LabXpert - Système de Gestion pour Laboratoire Médical

Le projet LabXpert, un système de gestion développé par TechLab pour étendre les capacités opérationnelles d'un laboratoire médical. Ce projet vise à fournir des services et des API permettant une intégration facile avec d'autres systèmes, un accès sécurisé aux données, et la mise en œuvre de pratiques de développement modernes.

Fonctionnalités du Projet

Le système LabXpert offre les fonctionnalités suivantes:

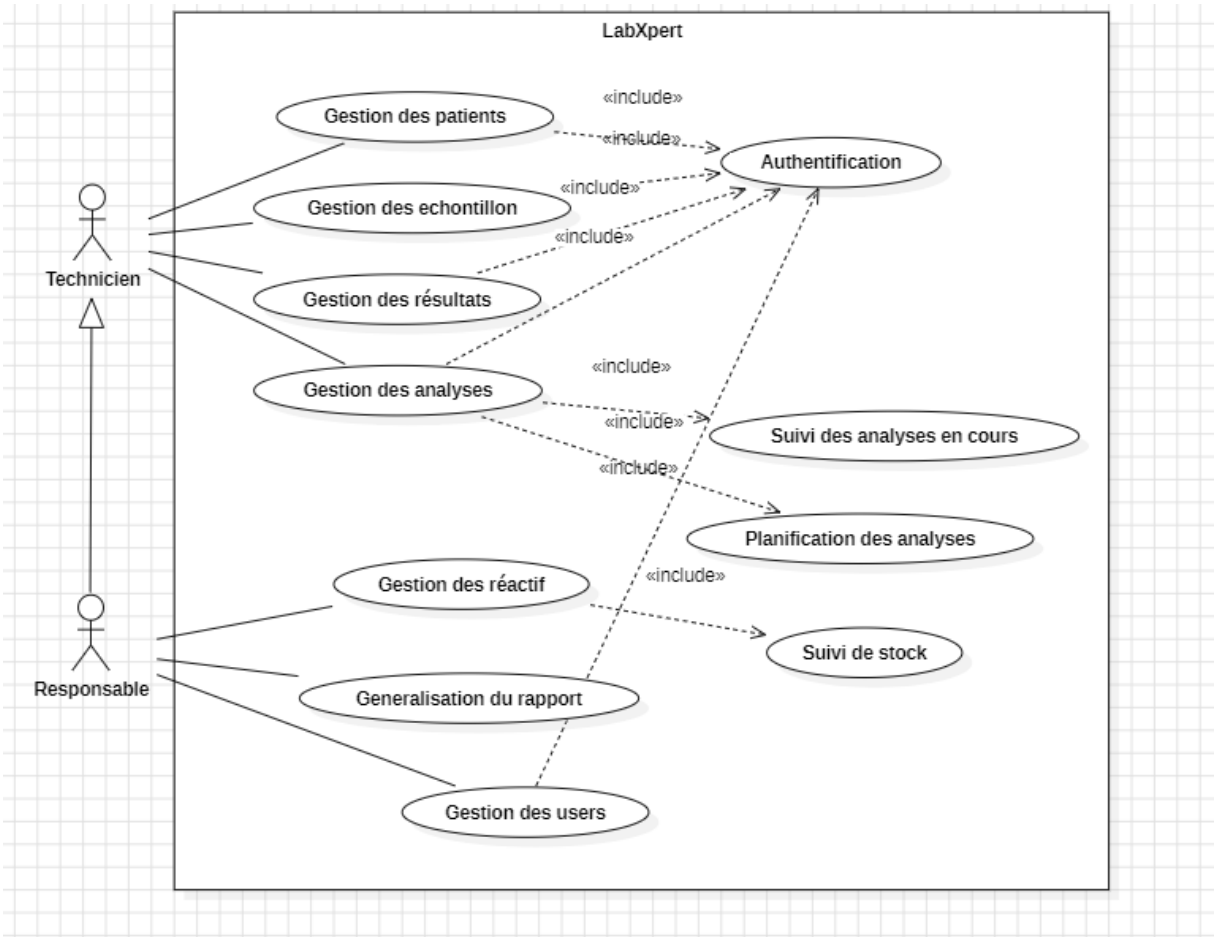


Diagramme cas d'utilisation

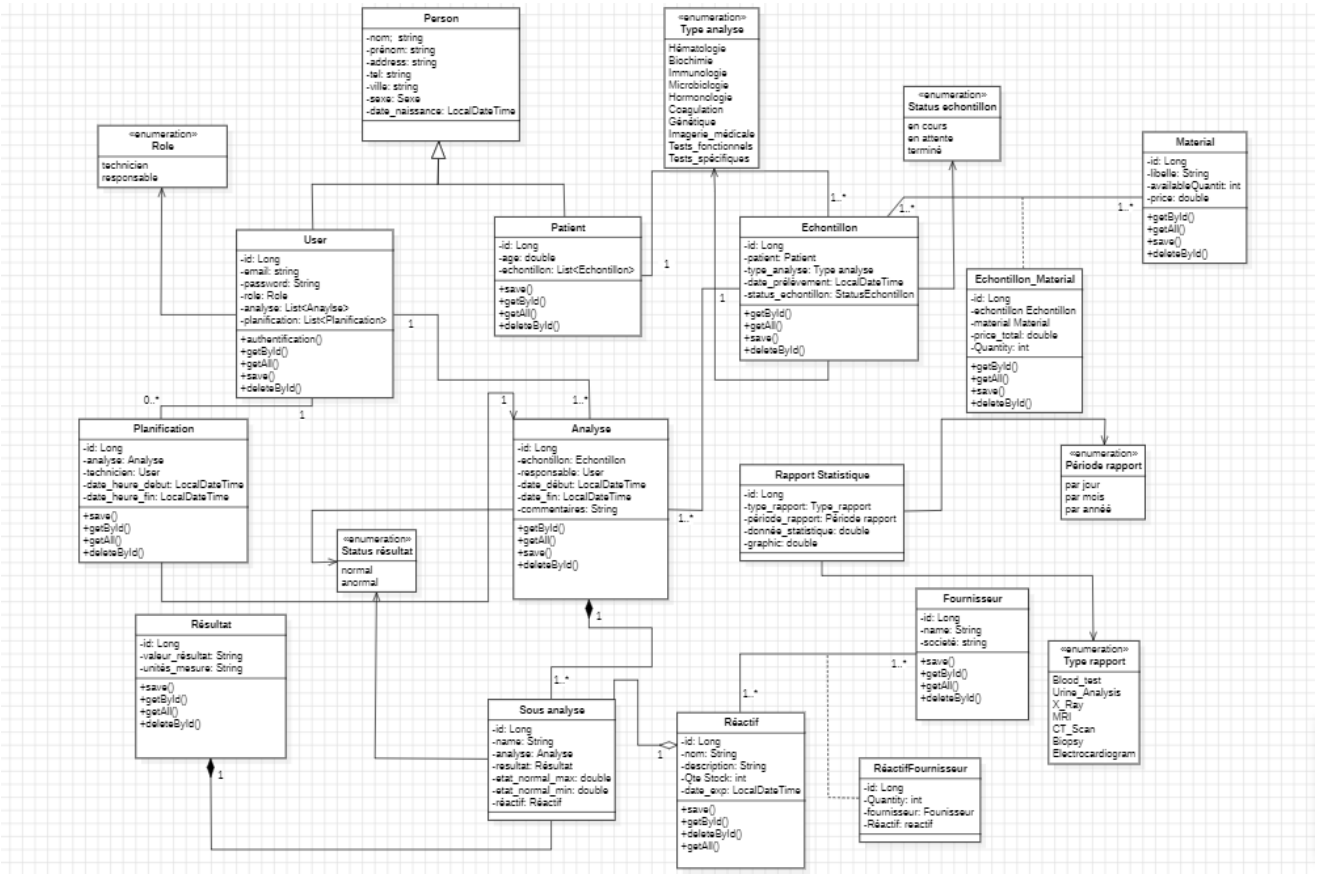


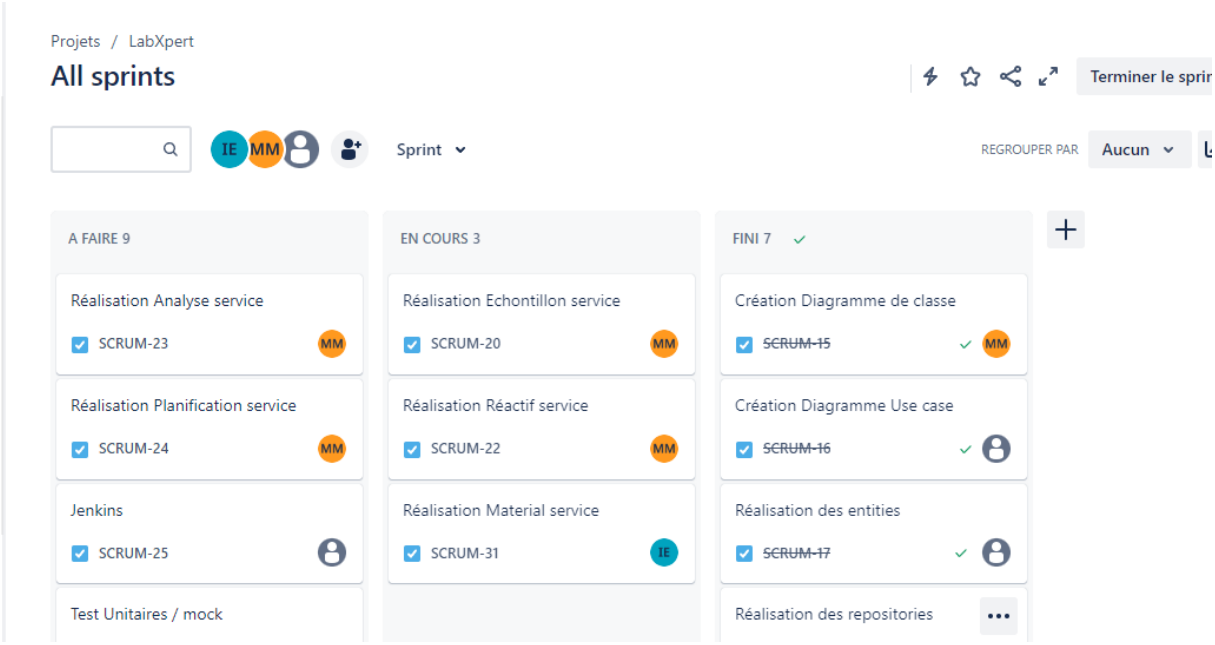
Diagramme de Classes

Stack Technique

- Langage de Programmation : Java Backend : Spring Boot API RESTful Gestion de Dépendances : Apache Maven Base de Données : PostgreSQL Serveur d'Application : Apache Tomcat Testing : JUnit & Mockito CI/CD : Jenkins Gestion des Tâches : Jira Système de Gestion de Version : Git et Github Documentation de l'API : Swagger

