

TP : Crédation d'une API REST avec Spring Boot (Modèle MVC)

Objectifs pédagogiques

- Comprendre et appliquer l'architecture MVC avec Spring Boot.
- Créeer une API RESTful complète avec Java et Spring Boot.
- Manipuler les couches Controller, Service et Repository.
- Utiliser JPA/Hibernate pour la persistance des données.
- Implémenter la validation, la gestion d'erreurs et la documentation.

Sujet : Gestion d'une Bibliothèque

Vous devez développer une API REST permettant de gérer des livres et des auteurs.

1 Entités à gérer

Author : id, firstName, lastName, birthYear

Book : id, title, isbn (unique), year, category (enum : NOVEL, ESSAY, POETRY, OTHER), author (relation ManyToOne)

2 Fonctionnalités attendues

Auteurs

GET /authors → liste des auteurs

GET /authors/{id} → auteur par id

POST /authors → ajouter un auteur

PUT /authors/{id} → modifier un auteur

DELETE /authors/{id} → supprimer un auteur

Livres

GET /books → liste paginée des livres avec filtres :
title, authorId, category, yearFrom, yearTo, tri possible (sort=year,desc)

GET /books/{id} → détail d'un livre

POST /books → créer un livre (validation obligatoire)

PUT /books/{id} → modifier un livre

DELETE /books/{id} → supprimer un livre

Statistiques

GET /stats/books-per-category → nombre de livres par catégorie

GET /stats/top-authors?limit=3 → auteurs ayant le plus de livres

3 Contraintes et règles

isbn doit être unique.

Validation des champs (title non vide, isbn format correct, year entre 1450 et l'année actuelle).

Si un auteur n'existe pas → erreur 404.

Les erreurs doivent être retournées en JSON structuré.

Les requêtes POST/PUT/PATCH nécessitent une clé API (X-API-KEY).

L'accès en lecture (GET) reste public.

4 Structure du projet attendue

Respectez l'architecture MVC :

controller/ → classes REST (API)

service/ → logique métier

repository/ → accès base de données (Spring Data)

domain/ → entités JPA

dto/ → objets de transfert et validation

exception/ → gestion des erreurs

config/ → configuration, sécurité, OpenAPI

5 Bonus possibles

Endpoint PATCH pour modification partielle

Authentification avec Spring Security (au lieu de clé API)

Cache sur les statistiques

Export CSV des livres

Tests unitaires et d'intégration avec MockMvc

6 Rendu

Ce que vous devez livrer

Le code complet du projet (Git ou archive ZIP)

Un fichier README.md expliquant :

Comment lancer l'application

Liste des endpoints disponibles

Les filtres et exemples de requêtes

Un export Postman ou Insomnia pour tester les routes

Projet à rendre sur GitHub (lien du repo) ou archive .zip nommée :

TP_SpringBoot_NomPrenom.zip

Présentation/démonstration en séance : 5–10 minutes par groupe.