

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA ĐÀO TẠO SAU ĐẠI HỌC**



**BÀI TẬP LỚN  
CƠ SỞ DỮ LIỆU NÂNG CAO**

**Đề tài: Xây dựng hệ thống quản lý  
một Toà nhà văn phòng.**

**GIẢNG VIÊN :**

**SINH VIÊN :**

**MÃ SINH VIÊN :**

**TS. DƯƠNG TRẦN ĐỨC**

**NGUYỄN QUANG HUY**

**B24CHKH012**

**Hà Nội – 2024**

# MỤC LỤC

Đề bài	3
Phần 1 : Xây dựng bản thiết kế CSDL với các thông tin và yêu cầu được cung cấp.	4
1.Thiết kế cơ sở dữ liệu	4
2. Nội dung các bảng dữ liệu :	5
3. Code tạo bảng trên mongodb :	7
Phần 2 :Xây dựng bộ dữ liệu mẫu, thực thi CSDL trong Hệ quản trị CSDL và nhập các dữ liệu mẫu.	9
1. Bộ Dữ Liệu Mẫu	10
2.Thực thi CSDL trong Hệ quản trị CSDL và nhập các dữ liệu mẫu.	10
Phần 3 : Viết các câu lệnh truy vấn theo yêu cầu.	13
Truy vấn 1	13
Truy vấn 2	15
Truy vấn 3	17
Phần 4 :Nâng cao: Viết chương trình hoặc phát triển ứng dụng thao tác với CSDL.	19
Model xây dựng trong nodeJS	19
Các API liên quan tới CongTy	22
Các API liên quan tới DichVu	25
Các API liên quan tới CongTyDichVu	27
Các API liên quan tới API NhanVienCongTy	32
Các API liên quan tới NhatKyRaVao	35
Các API liên quan tới NhanVienToaNha	38
Các API liên quan tới NhanVienDichVu	42

## Đề bài

### **PROJECT 4:**

Kịch bản thế giới thực: **Xây dựng hệ thống quản lý một Toà nhà văn phòng.**

**Các yêu cầu về CSDL bao gồm:**

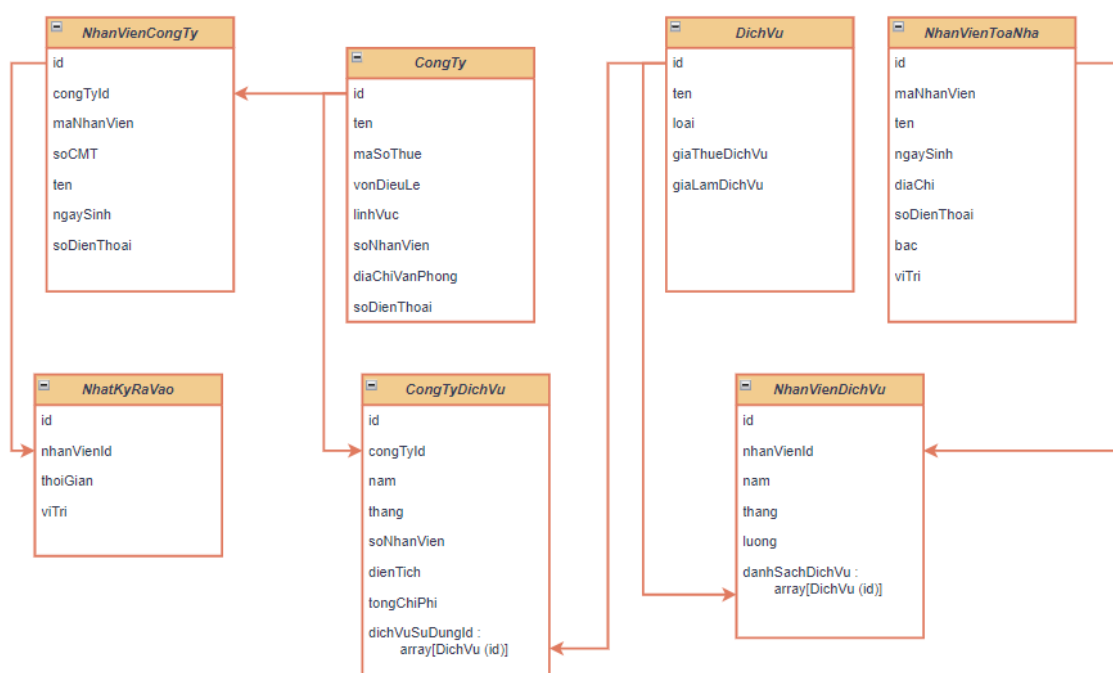
- Các công ty thuê văn phòng trong toà nhà có các thông tin cơ bản như tên công ty, mã số thuế, vốn điều lệ, lĩnh vực hoạt động, số nhân viên, địa chỉ trong toà nhà, số điện thoại, diện tích mặt bằng.
- Các nhân viên trong công ty có thông tin về mã nhân viên, CMT, tên, ngày sinh, số điện thoại.
- Các dịch vụ trong toà nhà gồm thông tin về mã số dịch vụ, tên dịch vụ, loại dịch vụ, đơn giá.
- Các Nhân viên của toà nhà gồm các thông tin về mã nhân viên, tên, ngày sinh, địa chỉ, số điện thoại, bậc, vị trí.
- Các công ty sử dụng các dịch vụ của toà nhà, bao gồm dịch vụ vệ sinh, ăn uống, trông giữ xe, bảo vệ, bảo trì thiết bị. Đơn giá các dịch vụ tỉ lệ thuận với **số người trong công ty và diện tích sàn** mà công ty đó thuê trong toà nhà. Các đơn giá của mỗi dịch vụ được tính theo công thức như sau: công ty dưới 10 người và thuê dưới 100 m<sup>2</sup> thì có cùng mức giá dịch vụ cho mỗi dịch vụ, cứ thêm 5 người hoặc thêm 10 m<sup>2</sup> diện tích sàn thì đơn giá tăng lên 5% cho mỗi mục). Mỗi công ty bắt buộc phải sử dụng dịch vụ bảo vệ và vệ sinh, các dịch vụ khác có thể sử dụng hoặc không. Tiền dịch vụ được tính từ ngày đăng ký, hoặc từ đầu tháng đến thời điểm hiện tại. Số tiền dịch vụ được tính bằng tỉ lệ giữa ngày đã sử dụng trên tổng số ngày trong tháng.
- Mỗi nhân viên công ty sẽ được cấp một thẻ ra vào và thông tin về số lần ra/vào toà nhà cần được lưu trữ lại. Thông tin về mỗi lần ra/vào toà nhà bao gồm vị trí ra/vào, thời gian ra/vào. Vị trí ra/vào bao gồm tầng 1, hầm B1, hầm B2.
- Các nhân viên của toà nhà thực hiện cung cấp và giám sát các dịch vụ. Lương của các nhân viên được tính theo từng vị trí và theo từng loại dịch vụ mà họ thực hiện. Lương nhân viên tỉ lệ thuận với doanh thu của từng loại dịch vụ. Học viên tự nghĩ ra cách tính lương theo kịch bản này.

