

Documrntation LabXpert l'API REST via Spring Boot (Part 2)



Introduction

LabXpert représente une avancée majeure dans l'optimisation des opérations du laboratoire médical de TechLab. Ce document vise à fournir une documentation complète sur les services et technologies intégrés dans le projet.

Services Disponibles

1. Services d'Échantillons

- Enregistrement d'Échantillons: Permet d'enregistrer de nouveaux échantillons avec les détails du patient, le type d'analyse, et la date de prélèvement.
- **Récupération d'Échantillons:** Permet de récupérer les détails d'un échantillon spécifique.

2. Services d'Analyses

- Suivi en Temps Réel: Service pour suivre l'état d'avancement des analyses en cours en temps réel.
- Lancement d'Analyse: Permet de lancer une nouvelle analyse avec les détails associés.
- **Récupération des Résultats:** Service pour récupérer les résultats d'une analyse spécifique.

3. Services de Gestion des Résultats

- Consignment des Résultats: Service pour consigner de manière systématique les résultats des analyses.
- Partage des Résultats: Permet de partager les résultats avec les professionnels de la santé.

4. Services de Gestion des Patients

- **Gestion des Informations Patient:** Service pour gérer les informations relatives aux patients.
- **Historique des Analyses:** Permet de récupérer l'historique des analyses d'un patient spécifique.

5. Services d'Inventaire des Réactifs

- Suivi des Stocks: Service pour suivre les stocks de réactifs.
- **Notifications de Stock Basse:** Permet de notifier lorsque les réactifs sont bas.

6. Services de Gestion des Utilisateurs

• Administration des Utilisateurs: Service d'administration pour gérer les droits d'accès et les informations des utilisateurs.

7. Services de Planification des Analyses

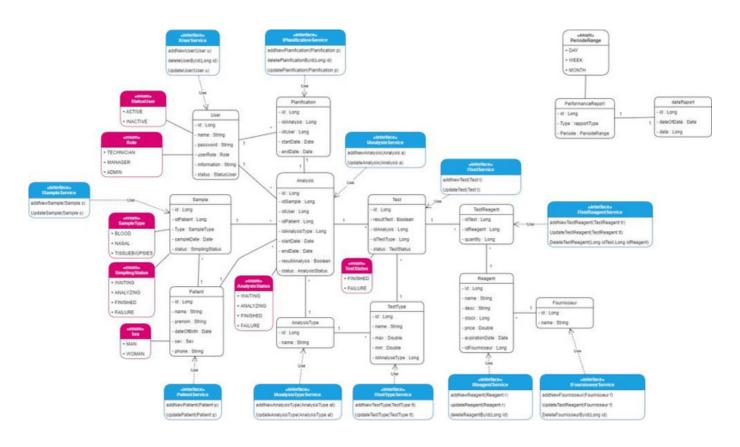
• **Planification des Analyses:** Service pour planifier les analyses en fonction de la charge de travail.

8. Services de Rapports Statistiques

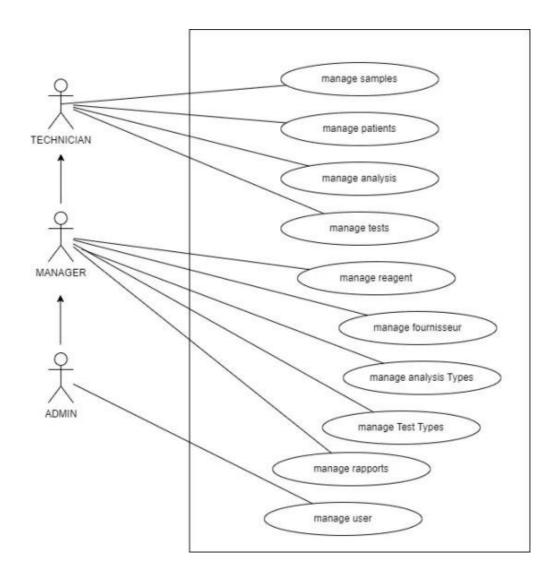
• **Génération de Rapports:** Service pour générer des rapports statistiques sur les performances du laboratoire.

Diagrammes

1. Diagramme de classes



2. Diagramme de cas d'utilisation



Technologies et outils techniques

- Langage de Programmation: Java
- Backend: Spring Boot
- API RESTful
- Gestion de Dépendances: Apache Maven
- Base de Données: PostgreSQL (H2 pour test)
- Serveur d'Application: Apache Tomcat
- Testing: JUnit & Mockito
- CI/CD: Jenkins
- Gestion des Tâches: Outil de votre choix
- Système de Gestion de Version: Git et Github
- Documentation de l'API: Swagger

Jenkins

Stage View

	Declarative:	Declarative:		2:03		Declarative
	Checkout SCM	Tool Install	Checkout	Build	Test	Post Actions
Average stage times: (Average <u>full</u> run time: ~1min 24s)	25	380ms	25	9s	1min 23s	194ms
Jan 23 4 14:51 commits	2s	355ms	2s	7s	1min 24s	256ms
Jan 23 No 14:49 Changes						
Jan 23 1 13:06 commit	2s	245ms	2s	6s	48s	139ms
Jan 23 No 13:04 Changes	2s	372ms	25	75	1min 0s	147ms
Jan 23 No 12:50 Changes	2s	549ms	3s	18s	2min 19s	236ms

Unit test

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 44, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 36.758 s
[INFO] Finished at: 2024-01-23T20:07:45+01:00
[INFO]
```

git

```
International continues and co
```

Postman

