

# LabXpert - Système de Gestion pour Laboratoire Médical

Le projet LabXpert est un système de gestion développé par TechLab pour augmenter les capacités opérationnelles d'un laboratoire médical. Il propose des services et des API pour faciliter l'intégration avec d'autres systèmes, assurer un accès sécurisé aux données, et mettre en œuvre des pratiques de développement modernes

## Stack Technique

Langage de programmation : Java Framework Backend : Spring Boot Type d'API RESTful Outil de gestion des dépendances : Apache Maven Système de gestion de base de données : PostgreSQL

Serveur d'applications : Apache Tomcat

Tests : JUnit & Mockito Intégration continue / Délivrance continue : Jenkins Gestion des tâches : Jira Système de contrôle de version : Git et Github Documentation de l'API : Swagger

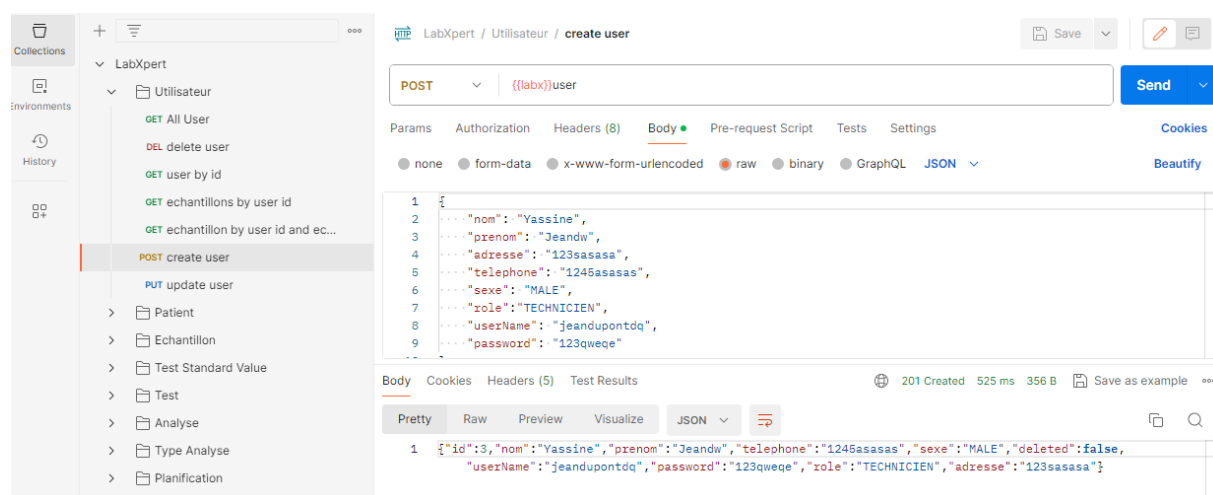
## Implémentation Backend

### 1. Implémentation Backend (Partie 2) :

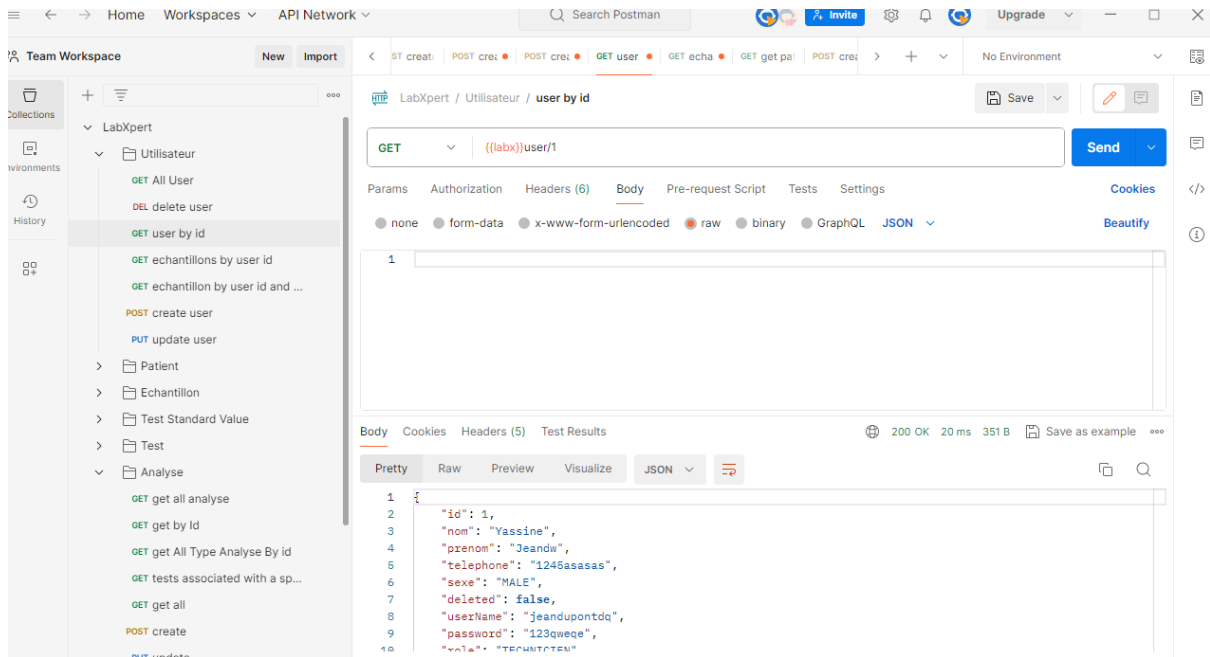
- Création des services et des contrôleurs pour la gestion des API REST.