**Các yêu cầu truy vấn:**

- Liệt kê thông tin của các công ty cùng với tổng số tiền mỗi tháng tính đến thời điểm hiện tại mà các công ty phải trả bao gồm tiền thuê mặt bằng (diện tích mặt bằng nhân đơn giá) và tổng tiền dịch vụ. Danh sách công ty được sắp xếp theo thứ tự giảm dần các chi phí.
- Kiểm tra thông tin của mỗi nhân viên của các công ty cùng với số lần và vị trí ra/vào toà nhà trong ngày của họ.
- Liệt kê thông tin của các nhân viên toà nhà cùng lương tháng của họ. Một nhân viên toà nhà có thể đổi vị trí (bậc công việc, và tên dịch vụ) làm việc theo mỗi tháng khác nhau.
- Các ràng buộc về số lượng bản ghi cần phải được thể hiện trong ứng dụng.

## Phần 1 : Xây dựng bản thiết kế CSDL với các thông tin và yêu cầu được cung cấp.

### 1. Thiết kế cơ sở dữ liệu



#### Giải thích về thiết kế

1. Bảng CongTy: Bảng này lưu trữ thông tin cơ bản về các công ty, như tên, mã số thuế, vốn điều lệ, lĩnh vực hoạt động, số nhân viên, địa chỉ văn phòng, số điện thoại và diện tích văn phòng.

2. Bảng DichVu : Lưu trữ thông tin về các dịch vụ mà tòa nhà cung cấp, gồm tên, loại dịch vụ, giá thuê và giá làm dịch vụ. Ở đây trường giaThueDichVu và giaLamDichVu cần thiết cho việc đánh giá tổng tiền công ty thuê dịch vụ và tổng tiền nhân viên tòa nhà đăng ký làm dịch vụ

3. Bảng CongTyDichVu : Bảng này lưu giữ thông tin về các dịch vụ mà công ty sử dụng trong từng tháng, bao gồm chi tiết về số lượng nhân viên và diện tích văn phòng liên quan. congTyId tham chiếu đến CongTy để duy trì quan hệ 1-nhiều giữa công ty và các dịch vụ mà họ sử dụng. "nam" và "thang" giúp theo dõi các dịch vụ theo từng thời điểm, điều này rất quan trọng cho việc lập báo cáo và phân tích chi phí theo thời gian.

Chỉ mục kết hợp congTyId, nam, thang được tạo để đảm bảo một công ty không bị trùng lặp việc sử dụng dịch vụ cho cùng một tháng và năm, giúp duy trì tính toàn vẹn dữ liệu.

4. Bảng NhanVienCongTy: Lưu trữ thông tin về nhân viên của mỗi công ty, bao gồm tên, mã nhân viên, chứng minh thư (CMT), ngày sinh, và số điện thoại. congTyId tham chiếu đến CongTy để liên kết mỗi nhân viên với công ty họ làm việc, giúp dễ dàng quản lý thông tin nhân viên.

5. Bảng NhatKyRaVao: Bảng này ghi lại thời gian ra vào và vị trí của nhân viên. nhanVienId tham chiếu đến NhanVienCongTy, liên kết mỗi bản ghi ra vào với nhân viên tương ứng. thoiGian và viTri giúp theo dõi hoạt động và vị trí của nhân viên, cần thiết cho việc bảo mật và giám sát trong tòa nhà.

6. Bảng NhanVienToaNha : Lưu trữ thông tin nhân viên làm việc trực tiếp tại tòa nhà, như nhân viên bảo vệ, vệ sinh, v.v.

7. Bảng NhanVienDichVu : Lưu trữ thông tin về các dịch vụ mà nhân viên tòa nhà thực hiện trong từng tháng, bao gồm lương và danh sách dịch vụ. nhanVienId tham chiếu đến NhanVienToaNha, giúp xác định nhân viên thực hiện dịch vụ. nam và thang giúp theo dõi hoạt động và lương của nhân viên theo từng thời gian. danhSachDichVu là một mảng các tham chiếu đến DichVu, cho phép quản lý chi tiết những dịch vụ mà nhân viên đã thực hiện.