Implémentation Backend

1. Jira est l'outil choisi pour gérer les tâches du projets:

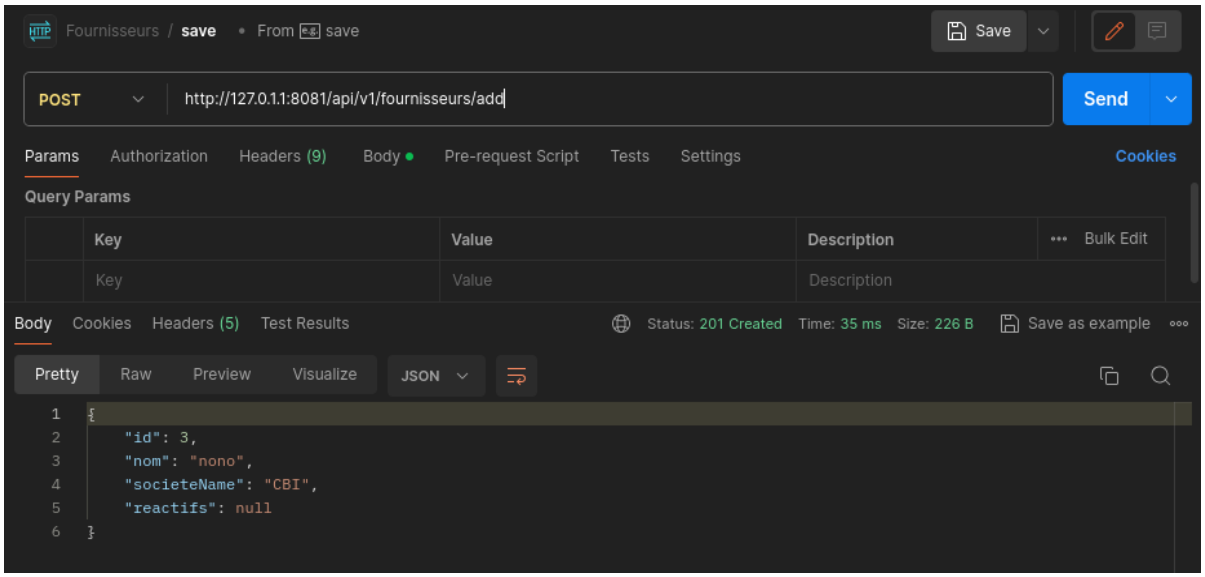


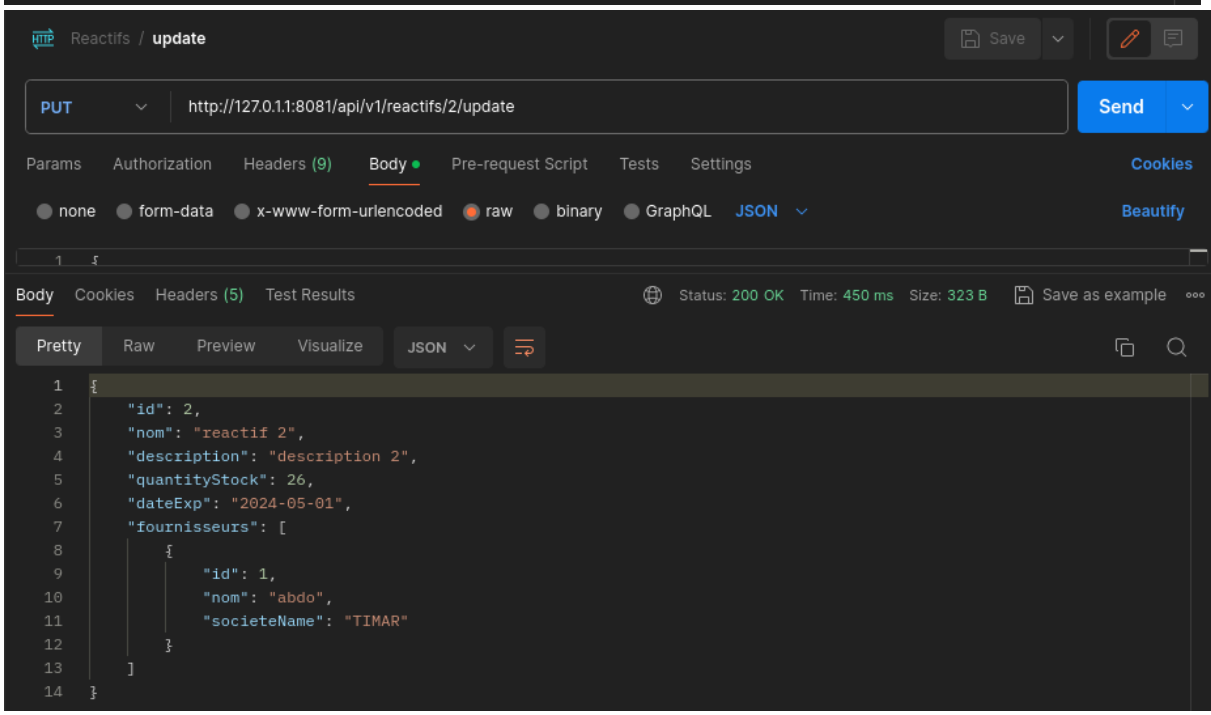
2. Implémentation Backend (Partie 2) :