### 2. Tests des APIs avec POSTMAN :

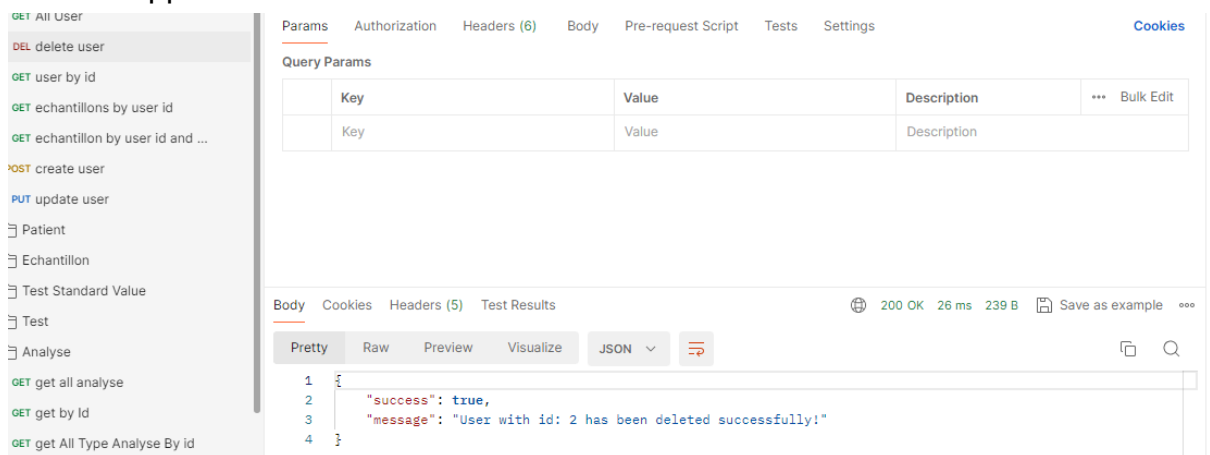
- Captures d'écran des tests avec POSTMAN.  
screne pour ajouter un utilisateur