## 2. Nội dung các bảng dữ liệu :

### 2.1. Bảng CongTy (Công ty)

Lưu thông tin công ty.

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc
id	ObjectID	PRIMARY KEY
ten	string	NOT NULL
maSoThue	string	
vonDieuLe	number	
linhVuc	string	
soNhanVien	number	
diaChiVanPhong	string	
soDienThoai	string	
dienTich	number	

### 2.2. Bảng DichVu (Dịch vụ)

Lưu thông tin dịch vụ.

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc
id	ObjectID	PRIMARY KEY
ten	string	
loai	string	
giaThueDichVu	number	
giaLamDichVu	number	

### 2.3. Bảng CongTyDichVu (Công ty - Dịch vụ)

Lưu thông tin về dịch vụ mà công ty sử dụng trong từng tháng.

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc
id	ObjectID	PRIMARY KEY
congTyId	number	FOREIGN KEY REFERENCES CongTy(id)
nam	number	NOT NULL
thang	number	NOT NULL
soNhanVien	number	
dienTich	number	
tongChiPhi	number	
dichVuSuDungId	array	Chứa danh sách các DịchVu (id)

### 2.4. Bảng NhanVienCongTy (Nhân viên công ty)

Lưu thông tin nhân viên thuộc các công ty.

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc
id	ObjectID	PRIMARY KEY
congTyId	number	FOREIGN KEY REFERENCES CongTy(id)
maNhanVien	string	UNIQUE
soCMT	string	
ten	string	
ngaySinh	date	
soDienThoai	string	

### 2.5. Bảng NhatKyRaVao (Nhật ký ra vào)

Lưu nhật ký ra vào của nhân viên.

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc
id	ObjectID	PRIMARY KEY
nhanVienId	number	FOREIGN KEY REFERENCES NhanVienCongTy(id)
thoiGian	date	
viTri	string	

## 2.6. Bảng NhanVienToaNha (Nhân viên tòa nhà)

Lưu thông tin nhân viên làm việc tại tòa nhà.

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc
id	ObjectID	PRIMARY KEY
maNhanVien	string	UNIQUE
ten	string	
ngaySinh	date	
diaChi	string	
soDienThoai	string	
bac	string	
viTri	string	

## 2.7. Bảng NhanVienDichVu (Nhân viên dịch vụ)

Lưu thông tin nhân viên tòa nhà và các dịch vụ mà họ thực hiện hàng tháng.

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc
id	ObjectID	PRIMARY KEY
nhanVienId	number	FOREIGN KEY REFERENCES NhanVienToaNha(id)
nam	number	NOT NULL
thang	number	NOT NULL
luong	number	NOT NULL
danhSachDichVu	array	Chứa danh sách các DịchVu (id)

## 3. Code tạo bảng trên mongodb :

Dưới đây là cách chuyển đổi các đoạn mã sử dụng Mongoose để tạo ra các bảng tương ứng trong MongoDB, sử dụng script MongoDB để tạo các collections và chỉ mục (indexes).

### 3.1. Tạo Collection CongTy

```
db.createCollection("CongTy", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["ten"],
      properties: {
        ten: { bsonType: "string", description: "Tên công ty" },
        maSoThue: { bsonType: "string" },
        vonDieuLe: { bsonType: "double" },
        linhVuc: { bsonType: "string" },
        soNhanVien: { bsonType: "int" },
        diaChiVanPhong: { bsonType: "string" },
        soDienThoai: { bsonType: "string" },
        dienTich: { bsonType: "double" }
      }
    }
  }
})
```

```

    }
  }
}
});

```

### 3.2. Tạo Collection DichVu

```

db.createCollection("DichVu", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      properties: {
        ten: { bsonType: "string" },
        loai: { bsonType: "string" },
        giaThueDichVu: { bsonType: "double" },
        giaLamDichVu: { bsonType: "double" }
      }
    }
  }
});

```

### 3.3. Tạo Collection CongTyDichVu

```

db.createCollection("CongTyDichVu", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      properties: {
        congTyId: { bsonType: "objectId" },
        nam: { bsonType: "int" },
        thang: { bsonType: "int" },
        soNhanVien: { bsonType: "int" },
        dienTich: { bsonType: "double" },
        tongChiPhi: { bsonType: "double" },
        dichVuSuDungId: {
          bsonType: "array",
          items: { bsonType: "objectId" }
        }
      }
    }
  }
});

```

```

// Thêm chỉ mục kết hợp cho `congTyId`, `nam`, `thang`
db.CongTyDichVu.createIndex({ congTyId: 1, nam: 1, thang: 1 }, { unique: true });

```

### 3.4. Tạo Collection NhanVienCongTy

```

db.createCollection("NhanVienCongTy", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      properties: {
        congTyId: { bsonType: "objectId" },
        maNhanVien: { bsonType: "string" },
        soCMT: { bsonType: "string" },
        ten: { bsonType: "string" },
        ngaySinh: { bsonType: "date" },
        soDienThoai: { bsonType: "string" }
      }
    }
  }
});

