# Contexte du projet

Le but du projet est de **gérer l'accès aux documents** sur un **serveur local** en utilisant une solution de **GED (Gestion Électronique des Documents)**. L'application repose sur une **application Java Spring Boot** déjà existante, et l'objectif est de créer une **API** pour intégrer et sécuriser l'accès aux documents.

# Mots-clés et concepts clés

1. **Gestion des accès aux documents** :
   * Contrôle des permissions (lecture, écriture, suppression).
   * Sécurisation des documents stockés sur le serveur local.
2. **Authentification** :
   * Utilisation d'**Active Directory (AD)** et **LDAP** pour l'authentification des utilisateurs.
   * Intégration avec le système d'authentification existant.
3. **Alfresco** :
   * Solution de **GED** utilisée pour la gestion des documents.
   * Alfresco permet de stocker, organiser et gérer les documents de manière centralisée.
4. **Application Java Spring Boot existante** :
   * L'application est déjà développée, mais nécessite une **intégration** avec Alfresco et l'API à créer.
   * Spring Boot facilite la création d'API RESTful.
5. **Création d'une API** :
   * Développement d'une interface de programmation pour permettre l'interaction avec la GED.
   * L'API servira de pont entre l'application Spring Boot et Alfresco.
6. **Boîte noire** :
   * L'application existante est considérée comme une **boîte noire** (son fonctionnement interne n'est pas modifié).
   * L'objectif est de l'intégrer sans altérer son code source.
7. **GED (Gestion Électronique des Documents)** :
   * Solution pour stocker, organiser et gérer les documents de manière structurée.
   * Alfresco est une GED open source largement utilisée.
8. **Intégration** :
   * Intégration de l'application Spring Boot avec Alfresco via l'API.
   * Synchronisation des utilisateurs et des permissions via Active Directory/LDAP.

# Étapes du projet

1. **Analyse des besoins** :
   * Identifier les fonctionnalités nécessaires pour la gestion des accès.
   * Définir les rôles et permissions des utilisateurs.
2. **Configuration d'Alfresco** :
   * Installer et configurer Alfresco sur le serveur local.
   * Intégrer Alfresco avec Active Directory/LDAP pour l'authentification.
3. **Développement de l'API** :
   * Créer une API RESTful en Java Spring Boot pour interagir avec Alfresco.
   * Implémenter les Endpoint nécessaires (ex : uploade, téléchargement, gestion des permissions).
4. **Intégration avec l'application existante** :
   * Utiliser l'API pour connecter l'application Spring Boot à Alfresco.
   * S'assurer que l'application respecte les permissions définies dans Alfresco.
5. **Tests et validation** :
   * Tester l'authentification via Active Directory/LDAP.
   * Vérifier que les accès aux documents sont correctement gérés.
6. **Documentation** :
   * Documenter l'API et les étapes d'intégration.
   * Fournir un guide d'utilisation pour les administrateurs et les utilisateurs.

# Technologies et outils

* **Java Spring Boot** : Framework pour développer l'API.
* **Alfresco** : Solution de GED pour la gestion des documents.
* **Active Directory/LDAP** : Authentification centralisée des utilisateurs.
* **RESTful API** : Interface de communication entre l'application et Alfresco.
* **Serveur local** : Hébergement des documents et des services.

**Résultat attendu**

Une application Java Spring Boot intégrée à Alfresco, permettant de gérer l'accès aux documents sur un serveur local via une API sécurisée. Les utilisateurs s'authentifient via Active Directory/LDAP, et les permissions sont gérées par Alfresco.

# Guide pour débutants : Intégration Spring Boot et Alfresco

## Étape 1 : Préparation et mise en place de l'environnement

1. **Installer les outils de base** :
   * Java JDK (version 11 ou supérieure)
   * Maven ou Gradle (pour gérer les dépendances)
   * IDE (IntelliJ IDEA, Eclipse ou VS Code)
   * Git (pour le contrôle de version)
2. **Comprendre les concepts de base** :
   * Prendre le temps de comprendre les principes fondamentaux de Spring Boot
   * Se familiariser avec les concepts d'API REST
   * Comprendre ce qu'est Alfresco et son rôle dans la gestion documentaire