## Afficher un utilisateur par id



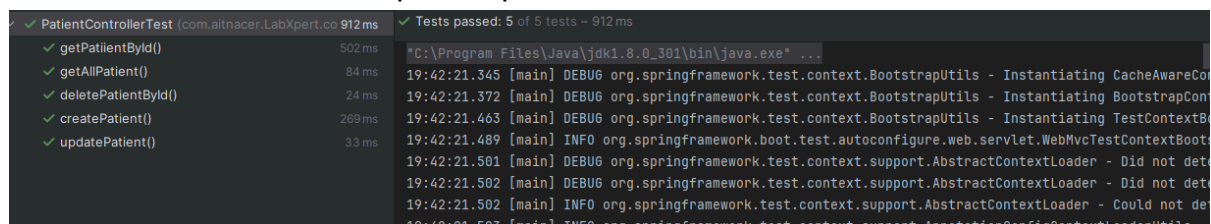
## Supprimer un utilisateur



### 3. 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

voila un test sur controller patient par Mockito



## un test sur fournisseur

```
✓ FournisseurServiceImplTest (com.altnacer.) 1 sec 742 ms
  ✓ deleteFournisseur() 1 sec 574 ms
  ✓ createFournisseur() 94 ms
  ✓ getFournisseurById() 30 ms
  ✓ updateFournisseur() 10 ms
  ✓ getAllFournisseur() 34 ms
  Tests passed: 5 of 5 tests - 1 sec 742 ms

"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe" ...
FournisseurDTO(idFournisseur=1, nom=salma, adresse=casa, tel=1234567, deleted=false)
FournisseurDTO(idFournisseur=1, nom=salma, adresse=casa, tel=1234567, deleted=false)
FournisseurDTO(idFournisseur=1, nom=salma, adresse=casa, tel=1234567, deleted=false)
Process finished with exit code 0
```

## 4. Documentation par Swagger:

- Captures d'écran de l'interface SWAGGER avec la documentation automatique de l'API.

The screenshot displays the Swagger API Documentation interface. At the top, the Swagger logo is visible, along with the text 'Supporting SMARTBEAR'. A dropdown menu labeled 'Select a definition' is set to 'default'. The main heading is 'Api Documentation 1.0', with a base URL of 'localhost:8088/' and a link to 'http://localhost:8088/v2/api-docs'. Below this, there are links for 'Api Documentation', 'Terms of service', and 'Apache 2.0'.

The interface lists several controllers, each with a right-pointing arrow:

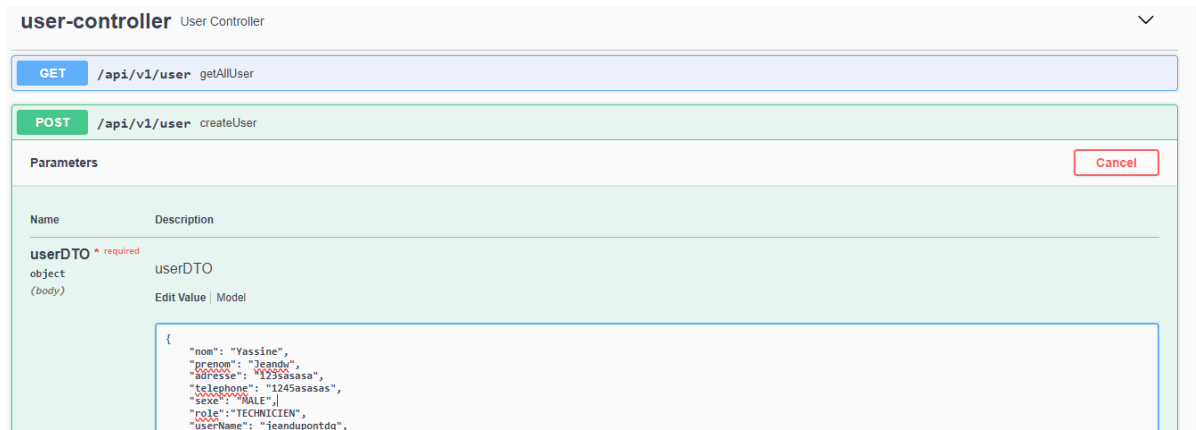
- analyse-controller** Analyse Controller
- echantillon-controller** Echantillon Controller
- fournisseur-controller** Fournisseur Controller
- patient-controller** Patient Controller

The 'fournisseur-controller' is expanded, showing a list of API endpoints with their respective HTTP methods:

- GET** /api/v1/fournisseur getAllFournisseur
- POST** /api/v1/fournisseur createFournisseur
- GET** /api/v1/fournisseur/{id} getFournisseurById
- PUT** /api/v1/fournisseur/{id} updateFournisseur
- DELETE** /api/v1/fournisseur/{id} delete

Below the expanded controller, there are more controllers listed with right-pointing arrows:

- patient-controller** Patient Controller
- reactif-controller** Reactif Controller
- result-controller** Result Controller



La documentation complète du code est disponible via Swagger. Accédez à <http://localhost:8080/swagger-ui/index.html> après le déploiement de l'application.

## 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.