```

### 3.5. Tạo Collection NhatKyRaVao

```

db.createCollection("NhatKyRaVao", {

```



```

    validator: {
      $jsonSchema: {
        bsonType: "object",
        properties: {
          nhanVienId: { bsonType: "objectId" },
          thoiGian: { bsonType: "date" },
          viTri: { bsonType: "string" }
        }
      }
    }
  });

```

### 3.6. Tạo Collection NhanVienToaNha

```

db.createCollection("NhanVienToaNha", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      properties: {
        maNhanVien: { bsonType: "string" },
        ten: { bsonType: "string" },
        ngaySinh: { bsonType: "date" },
        diaChi: { bsonType: "string" },
        soDienThoai: { bsonType: "string" },
        bac: { bsonType: "string" },
        viTri: { bsonType: "string" }
      }
    }
  }
});

```

### 3.7. Tạo Collection NhanVienDichVu

```

db.createCollection("NhanVienDichVu", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["nhanVienId", "nam", "thang", "luong"],
      properties: {
        nhanVienId: { bsonType: "objectId" },
        nam: { bsonType: "int" },
        thang: { bsonType: "int" },
        luong: { bsonType: "double" },
        danhSachDichVu: {
          bsonType: "array",
          items: { bsonType: "objectId" }
        }
      }
    }
  }
});

```

```

// Đảm bảo `nam` + `thang` là duy nhất cho mỗi nhân viên
db.NhanVienDichVu.createIndex({ nhanVienId: 1, nam: 1, thang: 1 }, { unique: true });

```

**Phần 2 :Xây dựng bộ dữ liệu mẫu, thực thi CSDL trong Hệ quản trị CSDL và nhập các dữ liệu mẫu.**

## 1. Bộ Dữ Liệu Mẫu

1. **CongTy (Công ty)** :Số Lượng Bản Ghi: Có thể có **10** bản ghi mẫu, mỗi bản ghi đại diện cho một công ty khác nhau với các thông tin chi tiết.
2. **DichVu (Dịch vụ)** :Mô Tả: Lưu thông tin về các dịch vụ được cung cấp trong tòa nhà, như dịch vụ vệ sinh, an ninh, bảo trì, v.v. Số Lượng Bản Ghi: Có thể có **6** bản ghi, mỗi bản ghi đại diện cho một loại dịch vụ khác nhau.
3. **CongTyDichVu (Công ty - Dịch vụ)** :Lưu trữ thông tin về các dịch vụ mà công ty sử dụng trong tòa nhà, bao gồm chi tiết về dịch vụ được sử dụng, số lượng nhân viên và diện tích được thuê trong từng tháng.Số Lượng Bản Ghi: Khoảng **250** bản ghi, mỗi bản ghi đại diện cho một công ty sử dụng một dịch vụ vào một khoảng thời gian cụ thể (theo năm và tháng).
4. **NhanVienCongTy (Nhân viên công ty)** :Số Lượng Bản Ghi: **1500** bản ghi, mỗi bản ghi đại diện cho một nhân viên thuộc các công ty khác nhau.
5. **NhatKyRaVao (Nhật ký ra vào)** Mô Tả: Lưu thông tin về các lần ra vào tòa nhà của nhân viên, bao gồm thời gian và vị trí ra vào. Số Lượng Bản Ghi: **4500** bản ghi, mỗi bản ghi đại diện cho một lần ra vào của nhân viên, được ghi lại để quản lý an ninh.
6. **NhanVienToaNha (Nhân viên tòa nhà)** :Số Lượng Bản Ghi: có **50** bản ghi, mỗi bản ghi đại diện cho một nhân viên thuộc tòa nhà.
7. **NhanVienDichVu (Nhân viên dịch vụ)** :Mô Tả: Lưu trữ thông tin về các nhân viên của tòa nhà làm việc với các dịch vụ cụ thể trong từng tháng. Thông tin này bao gồm lương và các dịch vụ mà nhân viên phụ trách. Số Lượng Bản Ghi: **3000** bản ghi, mỗi bản ghi đại diện cho một nhân viên làm việc tại tòa nhà và các dịch vụ mà họ đảm nhiệm.

## 2.Thực thi CSDL trong Hệ quản trị CSDL và nhập các dữ liệu mẫu.

## Kết quả:

**CongTy (Công ty)**

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 10 of 10

```
_id: ObjectId('66ff5f9939539f59aa855a45')
ten: "Công ty Công ty TNHH Hòa Phát"
maSoThue: "585608111"
vonDieuLe: 26014547
linhVuc: "Xây dựng"
soNhanVien: 122
diaChiVanPhong: "Số 84, Đường ABC, TP Hồ Chí Minh"
soDienThoai: "0900861064"
dienTich: 16
```

```
_id: ObjectId('66ff5f9939539f59aa855a46')
ten: "Công ty Công ty CP Bình Dương"
maSoThue: "756607911"
vonDieuLe: 21043466
linhVuc: "Giáo dục"
soNhanVien: 105
diaChiVanPhong: "Số 62, Đường ABC, TP Hồ Chí Minh"
soDienThoai: "0952946029"
dienTich: 25
```

**DichVu (Dịch vụ)**

+

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 6 of 6

<pre>_id: ObjectId('66ff5fd922150a3f1ebf438c') ten: "Thuê mặt bằng" loai: "Dịch vụ" giaThueDichVu: 1000000 giaLamDichVu: 800000</pre>
<pre>_id: ObjectId('66ff5fd922150a3f1ebf438d') ten: "Vệ sinh" loai: "Dịch vụ" giaThueDichVu: 5000000 giaLamDichVu: 4000000</pre>
<pre>_id: ObjectId('66ff5fd922150a3f1ebf438e') ten: "Ăn uống" loai: "Dịch vụ" giaThueDichVu: 5000000 giaLamDichVu: 3000000</pre>