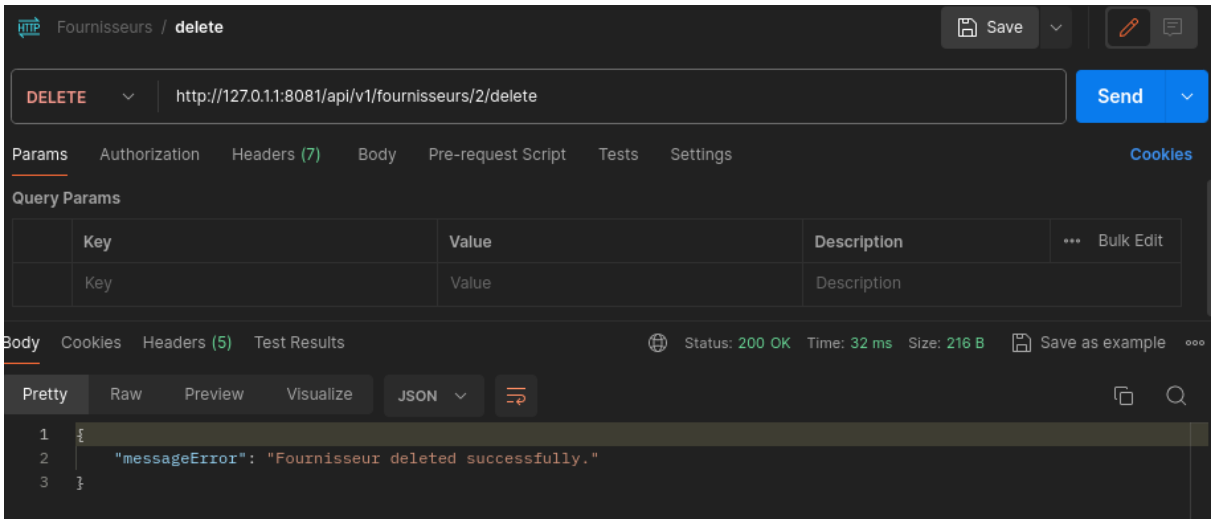
- Création des services et des contrôleurs pour la gestion des API REST.

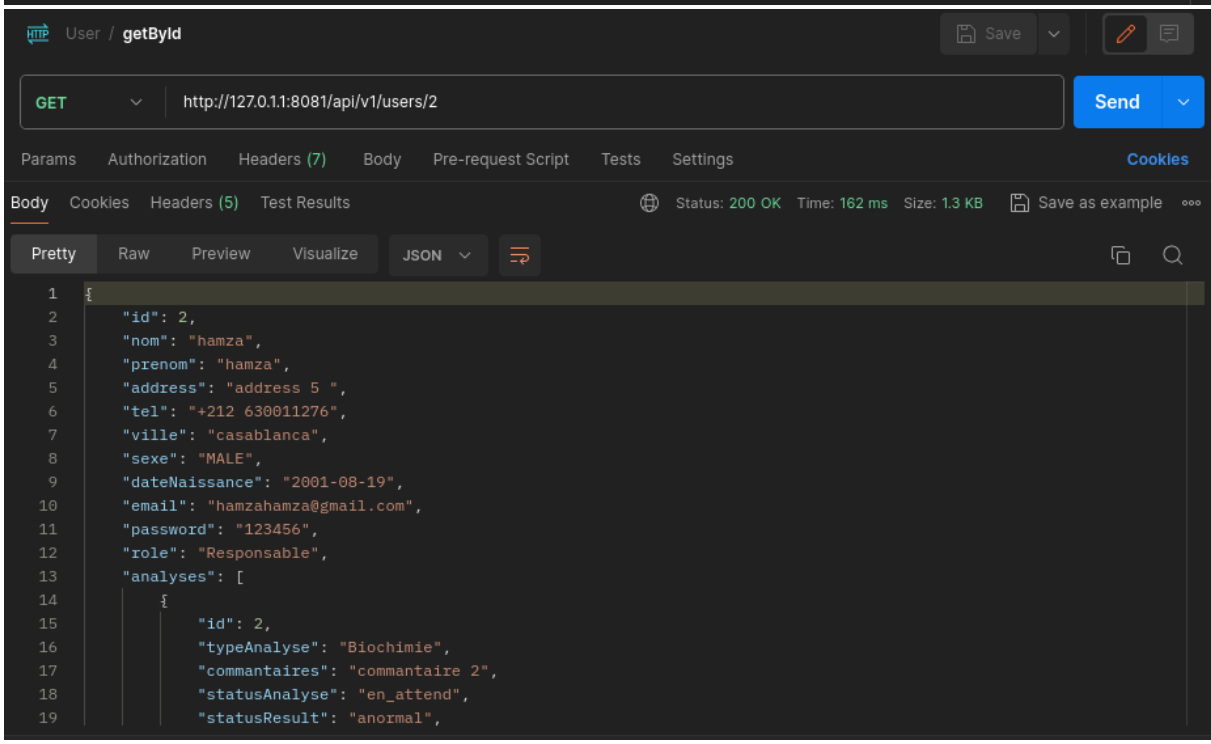
3. Tests des APIs avec POSTMAN :

- Captures d'écran des tests avec POSTMAN.


