

TP Final : Spring BOOT

Partie 1

Objectif

Le but de ce TP est de réaliser une petite application **Planning** permettant la gestion des salles de cours, des cours et des élèves, en appliquant une **architecture REST** en MVC. L'application aura un front développé en HTML/CSS/Javascript à l'aide de Spring Thymeleaf. Il est autorisé d'utiliser une autre technologie front si vous le souhaitez, dans ce cas, l'application "back Office" servira des **point d'accès REST** en **JSON**.

Pour initialiser votre projet, vous pouvez vous rendre sur Spring Initializr à l'adresse suivante : <https://start.spring.io> Vous ajouterez dans votre projet les dépendances **Spring Web**, **Spring DATA JPA** et **H2 Database** (la base de données utilisée sera en mémoire pour des raisons de simplicité) et **Thymeleaf**.

Vous devrez prévoir un ou plusieurs endpoints pour initialiser un ensemble d'élèves, les niveaux scolaires et de salles dans votre application.

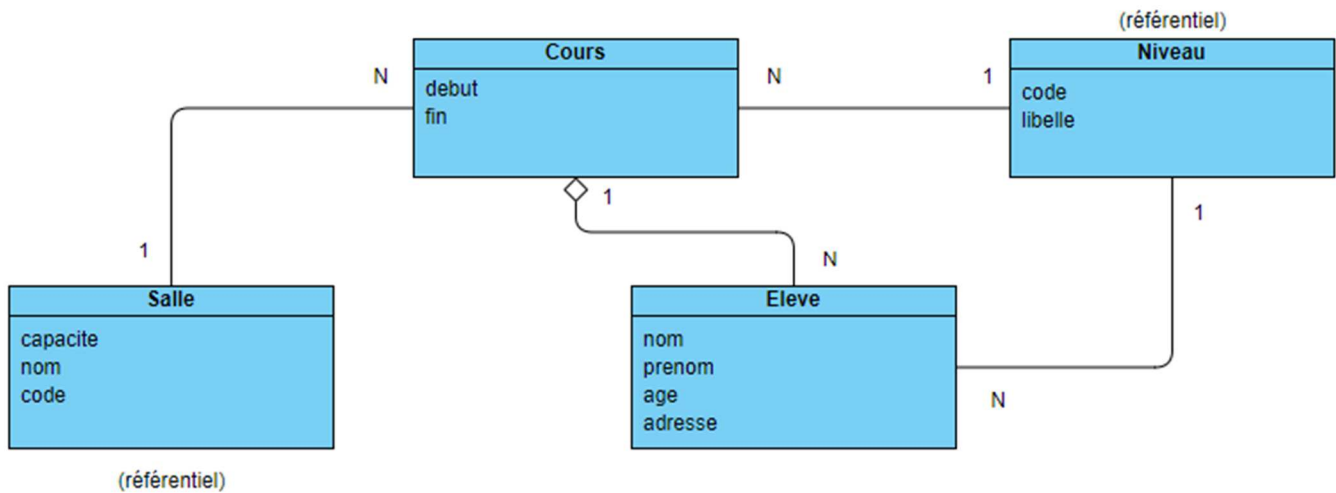
L'application doit proposer plusieurs pages et cas d'utilisation dont :

- La consultation des salles
- La création d'une nouvelle salle
- La modification d'une salle existante
- La suppression d'une salle existante
- La consultation des élèves
- La création d'un nouvel élève
- La modification d'un élève existant
- La suppression d'un élève
- La consultation des cours
- La création d'un cours
- La modification d'un cours
- La suppression d'un cours

Pour les pages de consultation, on doit pouvoir trier sur les colonnes, et au mieux, filtrer sur un ou plusieurs champs.

Certaines des actions seront régies par des contraintes ou des règles :

- Un cours ne peut être mis en place que si la salle n'est pas déjà utilisée sur les horaires
- Un élève ne peut être ajouté dans un cours que si la capacité de la salle n'est pas atteinte, et que l'élève ne suit pas un autre cours sur les mêmes horaires
- La suppression d'une salle ne peut se faire que si aucun cours ne l'utilise



Modèle

Le domaine **Elève** a un **nom**, un **prénom** et un **âge**. Il a également un **niveau scolaire** qui doit être un référentiel. Une salle de cours a une **capacité**, un **nom** et un **code** (F1, F2, F3...). Un **Cours** a une date de **début**, une date de **fin**, un **niveau scolaire**, une **salle** et des **élèves**. Un **niveau scolaire** a un code et un libellé.

Important : Tous les endpoints doivent respecter l'architecture **REST**.

Vous pourrez prévoir un affichage de type planning/emploi du temps pour la consultation des cours.

Ce TP reprend toutes les notions vues dans le cours et les TP précédents. Il est à rendre à la fin sous forme d'une archive, contenant les **sources** et **votre livrable .jar**.