**CongTyDichVu (Công ty - Dịch vụ)**

+

ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 25 of 250

```
_id: ObjectId('66ff63696512d978bc070b25')
congTyId: ObjectId('66ff5f9939539f59aa855a45')
nam: 2020
thang: 1
soNhanVien: 122
dienTich: 16
tongChiPhi: 10000000
dichVuSuDungId: Array (2)
  0: ObjectId('66ff5fd922150a3f1ebf4391')
  1: ObjectId('66ff5fd922150a3f1ebf438d')
```

```
_id: ObjectId('66ff63696512d978bc070b26')
congTyId: ObjectId('66ff5f9939539f59aa855a45')
nam: 2020
thang: 2
soNhanVien: 122
dienTich: 16
tongChiPhi: 6000000
dichVuSuDungId: Array (1)
```

**NhanVienCongTy (Nhân viên công ty)**



<div> <div>ADD DATA</div> <div>EXPORT DATA</div> <div>UPDATE</div> <div>DELETE</div> </div> <div>251 - 25 of 50</div>
<pre> _id: ObjectId('66ff6e63e3eaaa7efe9f7167') maNhanVien: "623d9f06540644c7bdfd4bddb39653f7" ten: "Anh Tấn Hoàng" ngaySinh: "2003-10-17" diaChi: "65 Đặng Đường         Quận JaneQuận, 556442" soDienThoai: "(07)915-2086" bac: "Mid" viTri: "Luật sư" </pre>
<pre> _id: ObjectId('66ff6e63e3eaaa7efe9f7168') maNhanVien: "a6ecbe48daf7498f91153f7561806a23" ten: "Vũ Phú Phạm" ngaySinh: "1993-03-01" diaChi: "11 Mai Khu         Thị xã JanePhường, 139249" soDienThoai: "+84-37-941 0998" bac: "Mid" viTri: "Project Manager" </pre>
<pre> _id: ObjectId('66ff6e63e3eaaa7efe9f7169') maNhanVien: "dd8944633d9c41cab9458e4ca16bfbbc" </pre>

NhanVienDichVu (Nhân viên dịch vụ)

<div> <div>ADD DATA</div> <div>EXPORT DATA</div> <div>UPDATE</div> <div>DELETE</div> </div> <div>251 - 25 of 3000</div>
<pre> _id: ObjectId('66ff6e75ac9c2381a2b22499') nhanVienId: ObjectId('66ff6e63e3eaaa7efe9f7167') nam: 2020 thang: 1 luong: 9000000 danhSachDichVu: Array (3)   0: ObjectId('66ff5fd922150a3f1ebf438c')   1: ObjectId('66ff5fd922150a3f1ebf4391')   2: ObjectId('66ff5fd922150a3f1ebf438f') </pre>
<pre> _id: ObjectId('66ff6e75ac9c2381a2b2249a') nhanVienId: ObjectId('66ff6e63e3eaaa7efe9f7167') nam: 2020 thang: 2 luong: 13000000 danhSachDichVu: Array (3) </pre>
<pre> _id: ObjectId('66ff6e75ac9c2381a2b2249b') nhanVienId: ObjectId('66ff6e63e3eaaa7efe9f7167') nam: 2020 thang: 3 luong: 11000000 </pre>

Phần 3 : Viết các câu lệnh truy vấn theo yêu cầu.

1.Truy vấn 1

Liệt kê thông tin của các công ty cùng với tổng số tiền mỗi tháng tính đến thời điểm hiện tại mà các công ty phải trả bao gồm tiền thuê mặt bằng (diện tích mặt bằng nhân đơn giá) và tổng tiền dịch vụ. Danh sách công ty được sắp xếp theo thứ tự giảm dần các chi phí.

Code :

```
db.CongTyDichVu.aggregate([
  {
    // Chỉ lấy các bản ghi thuộc năm 2023
    $match: {
      nam: 2023
    }
  },
  {
    // Nhóm các bản ghi lại dựa trên `congTyId`
    $group: {
      _id: "$congTyId",
      tongChiPhi: { $sum: "$tongChiPhi" } // Tính tổng chi phí theo mỗi
      công ty cho năm 2023
    }
  },
  {
    // Tham gia thêm thông tin từ collection CongTy vào
    $lookup: {
      from: "CongTy",
      localField: "_id",
      foreignField: "_id",
      as: "congTy"
    }
  },
  {
    // Giải nén (unwind) trường congTy để có được thông tin công ty như
    một đối tượng
    $unwind: "$congTy"
  },
  {
    // Tạo ra các trường cần thiết để trả về kết quả
    $project: {
      _id: 0,
      //congTyId: "$_id",
      ten: "$congTy.ten",
      //maSoThue: "$congTy.maSoThue",
      //soNhanVien: "$congTy.soNhanVien",
      //diaChiVanPhong: "$congTy.diaChiVanPhong",
      tongChiPhi: 1
    }
  },
  {
    // Sắp xếp theo tổng chi phí giảm dần
    $sort: { tongChiPhi: -1 }
  }
]).pretty();
```

Kết quả :

```
)).pretty();
< {
  tongChiPhi: 71000000,
  ten: 'Công ty Công ty TNHH Đầu Tư An Phát'
}
{
  tongChiPhi: 65000000,
  ten: 'Công ty Công ty TNHH Dịch Vụ Hồng Hà'
}
{
  tongChiPhi: 63000000,
  ten: 'Công ty Công ty TNHH Hòa Phát'
}
{
  tongChiPhi: 60000000,
  ten: 'Công ty Công ty CP Bình Dương'
}
{
  tongChiPhi: 58000000,
  ten: 'Công ty Công ty TNHH Xây Dựng Đông Á'
}
{
  tongChiPhi: 58000000,
  ten: 'Công ty Công ty CP Sản Xuất Hoàng Gia'
}
```

## 2.Truy vấn 2

Kiểm tra thông tin của mỗi nhân viên của các công ty cùng với số lần và vị trí ra/vào toà nhà trong ngày của họ.

