**Yêu cầu**

Bạn cần xây dựng một RESTful API cho một ứng dụng quản lý nhân sự. Cơ sở dữ liệu đã được chuẩn bị với các bảng (collections) sau:

1. **Employee**:
   * \_id: ObjectId
   * FullName: Object chứa {FirstName, LastName}
   * BasicSalary: Number
   * Departments: Array chứa các \_id của Department
   * Position: Object chứa \_id của Position
2. **Department**:
   * \_id: ObjectId
   * Name: Tên phòng ban
   * Code: Mã phòng ban
3. **Position**:
   * \_id: ObjectId
   * Name: Tên chức vụ
   * Coefficient: Hệ số lương

Yêu cầu

1. Tạo mới một Employee và tự động cập nhật lương dựa trên hệ số Position

Phương thức: POST

Endpoint: /api/employees

Payload:

{

"FullName": { "FirstName": "Nguyen", "LastName": "Anh" },

"BasicSalary": 1000,

"Departments": ["6489c29fae513b6ee8c6a001"],

"Position": "6489c29fae513b6ee8c6a004"

}

Chức năng:

* Thêm một Employee mới.
* Tìm Coefficient của Position từ database. Dựa vào BasicSalary và Coefficient, tự động tính lương thực nhận (TotalSalary = BasicSalary \* Coefficient).

Trả về thông tin Employee sau khi tạo:

{

"\_id": "6489c29fae513b6ee8c6a007",

"FullName": { "FirstName": "Nguyen", "LastName": "Anh" },

"BasicSalary": 1000,

"TotalSalary": 1500,

"Departments": ["6489c29fae513b6ee8c6a001"],

"Position": "Manager"

}

2. Lấy danh sách Employee và lọc theo các điều kiện

Phương thức: GET

Endpoint: /api/employees/:minSalary

Chức năng:

Lọc những nhân viên có TotalSalary >= minSalary.

Mỗi Employee trả về sẽ bao gồm tên đầy đủ của phòng ban và vị trí thay vì chỉ \_id.

Response mẫu:

[

{

"\_id": "6489c29fae513b6ee8c6a007",

"FullName": { "FirstName": "Nguyen", "LastName": "Anh" },

"BasicSalary": 1000,

"TotalSalary": 1500,

"Departments": ["Human Resources"],

"Position": "Manager"

}

]

3. Cập nhật thông tin Employee và tự động kiểm tra xung đột phòng ban

Phương thức: PUT

Endpoint: /api/employees/:id

{

"Departments": ["6489c29fae513b6ee8c6a002", "6489c29fae513b6ee8c6a003"],

"Position": "6489c29fae513b6ee8c6a006"

}

Chức năng:

Cập nhật thông tin Employee.

Kiểm tra nếu Employee được thêm vào nhiều hơn 4 phòng ban thì trả về lỗi:

{

"error": "An employee cannot belong to more than 4 departments."

}

Tính lại TotalSalary dựa trên Position mới.

Trả về thông tin Employee sau khi cập nhật:

{

"\_id": "6489c29fae513b6ee8c6a007",

"FullName": { "FirstName": "Nguyen", "LastName": "Anh" },

"BasicSalary": 1200,

"TotalSalary": 1200,

"Departments": ["IT Department", "Marketing"],

"Position": "Staff"

}

4. Xóa Employee và kiểm tra ràng buộc

Phương thức: DELETE

Endpoint: /api/employees/:id

Chức năng:

Trước khi xóa, kiểm tra xem Employee có phải là người duy nhất thuộc một phòng ban không.

Nếu Employee là người duy nhất thuộc một phòng ban, trả về lỗi:

{

"error": "Cannot delete the last employee in a department."

}

Nếu điều kiện thỏa mãn, xóa Employee và trả về thông báo thành công:

{

"message": "Employee deleted successfully."

}