## Étape 2 : Installation et configuration d'Alfresco

1. **Télécharger Alfresco Community Edition** :
   * Aller sur le site d'Alfresco et télécharger la version Community
   * Suivre les instructions d'installation pour votre système d'exploitation
2. **Configuration de base d’Alfresco :**
   * Démarrer Alfresco et vérifier que vous pouvez accéder à l'interface web (habituellement <http://localhost:8080/share>)
   * Créer un compte administrateur si ce n'est pas déjà fait
3. **Explorer l'interface d’Alfresco :**
   * Se familiariser avec la création de sites, dossiers et documents
   * Comprendre comment les permissions sont gérées

## Étape 3 : Créer le projet Spring Boot

1. **Initialiser un nouveau projet Spring Boot** :
   * Utiliser Spring Initializr (<https://start.spring.io/>)
   * Sélectionner les dépendances : Web, Security, LDAP
2. **Ajouter les dépendances Alfresco** :
   * Ajouter les dépendances pour l'API CMIS dans votre fichier pom.xml :

**<dependency>**

**<groupId>org.apache. chemistry. opencmis</groupId>**

**<artifactId>chemistry-opencmis-client-impl</artifactId>**

**<Version>1.1.0</version>**

**</dependency>**

1. **Structurer le projet** :
   * Créer les package pour les modèles, services, contrôleurs, etc.
   * Configurer application. Properties avec les paramètres de base

## Étape 4 : Implémenter les modèles et services de base

1. **Créer la classe Document** :
   * Implémenter la classe Document avec ses attributs de base
   * Ajouter les getters et setters
2. **Créer l'interface DocumentService** :
   * Définir les méthodes de base (upload, download, List, delete)
3. **Implémenter AlfrescoDocumentService** :
   * Commencer par les fonctionnalités de base : récupérer la session CMIS
   * Implémenter une méthode à la fois, en testant chacune avant de passer à la suivante

## Étape 5 : Configuration de l'authentification

1. **Configurer la connexion LDAP/AD** :
   * Mettre en place la configuration de sécurité dans un fichier de configuration séparé
   * Commencer avec une configuration simple, sans AD pour débuter
2. **Tester l’authentification :**
   * Vérifier que vous pouvez vous authentifier avec des utilisateurs locaux
3. **Intégrer avec Active Directory** (version simplifiée pour commencer) :
   * Configurer les paramètres de base pour se connecter à un AD de test

## Étape 6 : Développer l'API REST

1. **Créer le contrôleur DocumentController** :
   * Implémenter les endpoints de base un par un
   * Commencer par la méthode pour lister les documents, puis celle pour télécharger
2. **Tester les endpoints** :
   * Utiliser Postman ou un autre outil pour tester chaque endpoint
   * Vérifier les réponses HTTP et le format des données

## Étape 7 : Intégrer avec l'application existante

1. **Comprendre l'application existante** :
   * Examiner comment l'application gère actuellement les documents
   * Identifier les points d'intégration
2. **Développer un client simple** :
   * Créer un client qui utilise votre API
   * Tester l'intégration avec des cas simples

## Étape 8 : Tests et validation

1. **Écrire des tests unitaires** :
   * Tester les services et contrôleurs individuellement
   * Utiliser des mocks pour simuler Alfresco
2. **Réaliser des tests d’intégration :**
   * Vérifier que l'ensemble du système fonctionne correctement
   * Tester différents scénarios d'utilisation

## Étape 9 : Amélioration progressive

1. **Ajouter la gestion des permissions** :
   * Implémenter les méthodes pour gérer les permissions sur les documents
   * Tester avec différents rôles et utilisateurs
2. **Optimiser les performances** :
   * Ajouter la mise en cache si nécessaire
   * Améliorer les requêtes pour une meilleure efficacité

## Étape 10 : Documentation et déploiement

1. **Documenter l’API :**
   * Utiliser Swagger/OpenAPI pour documenter les endpoints
   * Créer un guide d'utilisation simple
2. **Préparer le déploiement** :
   * Configurer l'application pour un environnement de production
   * Mettre en place les variables d'environnement

## 1. Comprendre les bases des technologies utilisées

Avant de commencer, familiarisez-vous avec les concepts et technologies clés :

* **Java Spring Boot** : Apprenez à créer une application de base avec Spring Boot (controllers, services, repositories).
* **API REST** : Comprenez ce qu'est une API REST, comment elle fonctionne et comment en créer une avec Spring Boot.
* **Alfresco** : Explorez les fonctionnalités de base d'Alfresco (stockage de documents, gestion des utilisateurs, permissions).
* **Active Directory/LDAP** : Apprenez comment LDAP fonctionne et comment il est utilisé pour l'authentification.
* **GED (Gestion Électronique des Documents)** : Comprenez les concepts de base de la gestion des documents.

