Documentation Technique — Système d'authentification sécurisé

Présentation du Projet

Le Système d'authentification sécurisé est une API backend robuste développée avec Node.js et Express, permettant la gestion complète des utilisateurs avec des fonctionnalités avancées de sécurité. L'objectif est d'offrir une architecture réutilisable pour toute application nécessitant une authentification avec rôles et gestion des accès.

Fonctionnalités

- Authentification via JWT (access + refresh tokens)
- fall création, connexion et suppression d'utilisateurs
- Kecupération de mot de passe avec jeton sécurisé
- 🔐 Bcrypt pour le hashage des mots de passe
- 🖈 Séparation claire des rôles (user, admin, etc.)
- Middleware de protection des routes selon les rôles
- Journalisation des connexions (IP, date, user-agent)
- Fichiers de tests ou collection Postman fournis

Design API & Architecture

- Architecture modulaire : controllers/, routes/, middlewares/, models/
- Connexion MongoDB via Mongoose
- Endpoints RESTful clairs (/api/auth, /api/users)
- Sécurité via CORS, rate limiting, gestion d'erreurs centralisée

Architecture Technique

- Backend : Node.js + Express
- Sécurité : JWT (Access + Refresh), Bcrypt, validation
- **Base de données : MongoDB avec schémas Mongoose**
- Environnement : Fichiers .env pour config
- Outils de test recommandés : Postman, Jest ou Insomnia

X Technologies utilisées

- Node.js, Express.js
- JWT (jsonwebtoken), Bcrypt
- MongoDB, Mongoose
- dotenv, CORS, Helmet

Gestion des Rôles Utilisateurs

Le système met en place trois rôles distincts avec des droits d'accès hiérarchisés, pour simuler une gestion réaliste des utilisateurs dans une application sécurisée.

Administrateur

- Identifiants de test :
 - admin@secure.com / admin123
- Permissions:
 - Accès total à toutes les routes protégées
 - Création / suppression de comptes utilisateurs
 - Attribution et modification des rôles
 - Consultation des logs de connexion
 - o Réinitialisation de mots de passe pour tout le monde
- Utilisation : Pour gérer l'application de manière globale (gestion des utilisateurs, sécurité, monitoring)

Modérateur

• Identifiants de test :

moderator@secure.com / mod123

- Permissions:
 - Accès partiel aux routes protégées
 - Lecture des utilisateurs
 - Possibilité de désactiver temporairement un compte
 - o Peut voir les logs mais pas modifier les rôles
- Utilisation : Pour encadrer la communauté ou l'équipe sans avoir de privilèges critiques

Utilisateur standard

Identifiants de test :

user@secure.com / user123

- Permissions:
 - Accès à son propre profil
 - Modifier ses informations personnelles
 - Changer son mot de passe
 - Accès restreint aux fonctionnalités de l'application
- Utilisation : Pour accéder aux services de l'application tout en restant limité en termes de permissions

Vue sur le projet:

