

PROJET JEE

CAHIER DE CHARGE-V2

SUJET: SECUREAUTH+ - PLATEFORME CENTRALISÉE DE GESTION DES IDENTITÉS ET DES ACCÈS (IAM)

**REALISER PAR :**

- Mohamed AZZAM
- MAJGHIROU MOHAMED RIYAD



Table de Matiere



Problématique **3**

Solution proposée **4**

Présentation générale du système **5**

Analyse des cas d'utilisation du système **6**

Architecture fonctionnelle **9**

Stack technique utilisée **10**

Problématique



Comment assurer une gestion centralisée, sécurisée et traçable des utilisateurs et de leurs accès dans une application web, tout en respectant les principes de cybersécurité moderne ?

Sous-questions possibles :

- Comment authentifier les utilisateurs de manière fiable ?
- Comment attribuer les bons rôles à chaque profil ?
- Comment éviter les accès non autorisés ou abusifs ?
- Comment tracer les actions sensibles réalisées dans le système ?

Solution proposée



Pour répondre à cette problématique, nous avons conçu un système de gestion des identités et des accès (IAM) basé sur le framework Spring Boot.

Ce système permet à un administrateur de gérer les utilisateurs, de définir leurs rôles, de contrôler les accès aux fonctionnalités sensibles et d'assurer un suivi des actions grâce à un module de traçabilité.

Ce projet vise à développer une **plateforme d'identité et de gestion des accès centralisée (IAM)** réutilisable par plusieurs applications internes.

Les principaux objectifs sont :

- Assurer une **authentification sécurisée avec JWT** et refresh token.
- Permettre une **gestion avancée des rôles et permissions**.
- Garantir la **traçabilité complète** des connexions et actions sensibles.
- Fournir une **API REST documentée et extensible** pour d'autres applications.
- Offrir une **interface d'administration** claire pour la gestion des utilisateurs et logs.

Présentation générale du système



Domaine	Fonctionnalité	Description
Authentification	Login / Register avec vérification d'email	Génère un JWT sécurisé et un refresh token
Gestion des utilisateurs	CRUD + activation/désactivation	L'admin gère les comptes et leurs rôles
Audit & Sécurité	Journalisation des connexions et actions	Enregistre chaque login, échec ou changement de rôle
Protection	Blocage automatique après 3 tentatives	Défense contre les attaques par force brute
Interface Admin	Dashboard des utilisateurs, rôles et logs	Interface claire et filtrable pour l'admin
Documentation	Swagger / OpenAPI	Pour tester et intégrer facilement les endpoints

Analyse des cas d'utilisation du système



1. Utilisateur

Cas d'utilisation	Description
S'inscrire	L'utilisateur saisit son nom, prénom et email pour demander la création d'un compte.
Se connecter	Authentification via username et mot de passe. En cas de 3 échecs, le compte est bloqué.
Modifier son profil	Mise à jour de ses informations personnelles (nom, email, etc.).
Mettre à jour son mot de passe	L'utilisateur change volontairement son mot de passe depuis son espace.

2. Manager

Cas d'utilisation	Description
Consulter la liste des utilisateurs	Visualise les comptes actifs, bloqués ou en attente.
Modifier les rôles d'un utilisateur	Change le rôle d'un utilisateur dans son périmètre (ex : de USER à MANAGER).
Désactiver un compte temporairement	Suspend un compte sans le supprimer. Utilisé pour les collaborateurs inactifs.

3. Administrateur

Cas d'utilisation	Description
Valider une inscription	Accepte ou rejette une demande de création de compte. Génère un username et un mot de passe temporaire.
Réinitialiser un mot de passe oublié	Crée un nouveau mot de passe temporaire pour un utilisateur.
Bloquer / Débloquer un compte	Intervient en cas de blocage automatique ou sur demande d'un utilisateur.
Créer un rôle personnalisé	Définit un nouveau rôle avec permissions spécifiques.
Consulter la liste des utilisateurs	Accède à la liste complète des comptes avec filtres (actif, bloqué, en attente).
Modifier les rôles d'un utilisateur	Attribue ou retire un rôle à un utilisateur.
Désactiver un compte temporairement	Gèle l'accès à un compte sans suppression définitive.
Consulter les logs d'activité	Visualise l'historique des connexions, actions et erreurs.
Exporter un rapport d'audit	Génère un rapport détaillé des logs et événements de sécurité.
Recevoir une alerte de blocage	Est notifié lorsqu'un compte est bloqué après 3 échecs de connexion.
Valider les tentatives échouées	Peut débloquer manuellement un compte après justification.

