

TD Framework

Rendu de l'exercice :

Le livrable attendu est le code source du mini-projet ainsi qu'une collection Postman permettant de tester votre code.

Le rendu de l'exercice se fera sous git. Vous pouvez le faire via Framagit, Github, Bitbucket, GitLab ou autre. Vous pouvez faire un repository privé ou public ; dans le cas d'un repository privé, ajoutez-moi comme membre de l'équipe avec l'adresse : q.leuly@iut.univ-paris8.fr

Vous créerez un README.md à la racine du projet dans lequel vous indiquerez :

- les nom et prénom de votre groupe
- les différents endpoints de votre projet et à quelles méthodes HTTP ils sont associés
- le chemin où trouver votre fichier d'export Postman
- comment configurez la base de données pour faire fonctionner votre projet :
 - s'agit-il d'une base h2 ? Ou Postgres ou MySQL ?
 - Spring est-il responsable de la création de votre schéma ou avez-vous un script de création ?

Sujet du mini-projet :

Repartez de la dernière version de votre code.

Rappel des fonctionnalités que votre projet doit avoir à présent:

- **CRUD pour les employés** (avec protection : un employé x n'a pas accès à aux informations d'un employé y)
- **Permettre de poser des congés pour un employé** (avec protection : on ne peut pas poser des congés pour une autre employé que celui connecté)
- **Calcul du salaire d'un employé** avec prise en compte des congés pris sur le mois pour lequel on calcule le salaire
 - on ne peut pas calculer 2 fois le salaire d'un même mois
- **Inscription d'un utilisateur et rattachement (ou non) d'un utilisateur à un employé** déjà existant
- **Protection des méthodes de lecture des informations d'un utilisateur et d'un employé** : limité à l'employé/utilisateur connecté si celui-ci correspond à l'employé/utilisateur que l'on souhaite lire

Si des fonctionnalités sont manquantes, développez-les.

Nouvelles fonctionnalités à développer :

- ajoutez une fonctionnalité permettant de calculer le salaire d'un employé sans l'enregistrer en base (une sorte d'aperçu) : ne pas rajouter de nouvel endpoint, rajoutez simplement un paramètre optionnel au endpoint de calcul des salaires

- si ce n'est pas fait, faire un CRUD (enfin, juste la partie manquante) sur les jours de congés d'un employé. Sécurisez les endpoints afin que seul l'utilisateur rattaché à l'employé puisse poser des jours de congés
- développez des méthodes permettant à un utilisateur de changer de mot de passe : pour changer de mot de passe, on a besoin de l'ancien mot de passe, du nouveau mot de passe et d'une confirmation du nouveau mot de passe.
- autoriser la création d'un utilisateur et d'un employé en même temps (du coup, l'utilisateur est associé à l'employé automatiquement)
- rajoutez une fonctionnalité permettant de trier en fonction de la date (croissant et décroissant) la liste des jours de congés d'un employé sans rajoute de nouveau endpoint. Faites pareil pour la liste des employés (en fonction du nom et du prénom : SMITH John doit être devant SMITH Roger dans un tri par ordre croissant).

N'oubliez pas les tests ! Faites des tests unitaires (et des tests d'intégration si le cours a assez avancé, si vous savez déjà en faire ou si vous êtes curieux). Vous n'avez pas besoin de tester forcément 100% de votre code, choisissez bien les éléments les plus pertinents à tester.