PARTIE 2 : FONCTIONNALITÉS AVANCÉES

Formateur : Makhmadane LO

Lead Senior Développeur / Cloud / Ops

OBJECTIFS DU PROJET

À la fin de ce projet, vous serez capable de :

- Implémenter l'authentification JWT avec Spring Security
- Intégrer l'envoi automatique d'emails avec Spring Mail et Thymeleaf
- Gérer l'upload et le téléchargement de fichiers avec MultipartFile
- Appliquer les bonnes pratiques de sécurité backend

CONTEXTE

SecureLife souhaite enrichir sa plateforme avec trois fonctionnalités essentielles pour améliorer la sécurité, la communication et la gestion documentaire de son système d'assurance.

MODULE 1 : ENVOI D'EMAILS AUTOMATIQUES

Objectif

Automatiser la communication avec les clients par email lors d'événements importants du cycle de vie des contrats.

Fonctionnalités requises

Le système doit envoyer automatiquement des emails lors de :

1. **Création de contrat :** Email de bienvenue avec récapitulatif complet
2. **Modification de contrat :** Notification des changements effectués
3. **Changement de statut :** Alerte en cas de suspension, annulation ou réactivation
4. **Upload de document :** Confirmation de réception du document

MODULE 2 : GESTION DE DOCUMENTS

Objectif

Permettre aux clients de joindre des documents à leurs contrats pour compléter leur dossier d'assurance.

Types de documents

Type	Description
IDENTITY_CARD	Carte d'identité nationale
PASSPORT	Passeport
DRIVING_LICENSE	Permis de conduire

Type	Description
VEHICLE_REGISTRATION	Carte grise du véhicule
PROOF_OF_RESIDENCE	Justificatif de domicile
OTHER	Autres documents justificatifs

Endpoints API

Méthode	Endpoint	Description
POST	/api/v1/insurances/{id}/documents	Upload document
GET	/api/v1/insurances/{id}/documents	Lister documents
GET	/api/v1/documents/{id}	Télécharger
DELETE	/api/v1/documents/{id}	Supprimer

Règles de validation

- **Formats autorisés :** PDF, JPG, JPEG, PNG uniquement
- **Taille maximale :** 5 MB par fichier
- **Nombre maximum :** 10 documents par contrat
- Génération nom unique avec UUID
- Organisation stockage par date : uploads/2025/01/15/
- Streaming pour téléchargement (ne pas charger en mémoire)

MODULE 3 : SÉCURITÉ JWT

Objectif

Sécuriser l'API pour que seuls les utilisateurs authentifiés puissent accéder aux contrats, avec différents niveaux d'autorisation selon le rôle.

Rôles et permissions

Rôle	Permissions
ADMIN	Accès total : CRUD tous contrats, statistiques, gestion users
AGENT	Créer, modifier, lire contrats (pas de suppression)
CLIENT	Lire uniquement SES propres contrats

Endpoints d'authentification

Méthode	Endpoint	Description
POST	/api/v1/auth/register	Inscription (Public)

Méthode	Endpoint	Description
POST	/api/v1/auth/login	Connexion (Public)
POST	/api/v1/auth/refresh	Refresh token (Public)
GET	/api/v1/auth/me	Profil utilisateur

Technologies utilisées

Technologie	Version	Usage
Spring Security	6.x	Framework sécurité
JWT (jjwt)	0.12.x	Tokens JWT
BCrypt	Inclus	Hash passwords

Configuration des tokens

- **Access Token :** 15 minutes de validité
- **Refresh Token :** 7 jours de validité
- **Clé secrète :** Minimum 256 bits (générer avec openssl rand -base64 32)
- **Signature :** HMAC-SHA256

Points clés

- Hasher les passwords avec BCrypt (force 12)
- Variables d'environnement pour les secrets
- Valider tokens à chaque requête
- Vérifier autorisations selon rôle
- Protection ownership : CLIENT ne peut voir que ses contrats

Note importante

Cette Partie 2 s'ajoute à la Partie 1. Votre projet final doit inclure toutes les fonctionnalités des deux parties.

Bon courage ! 🚀

Date de remise : À définir par le formateur
Format de remise : Lien GitHub