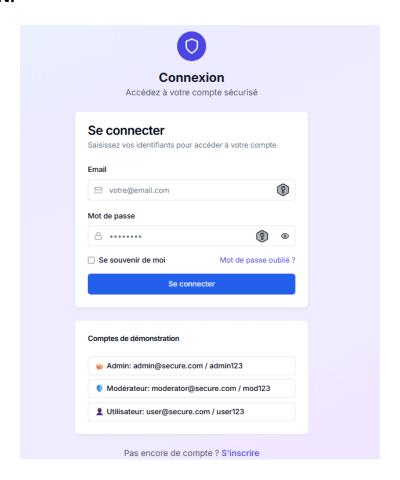
Comptes de démonstration

Admin: admin@secure.com / admin123

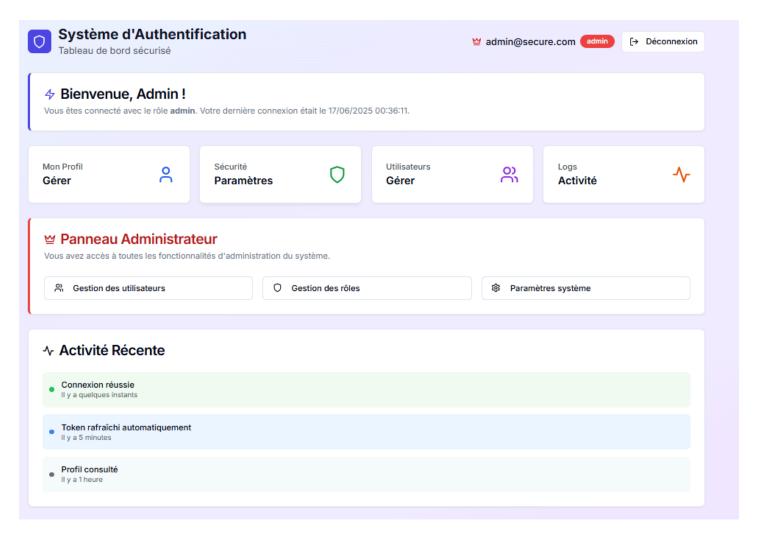
Modérateur: moderator@secure.com / mod123

1 Utilisateur: user@secure.com / user123

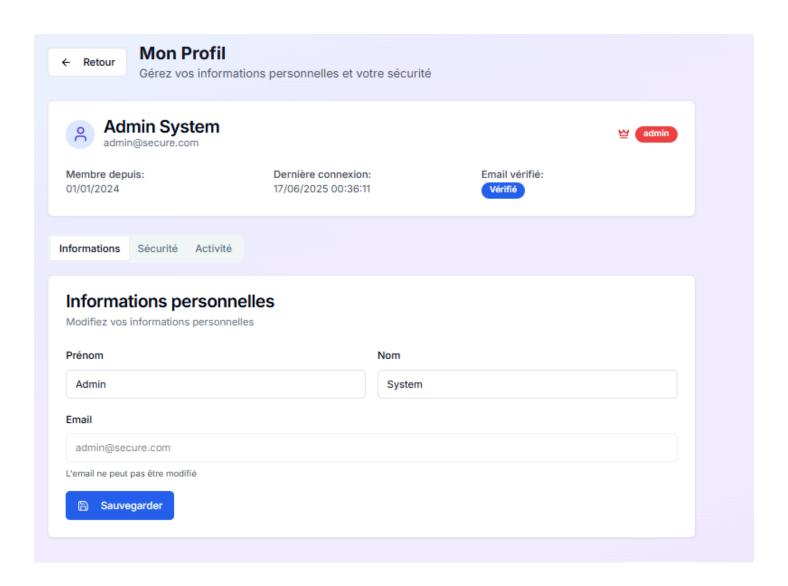
PAGE DE CONNEXION:



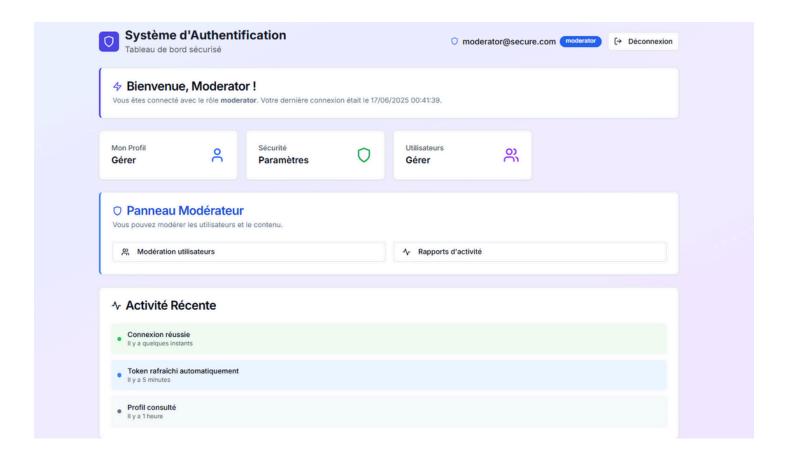
<u>"L'ADMIN:</u>



Le Profil de l'admin:



Je Modérateur:



L'Utilisateur:

