

TP 3 : Application de gestion de documents – Upload de fichiers

Description :

Dans ce TP, vous allez créer une application Spring Boot permettant :

- De **créer**, **lister**, **télécharger**, et **supprimer** des fichiers/documents.
- De **stocker** les fichiers dans un dossier local sur votre machine.
- De **garder** les informations des documents (nom, type, taille, chemin) dans une base de données H2.

Objectifs :

- Comprendre l'upload de fichiers avec Spring Boot (MultipartFile).
- Sauvegarder des fichiers dans le système de fichiers local.
- Enregistrer des métadonnées (nom du fichier, chemin, type MIME) dans une base de données.
- Exposer des API REST pour manipuler les fichiers.

Partie I : Création de l'application

- 1) Démarrez un projet Spring Boot en utilisant Spring Initializer (ou un outil similaire) en ajoutant les dépendances suivantes : Spring Data JPA, H2 Database, Lombok, Spring Web.
- 2) Créer une classe modèle « **Document** » pour représenter un document avec des attributs tels que :
 - L'ID : (Long)
 - Le nom du fichier: (String)
 - Le type du fichier : (String)
 - La taille du fichier: (Long)
 - Le chemin du fichier : (String)

- 3) Créez un package pour les repositories (couche DAO) où vous définirez l'interface `DocumentRepository`. Cette interface étend `JpaRepository` de Spring Data JPA et vous permettra d'accéder à la base de données.
- 4) Créez un package pour les services (couche métier) où vous définirez la classe `DocumentService`. Cette classe contient les méthodes suivantes :
 - **`uploadDocument(MultipartFile file)`** : Sauvegarde le fichier sur disque et enregistre ses métadonnées.
 - **`getAllDocuments()`** : Récupère la liste de tous les documents.
 - **`downloadDocument(Long id)`** : Télécharge un document par son ID.
 - **`deleteDocument(Long id)`** : Supprime un document (fichier + enregistrement en base).

Partie II : Création d'un contrôleur et la configuration de la base de données H2 en mémoire :

- 1) Mettre en place un contrôleur Spring Boot `DocumentController` pour gérer les opérations CRUD pour les documents
- 2) Créer des endpoints RESTful pour les opérations CRUD, en utilisant les méthodes HTTP appropriées (GET, POST, PUT, DELETE).
 - Endpoint qui permet d'uploader un fichier (POST)
 - Endpoint pour récupérer tous les documents (GET)
 - Endpoint pour récupérer un document par ID (GET)
 - Endpoint pour mettre à jour un document (PUT)
 - Endpoint pour supprimer un document (DELETE)
- 3) Sur `application.properties` ajouter la configuration de la base de données.
 - a. Les fichiers doivent être stockés dans un dossier local par exemple `uploads/`.
 - b. Le dossier doit être créé automatiquement s'il n'existe pas.