

TP sur SPRING BOOT-Maven

Jeudi 30/09/2021

Vous allez créer une API Rest (Crud Spring Boot).

Les API Rest permettent d'établir une communication entre un serveur backend et un frontend web ou des applications mobiles.

- **ODK** veut numériser son système de pointage , pour cela il demande votre expertise pour mettre en place une application web. Chaque apprenant doit avoir un compte sur l'application.
A partir de l'application,l' administrateur doit pouvoir gérer les apprenants,en faisant l'ajout,modification,suppression,listing des apprenants. Il doit pouvoir aussi nous fournir la liste de présence avec (date, heure_arrive_apprenant, nom_apprenant, prenom_apprenant) à partir de l'application. En tenant compte que nous travaillons sous mode **Agile**,le système peut évoluer au fur du temps avec plus de fonctionnalités.

Faire une analyse de votre système en créant:

- les cas utilisations,
- diagramme de classe
- diagramme de séquence

Le nom de la base de données sera **"odk_db."**

Créer un projet de spring boot avec maven **"OdkApprenant "**.

Utiliser le fichier **"application.properties"** pour connecter votre projet avec à votre base de données.

- Commencer par créer un package `.model` à partir de votre package racine : **com.OdkApprenant.demo**
 - Dans le package `.model` créé ci-dessus, créez une énumération Java avec le nom `ApprenantStatus` avec les éléments: `activer`, `désactiver`.

- Dans le package `.model` créé ci-dessus, créez une classe Java avec le nom `Apprenant` , qui va persister dans la base de données ayant comme attributs (id, nom, prenom, age, telephone, email, login, password, genre, ApprenantStatus, dateCreation, dateModification);
- Créez à partir du package racine de votre projet, un package avec le nom `repositories`. Dans le `repositories` package, créez une interface avec le nom `ApprenantRepository` qui étend le `JpaRepository` interface et est livrée avec les fonctions CRUD déjà implémentées,
- A partir du package racine de votre application, créez un package avec le nom `services`. Dans `services` package, créez une interface avec le nom `ApprenantService`.
- Dans `services` package, créez une classe service avec le nom `ApprenantServiceImp` qui implémente `ApprenantService` interface que vous avez créé.
- Faire une injection de `ApprenantRepository` dans la classe `ApprenantServiceImp` pour pouvoir utiliser les différentes méthodes fournies par **`JpaRepository`**.
- A partir du package racine de votre projet, créez un package avec le nom `controllers`. Dans le `controllers` package, créez une classe Java controller avec le nom `ApprenantController` qui va faire appel à `ApprenantService` pour mettre en place l'ensemble des services ci-dessous:

- Création d'un nouvel apprenant
- Modification d'un nouvel apprenant
- Afficher la liste des apprenants
- Afficher les informations d'un apprenant par ID
- Supprimer un apprenant

A FAIRE:

Testez les points de terminaison CRUD avec Postman :

- ☐ Création d'un nouvel apprenant
- ☐ Modification d'un nouvel apprenant
- ☐ Afficher la liste des apprenants
- ☐ Afficher les informations d'un apprenant par ID
- ☐ Supprimer un apprenant

NB: à faire en binôme, la restitution est pour lundi 04/10/2021 à partir de 14h pour 20min de présentation

Mettez le fichier de votre base de données dans le dossier de votre projet Spring et faire un dépôt sur git en envoyant le lien sur :
orangekalanso@gmail.com