Ressources utiles :

* Tutoriels Spring Boot sur [Spring.io](https://spring.io/guides).
* Documentation Alfresco : [Alfresco Documentation](https://docs.alfresco.com/).
* Tutoriels LDAP : Par exemple, [LDAP for Beginners](https://www.tutorialspoint.com/ldap/index.htm).

## 2. Configurer l'environnement de développement

* Installez les outils nécessaires :
  + **Java JDK** (version compatible avec Spring Boot).
  + **IDE** (IntelliJ IDEA, Eclipse, ou VS Code).
  + **Postman** (pour tester les API).
  + **Docker** (optionnel, pour simplifier l'installation d'Alfresco).
* Configurez un **serveur local** (ou utilisez votre machine pour les tests).

## 3. Installer et configurer Alfresco

* Téléchargez et installez Alfresco sur votre serveur local.
  + Vous pouvez utiliser Docker pour simplifier l'installation : [Alfresco Docker](https://hub.docker.com/r/alfresco/alfresco-content-repository/).
* Configurez Alfresco pour qu'il utilise **Active Directory/LDAP** pour l'authentification :
  + Ajoutez les utilisateurs et groupes dans Active Directory.
  + Configurez Alfresco pour se connecter à LDAP (voir la documentation Alfresco).

## 4. Créer une application Spring Boot de base

* Utilisez [Spring Initializr](https://start.spring.io/) pour générer un projet Spring Boot.
* Ajoutez les dépendances nécessaires :
  + **Spring Web** (pour créer l'API REST).
  + **Spring Security** (pour gérer l'authentification).
  + **Spring Data JPA** (optionnel, si vous avez besoin d'une base de données).
* Créez un contrôleur de base pour tester l'API (exemple : un endpoint GET /hello).

## 5. Intégrer Alfresco avec Spring Boot

* Utilisez l'**API REST d'Alfresco** pour interagir avec le système de gestion des documents.
  + Consultez la documentation de l'API Alfresco : [Alfresco REST API](https://api-explorer.alfresco.com/).
* Créez des endpoints dans votre application Spring Boot pour :
  + **Uploader des documents** vers Alfresco.
  + **Télécharger des documents** depuis Alfresco.
  + **Gérer les permissions** des utilisateurs sur les documents.

Exemple de code pour uploader un document :

**@RestController**

**public class DocumentController {**

**@PostMapping("/upload")**

**public ResponseEntity<String> uploadDocument(@RequestParam("file") MultipartFile file) { // Implémentez la logique pour uploader le fichier vers Alfresco**

**return ResponseEntity.ok("Document uploadé avec succès !"); }**

**}**

## 6. Implémenter l'authentification avec Active Directory/LDAP

* Configurez **Spring Security** pour utiliser LDAP :
  + Ajoutez la dépendance spring-boot-starter-security et spring-ldap-core.
  + Configurez les paramètres LDAP dans application.properties.
* Testez l'authentification avec des utilisateurs Active Directory.

Exemple de configuration LDAP dans application.properties :

**spring.ldap.urls=ldap://localhost:389**

**spring.ldap.base=dc=example,dc=com**

**spring.ldap.username=cn=admin,dc=example,dc=com**

**spring.ldap.password=adminpassword**

## 7. Tester et valider l'intégration

* Utilisez **Postman** pour tester les endpoints de l'API :
  + Vérifiez que les documents sont correctement uploadés et téléchargés.
  + Testez l'authentification avec différents utilisateurs LDAP.
* Assurez-vous que les permissions sont correctement appliquées dans Alfresco.

## 8. Documenter le projet

* Créez une documentation pour :
  + Les endpoints de l'API (exemple : utiliser **Swagger**).
  + Les étapes d'installation et de configuration.
  + Les instructions pour les utilisateurs finaux.

## 9. Déployer l'application

* Déployez l'application Spring Boot sur un serveur (exemple : Tomcat, Docker).
* Assurez-vous qu'Alfresco et l'API sont accessibles sur le réseau local.

## 10. Améliorer et maintenir

* Ajoutez des fonctionnalités supplémentaires (exemple : recherche de documents, notifications).
* Surveillez les performances et corrigez les bugs.