - Bug Zombie)

Type Jira : Test (Exploratory)

Note : Ce test est conçu pour reproduire le Bug #002.

#	Action (Step)	Résultat Attendu (Expected Result)
1	Supprimer TOUTES les tâches de la liste	La liste doit être vide.
2	Recharger la page (F5)	La liste doit rester vide ("Inbox Zero").
KO	<i>Observation Réelle</i>	Une tâche par défaut réapparaît automatiquement.

3 Module 2 : API Backend (Intégration)

3.1 US-05 : API CRUD complète

Type Jira : User Story (Technical)

Résumé : Le système doit fournir une API REST pour gérer les tâches.

Test Associé : TC-API-01 (Create Task)

Type Jira : Test (Automated - JUnit/Postman)

Précondition : Le serveur est lancé sur le port 8080.

Méthode : POST

Endpoint : /api/todoItems

Body : (Vide)

Résultat Attendu : Code 200 OK. Retourne un JSON avec "id": X, "task": "Click to edit task name".

Test Associé : TC-API-02 (Update Task)

Type Jira : Test (Automated)

Méthode : PUT

Endpoint : /api/todoItems/{id}

Body : { "task": "Updated", "isDone": true }

Résultat Attendu : Code 200 OK. L'objet retourné contient les nouvelles valeurs.

4 Module 3 : Performance & Charge

4.1 US-06 : Robustesse sous charge

Type Jira : Non-Functional Requirement

Résumé : Le système doit supporter 50 utilisateurs simultanés sans perte de données.

Test Associé : TC-PERF-01 (Stress Test Concurrency)

Type Jira : Test (Performance - JMeter)

#	Action (Step)	Résultat Attendu (Expected Result)
1	Configurer JMeter avec 50 Threads	Configuration prête.
2	Lancer le test "Create Task" (Ramp-up 0s)	Tous les threads envoient un POST simultanément.
3	Vérifier les IDs générés	Chaque tâche doit avoir un ID unique.
KO	<i>Observation Réelle</i>	Des IDs dupliqués sont observés (Bug #001).