Code

```
db.NhanVienCongTy.aggregate([
  {
    // Lọc ra các nhân viên thuộc công ty có congTyId =
    "66ff5f9939539f59aa855a45"
    $match: {
      congTyId: "66ff5f9939539f59aa855a45"
    }
  },
  {
    // Tham gia thêm thông tin từ collection NhatKyRaVao vào
    NhanVienCongTy qua nhanVienId
    $lookup: {
      from: "NhatKyRaVao",
      localField: "_id",
      foreignField: "nhanVienId",
      as: "nhatKyRaVao"
    }
  },
  {
```

```

    // Giải nén mảng nhatKyRaVao để có thể lọc theo năm 2024
    $unwind: {
      path: "$nhatKyRaVao",
      preserveNullAndEmptyArrays: true // Cho phép nhân viên không có
bản ghi ra vào
    }
  },
  {
    // Lọc chỉ các bản ghi có thoiGian thuộc năm 2024
    $match: {
      "nhatKyRaVao.thoiGian": {
        $gte: new Date("2024-01-01T00:00:00Z"),
        $lt: new Date("2025-01-01T00:00:00Z")
      }
    }
  },
  {
    // Nhóm lại theo nhanVienId để tính tổng số lần ra vào của mỗi nhân
viên
    $group: {
      _id: "$_id",
      ten: { $first: "$ten" },
      maNhanVien: { $first: "$maNhanVien" },
      soDienThoai: { $first: "$soDienThoai" },
      congTyId: { $first: "$congTyId" },
      tongSoLanRaVao: { $sum: 1 } // Đếm tổng số lần ra vào
    }
  },
  {
    // Lựa chọn các trường cần thiết để trả về kết quả
    $project: {
      _id: 0,
      maNhanVien: 1,
      ten: 1,
      soDienThoai: 1,
      congTyId: 1,
      tongSoLanRaVao: 1
    }
  }
]).pretty();

```

Kết quả :

Ví dụ công ty có id là **66ff5f9939539f59aa855a45** và test năm **2024**



```
>_MONGOSH
< {
  ten: 'Nhật Trần',
  maNhanVien: '436c84fac439419b913cc885a3ed4bd1',
  soDienThoai: '+84 94 4574608',
  congTyId: '66ff5f9939539f59aa855a45',
  tongSoLanRaVao: 1
}
{
  ten: 'Bác Trung Bùi',
  maNhanVien: '7c0f405004e1465a8e1466aa265d1966',
  soDienThoai: '(07)748-5536',
  congTyId: '66ff5f9939539f59aa855a45',
  tongSoLanRaVao: 2
}
{
  ten: 'Bà Vĩ Dương',
  maNhanVien: '4f26d04fec014bc7b8bdd4bb49ccc9ac',
  soDienThoai: '00 9674915',
  congTyId: '66ff5f9939539f59aa855a45',
  tongSoLanRaVao: 1
}
{
  ten: 'Chị Nhật Dương',
  maNhanVien: '96ba475718774a71a424f1a7a58a0061',
  soDienThoai: '+84-75-586931',
```

```
>_MONGOSH
  congTyId: '66ff5f9939539f59aa855a45',
  tongSoLanRaVao: 2
}
{
  ten: 'Nhật Hoàng Đặng',
  maNhanVien: '15d5d76ae30c4712ab9a8a99ebd12c60',
  soDienThoai: '+84 54 5044281',
  congTyId: '66ff5f9939539f59aa855a45',
  tongSoLanRaVao: 2
}
{
  ten: 'Hạnh Văn Nguyễn',
  maNhanVien: '5082f7acca214739b8867f8eb53d5e5d',
  soDienThoai: '(01) 0171 4730',
  congTyId: '66ff5f9939539f59aa855a45',
  tongSoLanRaVao: 1
}
{
  ten: 'Hạnh Mai Dương',
  maNhanVien: '1e9b27c2ee2c400cbb29b0b138d3b010',
  soDienThoai: '(07)226-9037',
  congTyId: '66ff5f9939539f59aa855a45',
  tongSoLanRaVao: 2
}
}
```

### 3.Truy vấn 3

Liệt kê thông tin của các nhân viên toà nhà cùng lương tháng của họ. Một nhân viên toà nhà có thể đổi vị trí (bậc công việc, và tên dịch vụ) làm việc theo mỗi tháng khác nhau.