Fournisseurs / save • From save
Save
POST
http://127.0.1.1:8081/api/v1/fournisseurs/add
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
Query Params
Key Value Description Bulk Edit
Key Value Description
Body Cookies Headers (5) Test Results
Status: 201 Created Time: 35 ms Size: 226 B
Save as example
Pretty Raw Preview Visualize JSON
{
"id": 3,
"nom": "nono",
"societeName": "CBI",
"reactifs": null
}

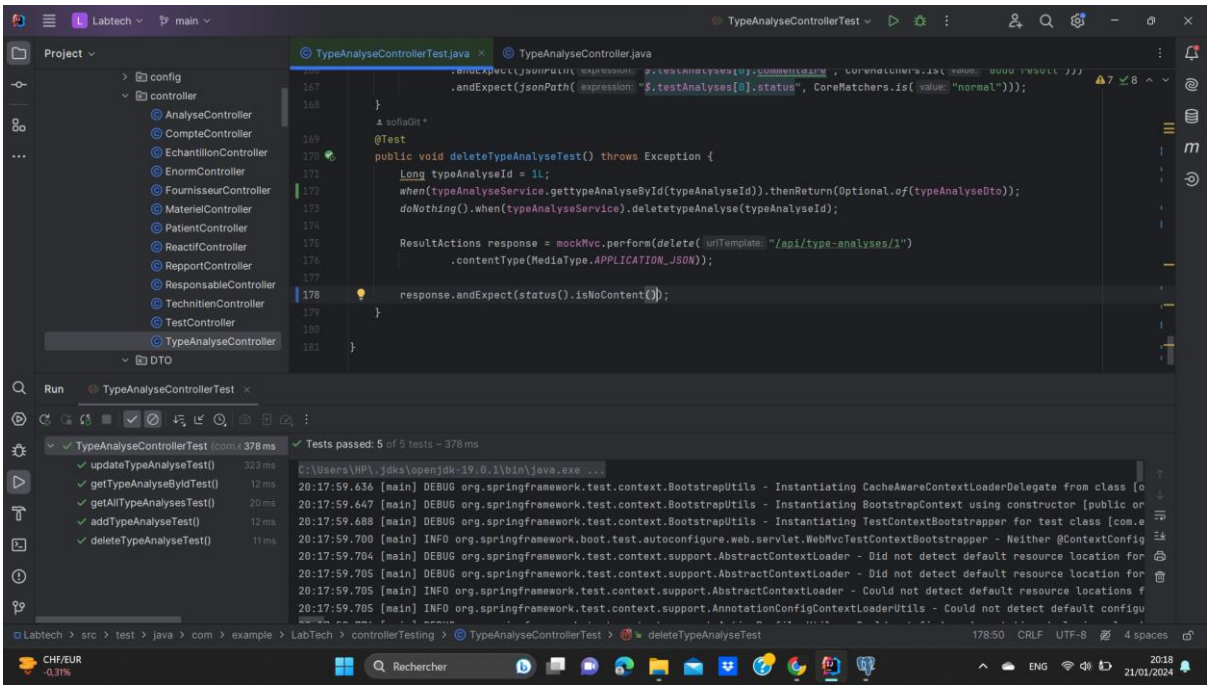

Reactifs / update
Save
PUT
http://127.0.1.1:8081/api/v1/reactifs/2/update
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 450 ms Size: 323 B
Save as example
Pretty Raw Preview Visualize JSON
{
"id": 2,
"nom": "reactif 2",
"description": "description 2",
"quantityStock": 26,
"dateExp": "2024-05-01",
"fournisseurs": [
{
"id": 1,
"nom": "abdo",
"societeName": "TIMAR"
}
]
}





4. **Tests Backend (Unitaires et d'Intégration) :**
   - Écriture des tests unitaires pour le backend.
   - Réalisation des tests d'intégration.
   - Captures d'écran des résultats des tests

- *PatientControllerTest*



5. Intégration avec Jenkins

- Configuration des scripts pour l'intégration continue.
- Captures d'écran du pipeline Jenkins.

Structure du Projet

Le code source est organisé en plusieurs contrôleurs et services, chacun dédié à une fonctionnalité spécifique du laboratoire. Les tests unitaires et d'intégration garantissent la qualité du code, et la documentation Swagger fournit des informations détaillées sur les API.

Mise en Place du CI/CD

Le fichier Jenkinsfile définit la pipeline CI/CD pour le projet. Chaque étape de la pipeline est dédiée à tester un service spécifique du laboratoire. Le code est automatiquement récupéré depuis le référentiel GitHub, nettoyé, testé, et déployé sur un environnement de test.