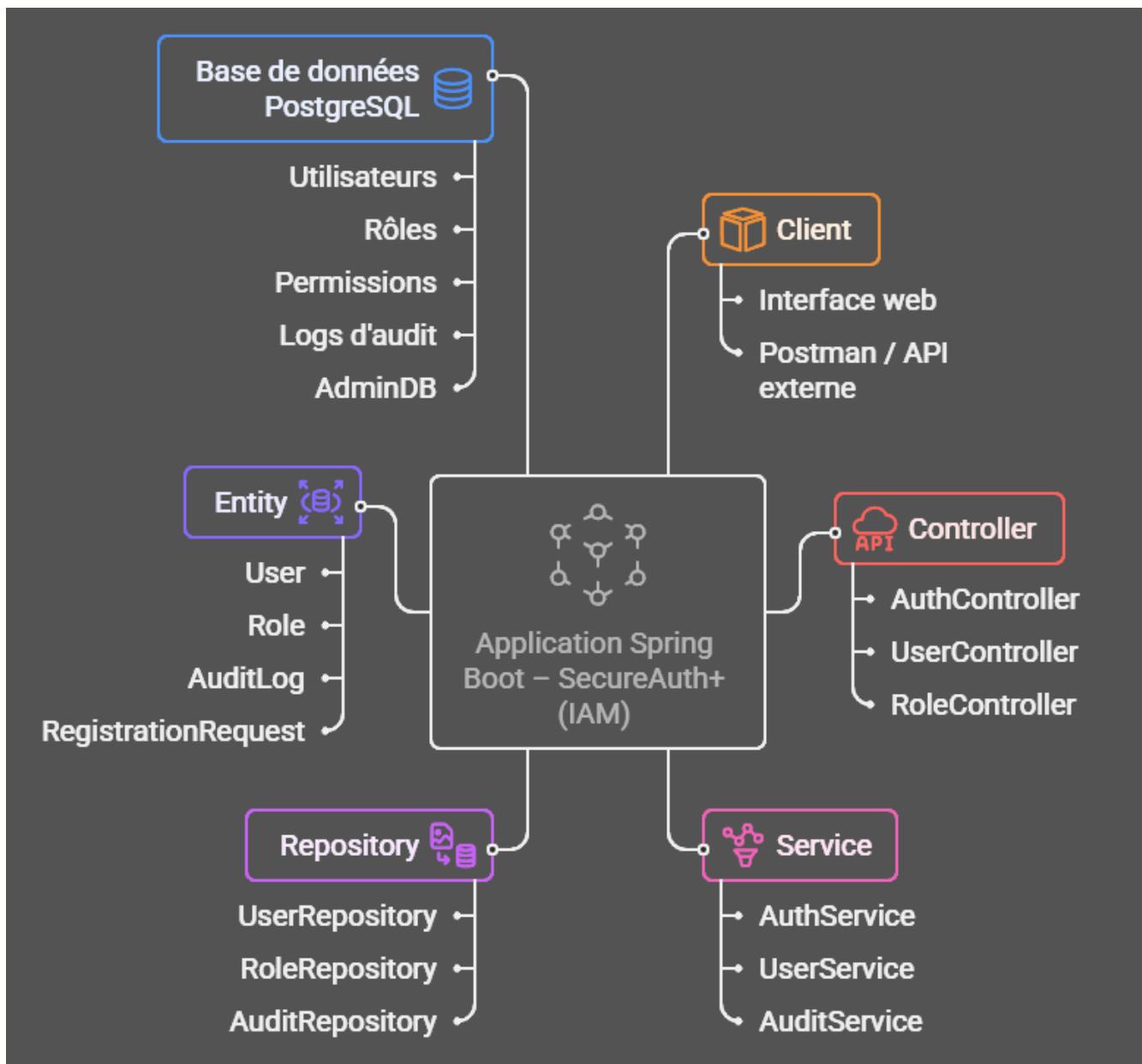
4. Responsable Sécurité

Cas d'utilisation	Description
Consulter les logs d'activité	Analyse les événements liés à la sécurité (connexions, blocages, changements de rôles).
Analyser les connexions suspectes	Examine les tentatives d'accès inhabituelles ou venant d'adresses IP non reconnues.
Exporter un rapport d'audit	Génère un rapport complet à des fins d'audit interne.
Recevoir une alerte de blocage	Est averti lorsqu'un utilisateur est bloqué pour activité suspecte ou échec répété.
Bloquer / Débloquer un compte	Peut intervenir sur les comptes suspects en coordination avec l'administrateur.

Relations transversales (communes à plusieurs rôles)

Fonction	Rôles impliqués	Description
Connexion / Authentification	Tous	Tous les acteurs se connectent via le même module d'authentification.
Traçabilité / Logs	Admin, Responsable Sécurité	Chaque action est enregistrée dans la base d'audit.
Blocage après 3 tentatives	Tous les utilisateurs	Le système bloque le compte automatiquement après 3 erreurs d'authentification.
Alerte de sécurité	Admin, Responsable Sécurité	Notification en cas de comportement anormal ou tentative d'accès multiple.

Architecture fonctionnelle



- L'architecture 3-tiers (Controller, Service, Repository) est appliquée, avec une communication interne assurée par des API REST sécurisées.

Stack technique utilisée



Catégorie	Technologie
Framework principal	Spring Boot 3.5.6
Langage	Java 17
Base de données	PostgreSQL /MongoDb
Sécurité	Spring Security + JWT + BCrypt
ORM	Spring Data JPA
Documentation	Swagger / OpenAPI
Outils de test	Postman
IDE	IntelliJ IDEA / VS Code
Gestion de dépendances	Maven

Conclusion



Summary

Ce projet met en pratique les notions fondamentales de la cybersécurité dans le développement web moderne.

Il démontre comment Spring Boot permet d'intégrer la gestion des rôles et permissions au cœur d'une application sécurisée.

Perspectives :

- Intégration d'une authentification à deux facteurs (2FA).
- Liaison avec LDAP / Active Directory.
- Interface web d'administration (React / Angular).
- Intégration d'un système d'alertes de sécurité.