## Code

```
db.NhanVienDichVu.aggregate([
  {
    // Lọc ra các bản ghi có năm là 2024
    $match: {
      nam: 2024
    }
  },
  {
    // Nhóm các bản ghi lại theo nhanVienId để tính lương cho từng nhân
    $group: {
      _id: "$nhanVienId",
      tongLuong: { $sum: "$luong" },
      chiTietLuong: {
        $push: {
          thang: "$thang",
          luong: "$luong"
        }
      }
    }
  },
  {
    // Tham gia thêm thông tin từ collection NhanVienToaNha vào từng nhân
    $lookup: {
      from: "NhanVienToaNha",
      localField: "_id",
      foreignField: "_id",
      as: "nhanVien"
    }
  },
  {
    // Giải nén thông tin nhân viên để dễ dàng truy cập
    $unwind: "$nhanVien"
  },
  {
    // Lựa chọn các trường cần thiết để trả về kết quả
    $project: {
      _id: 0,
      maNhanVien: "$nhanVien.maNhanVien",
      ten: "$nhanVien.ten",
      tongLuong: 1,
      chiTietLuong: 1
    }
  },
  {
    // Sắp xếp kết quả theo tên nhân viên
  }
])
```

```

    $sort: {
      ten: 1
    }
  }
  ]).pretty();

```

Kết quả

```

>_MONGOSH
{
  tongLuong: 143000000,
  chiTietLuong: [
    {
      thang: 1,
      luong: 9000000
    },
    {
      thang: 2,
      luong: 13000000
    },
    {
      thang: 3,
      luong: 12000000
    },
    {
      thang: 4,
      luong: 11000000
    },
    {
      thang: 5,
      luong: 12000000
    },
    {
      thang: 6,

```

```

>_MONGOSH
},
{
  thang: 6,
  luong: 12000000
},
{
  thang: 7,
  luong: 12000000
},
{
  thang: 8,
  luong: 9000000
},
{
  thang: 9,
  luong: 12000000
},
{
  thang: 10,
  luong: 14000000
},
{
  thang: 11,
  luong: 12000000
},
{

```

```

>_MONGOSH
{
  thang: 12,
  luong: 15000000
},
],
maNhanVien: '2cc5cac774ba42089a6dba8d0814c05a',
ten: 'Bà Phương Bùi'
},
{
  tongLuong: 131000000,
  chiTietLuong: [
    {
      thang: 1,
      luong: 14000000
    },
    {
      thang: 2,
      luong: 12000000
    },
    {
      thang: 3,
      luong: 8000000
    },
    {
      thang: 4,

```

Phần 4 :Nâng cao: Viết chương trình hoặc phát triển ứng dụng thao tác với CSDL.

1.Model xây dựng trong nodeJS

CongTy

```

const mongoose = require('mongoose');

const CongTySchema = new mongoose.Schema({
  ten: { type: String, required: true },
  maSoThue: { type: String },
  vonDieuLe: { type: Number },

```

```

    linhVuc: { type: String },
    soNhanVien: { type: Number },
    diaChiVanPhong: { type: String },
    soDienThoai: { type: String },
    dienTich: { type: Number }
  }, { collection: 'CongTy' }); // Chỉ định tên collection

module.exports = mongoose.models.CongTy || mongoose.model('CongTy',
CongTySchema);

```

'DichVu'

```

const mongoose = require('mongoose');

const DichVuSchema = new mongoose.Schema({
  ten: { type: String },
  loai: { type: String },
  giaThueDichVu: { type: Number }, //công ty thuê mất tiền
  giaLamDichVu: { type: Number } // nhân viên tòa nhà làm được tiền
}, { collection: 'DichVu' }); // Chỉ định tên collection

module.exports = mongoose.models.DichVu || mongoose.model('DichVu',
DichVuSchema);

```

'CongTyDichVu'

```

const mongoose = require('mongoose');

const CongTyDichVuSchema = new mongoose.Schema({
  congTyId: { type: mongoose.Schema.Types.ObjectId, ref: 'CongTy' },
  nam: { type: Number },
  thang: { type: Number },
  soNhanVien: { type: Number },
  dienTich: { type: Number },
  tongChiPhi: { type: Number },
  dichVuSuDungId: [ {
    type: mongoose.Schema.Types.ObjectId, ref: 'DichVu'
  } ]
}, { collection: 'CongTyDichVu' }); // Chỉ định tên collection

// Thêm chỉ mục kết hợp cho congTyId, nam, thang
CongTyDichVuSchema.index({ congTyId: 1, nam: 1, thang: 1 }, { unique: true });

```

```
module.exports = mongoose.models.CongTyDichVu ||
mongoose.model('CongTyDichVu', CongTyDichVuSchema);
```

### 'NhanVienCongTy'

```
const mongoose = require('mongoose');

const NhanVienCongTySchema = new mongoose.Schema({
  congTyId: { type: mongoose.Schema.Types.ObjectId, ref: 'CongTy' },
  maNhanVien: { type: String },
  soCMT: { type: String },
  ten: { type: String },
  ngaySinh: { type: Date },
  soDienThoai: { type: String }
}, { collection: 'NhanVienCongTy' }); // Chỉ định tên collection

module.exports = mongoose.models.NhanVienCongTy ||
mongoose.model('NhanVienCongTy', NhanVienCongTySchema);
```

### 'NhatKyRaVao'

```
const mongoose = require('mongoose');

const NhatKyRaVaoSchema = new mongoose.Schema({
  nhanVienId: { type: mongoose.Schema.Types.ObjectId, ref: 'NhanVienCongTy' },
  thoiGian: { type: Date },
  viTri: { type: String }
}, { collection: 'NhatKyRaVao' }); // Chỉ định tên collection

module.exports = mongoose.models.NhatKyRaVao || mongoose.model('NhatKyRaVao',
NhatKyRaVaoSchema);
```

