Pour gérer les permissions dans Alfresco tout en intégrant Active Directory (AD), voici quelques approches possibles :

**1. Utilisation directe des groupes AD dans Alfresco**

**Principe** :

* Synchroniser les groupes et utilisateurs d’AD avec Alfresco.
* Mapper les groupes AD avec des groupes Alfresco.
* Gérer les permissions via Alfresco en attribuant des droits aux groupes synchronisés.

**Avantages** :  
✅ Facilité d’administration (les droits sont gérés depuis AD).  
✅ Pas besoin de créer une logique de permission spécifique dans l’application.  
✅ Cohérence avec la gestion des accès existante dans l’entreprise.

**Inconvénients** :  
❌ Moins de flexibilité si les rôles Alfresco doivent être différents de ceux définis dans AD.  
❌ Nécessite une bonne gestion des groupes côté AD.

**2. Gestion hybride : Groupes AD + Rôles spécifiques dans Alfresco**

**Principe** :

* Synchroniser les utilisateurs d’AD avec Alfresco, mais pas forcément tous les groupes.
* Gérer les rôles spécifiques à Alfresco directement dans Alfresco.
* Associer manuellement les utilisateurs aux rôles définis dans Alfresco (ex. : Administrateur, Éditeur, Lecteur).

**Avantages** :  
✅ Plus de flexibilité pour gérer les permissions dans Alfresco.  
✅ Possibilité de créer des rôles spécifiques sans dépendre totalement d’AD.

**Inconvénients** :  
❌ Besoin d’une double gestion des utilisateurs (AD + Alfresco). ❌ Plus de configuration et de maintenance.

**3. Gestion des permissions côté API Spring Boot**

**Principe** :

* Authentification via AD/LDAP.
* L’API Spring Boot interroge AD pour récupérer les rôles.
* L’API applique des règles de gestion et ne transmet à Alfresco que les actions autorisées.

**Avantages** :  
✅ Contrôle total sur la logique d’accès.  
✅ Possibilité d’ajouter des règles métiers spécifiques.  
✅ Moins de dépendance sur la configuration Alfresco.

**Inconvénients** :  
❌ Plus complexe à développer. ❌ Peut générer des problèmes de performance si mal optimisé.

La **première option : utiliser les groupes AD directement dans Alfresco**. C'est la plus simple à mettre en place et elle évite d’écrire trop de logique métier dans ton API.

# Étapes pour débuter :

## 1. Configurer Alfresco pour se connecter à Active Directory (AD/LDAP)

Alfresco permet de synchroniser les utilisateurs et les groupes depuis AD. Tu devras configurer le fichier alfresco-global.properties avec les paramètres de ton serveur AD.

Exemple de configuration :

**ldap.synchronization.active=true**

**ldap.authentication.active=true**

**ldap.authentication.userNameFormat=%s@domaine.local**

**ldap.synchronization.java.naming.security.principal=admin@domaine.local**

**ldap.synchronization.java.naming.security.credentials=motdepasse**

**ldap.synchronization.groupSearchBase=ou=Groups,dc=domaine,dc=local**

**ldap.synchronization.userSearchBase=ou=Users,dc=domaine,dc=local**

➡️ Cette configuration permet d’authentifier les utilisateurs via AD et de synchroniser les groupes d’AD avec Alfresco.

## 2. Vérifier la synchronisation des utilisateurs et groupes

Une fois configuré, Alfresco synchronisera les groupes et utilisateurs d’AD.  
Vérifie dans **l’interface Alfresco Share** que tes utilisateurs AD apparaissent bien.

Commande pour tester la synchronisation :

java -jar alfresco-mmt.jar list-repo

Ou bien, regarde dans la base de données Alfresco (table alf\_authority).

## 3. Attribuer des permissions aux groupes AD dans Alfresco

* Va dans **l’interface d’administration Alfresco**.
* Dans la section **Permissions**, associe tes groupes AD à des rôles Alfresco (Lecteur, Éditeur, Contributeur...).
* Exemple : Groupe\_AD\_Commercial → **Lecteur** sur le dossier "Documents\_Clients".

## 4. Développer ton API Spring Boot pour interagir avec Alfresco

Ton API Spring Boot pourra utiliser l’authentification AD automatiquement (puisque Alfresco gère déjà l’accès).

Exemple de récupération d’un document via Alfresco API :

**RestTemplate restTemplate = new RestTemplate ();**

**HttpHeaders headers = new HttpHeaders();**

**headers.setBasicAuth("userAD", "passwordAD"); // Utilisateur AD**

**HttpEntity<String> entity = new HttpEntity<>(headers);**

**ResponseEntity<String> response = restTemplate.exchange(**

**"http://localhost:8080/alfresco/api/-default-/public/alfresco/versions/1/nodes/{nodeId}/content",**

**HttpMethod.GET,**

**entity,**

**String.class,**

**nodeId**

**);**

➡️ Cette requête permet de télécharger un document stocké dans Alfresco.

**Pourquoi cette approche est adaptée pour un débutant ?**

✅ **Peu de développement** : Alfresco gère déjà les permissions.  
✅ **Moins de maintenance** : Les groupes sont définis dans AD, donc pas besoin de tout gérer dans ton application.  
✅ **Facile à tester** : Tu peux voir directement dans Alfresco si tes groupes sont bien synchronisés.