**Rectifications et Améliorations du projet « Gestion de Personnel »**

1. **Mécanisme de recherche par nom :**

J’ai mis en place un système de recherche dans Spring Data JPA qui permet de trouver des employés par leur nom avec une méthode **findByNameContainingIgnoreCase**.

Spring s’occupe de tout : il cherche les noms qui contiennent le texte donné, peu importe les majuscules ou minuscules.

**Modifications :**

Dans **EmployeeController.java**

@GetMapping("/search")

    public List<Employee> searchEmployees(@RequestParam String name) {

        return employeeService.searchEmployeesByName(name);

    }

Dans **EmployeeRepository.java**

List<Employee> findByNameContainingIgnoreCase(String name);

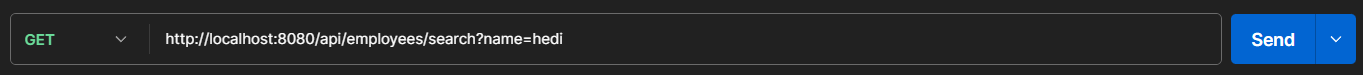
Dans **EmployeeService.java**

public List<Employee> searchEmployeesByName(String name) {

        return employeeRepository.findByNameContainingIgnoreCase(name);

    }

**Exemple d’usage (dans Postman ou un Browser):**

****

**Retourne la liste des employés ayant un nom qui contient « hedi »:**

[

    {

        "id": 1,

        "name": "hedi",

        "position": "Professeur",

        "salary": 4500.0

    }

]

1. **Vérification des données de l’employé :**

J’ai ajouté une vérification simple dans la méthode addEmployee pour m’assurer que les données des employés sont valides avant de les enregistrer. Si le salaire est négatif, ou si le nom ou le poste est vide, mon code bloque et envoie un message d’erreur clair.

**Modifications :**

Dans **EmployeeController.java**

@PostMapping

    public Employee addEmployee(@RequestBody Employee employee) {

*// Vérification des données d'entrée:*

        if (employee.getSalary() < 0 ||

            employee.getName() == null || employee.getName().isEmpty() ||

            employee.getPosition() == null || employee.getPosition().isEmpty()) {

            throw new InvalidEmployeeDataException("Invalid employee data");

        }

        return employeeService.addEmployee(employee);

    }

J’ai ajouté aussi 2 fichiers **GlobalExceptionHandler.java** et **InvalidEmployeeDataException.java** pour afficher un message d’errur intuitive et clair.

1. **Ajout de date d’embauche de l’employé:**

J’ai ajouté un nouvel attribut à la classe Employee : **hireDate**, qui représente la date d’embauche de l’employé. J’ai également mis en place un nouvel endpoint **/employment\_duration/{id}** permettant de retourner la durée d’emploi d’un employé.

Voir code sur Github : <https://github.com/YassWorks/GestionPersonnel.git>