### 'NhanVienToaNha'

```
const mongoose = require('mongoose');

const NhanVienToaNhaSchema = new mongoose.Schema({
  maNhanVien: { type: String },
  ten: { type: String },
  ngaySinh: { type: Date },
  diaChi: { type: String },
  soDienThoai: { type: String },
  bac: { type: String },
  viTri: { type: String },
```

```

}, { collection: 'NhanVienToaNha' }); // Chỉ định tên collection

module.exports = mongoose.models.NhanVienToaNha ||
mongoose.model('NhanVienToaNha', NhanVienToaNhaSchema);

```

## NhanVienDichVu

```

const mongoose = require('mongoose');

const NhanVienDichVuSchema = new mongoose.Schema({
  nhanVienId: { type: mongoose.Schema.Types.ObjectId, ref: 'NhanVienToaNha' },
  // Tham chiếu đến nhân viên
  nam: { type: Number, required: true },
  thang: { type: Number, required: true },
  luong: { type: Number, required: true },
  danhSachDichVu: [{ type: mongoose.Schema.Types.ObjectId, ref: 'DichVu' }]
}, { collection: 'NhanVienDichVu' });

// Đảm bảo năm + tháng là unique cho mỗi nhân viên
NhanVienDichVuSchema.index({ nhanVienId: 1, nam: 1, thang: 1 }, { unique: true });

module.exports = mongoose.models.LuongHangThang ||
mongoose.model('NhanVienDichVu', NhanVienDichVuSchema);

```

## 2.Các API liên quan tới CôngTy

API CôngTy bao gồm tổng cộng bốn endpoint để thực hiện các tác vụ CRUD (Create, Read, Update, Delete) cho dữ liệu công ty

Lấy danh sách tất cả các công ty

URL: GET <http://localhost:8080/api/congty>

HTTP CongTy / <http://localhost:8080/api/congty> Save Share

GET <http://localhost:8080/api/congty> Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
-----	-------	-------------	-----------

Body Cookies Headers (7) Test Results 200 OK • 116 ms • 3.58 KB • Save Response

Pretty Raw Preview Visualize JSON Copy Search

```
1 [
2   {
3     "_id": "66e53f2cdf97800ad11177a0",
4     "ten": "Công ty Công ty TNHH Hòa Phát 2",
5     "maSoThue": "142917545",
6     "vonDieuLe": 34412203,
7     "linhVuc": "Công nghệ",
8     "soNhanVien": 150,
9     "diaChiVanPhong": "Số 2, Đường ABC, TP Hồ Chí Minh",
10    "soDienThoai": "0918958161",
11    "dienTich": 24
12  },
13  {
14    "_id": "66e53f2cdf97800ad11177a1",
15    "ten": "Công ty Công ty CP Bình Dương",
16    "maSoThue": "754949007",
17    "vonDieuLe": 34104349,
18    "linhVuc": "Giáo dục",
```

Postbot Runner Start Proxy Cookies Vault Trash

Cập nhật thông tin một công ty cụ thể

URL: PUT <http://localhost:8080/api/congty/66e170906d0a41afc4acb...>

HTTP CongTy / <http://localhost:8080/api/congty/66e170906d0a41afc4acba38> Save Share

PUT <http://localhost:8080/api/congty/66e53f2cdf97800ad11177a0> Send

Params Authorization Headers (9) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "ten": "Công ty ABC Update",
3   "soNhanVien": 151
4 }
5
```

Body Cookies Headers (7) Test Results 200 OK • 115 ms • 485 B • Save Response

Pretty Raw Preview Visualize JSON Copy Search

```
1 {
2   "_id": "66e53f2cdf97800ad11177a0",
3   "ten": "Công ty ABC Update",
4   "maSoThue": "142917545",
5   "vonDieuLe": 34412203,
6   "linhVuc": "Công nghệ",
7   "soNhanVien": 151,
8   "diaChiVanPhong": "Số 2, Đường ABC, TP Hồ Chí Minh",
9   "soDienThoai": "0918958161",
10  "dienTich": 24
11 }
```

## Tạo mới một công ty

URL: POST <http://localhost:8080/api/congty>

CongTy / <http://localhost:8080/api/congty>

POST

[Save](#) [Share](#)

Send

Params

Authorization

Headers (9)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   "ten": "Công ty ABC",
3   "maSoThue": "123456789",
4   "vonDieuLe": 500000000,
5   "linhVuc": "Công nghệ thông tin",
6 }

```

Body

Cookies

Headers (7)

Test Results

201 Created

55 ms

473 B

[Save Response](#)

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "ten": "Công ty ABC",
3   "maSoThue": "123456789",
4   "vonDieuLe": 500000000,
5   "linhVuc": "Công nghệ thông tin",
6   "soNhanVien": 100,
7   "diaChiVanPhong": "Hà Nội",
8   "soDienThoai": "0123456789",
9   "dienTich": 200,
10  "_id": "66ff5c48e830809162e7d5ef",
11  "__v": 0
12 }

```

## Xóa một công ty dựa trên ID

URL: DELETE <http://localhost:8080/api/congty/id>

CongTy / <http://localhost:8080/api/congty/id>

DELETE

[Save](#) [Share](#)

Send

Params

Authorization

Headers (7)

Body

Scripts

Settings

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

200 OK

46 ms

290 B

[Save Response](#)

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "message": "Công ty đã được xóa thành công!"
3 }

```



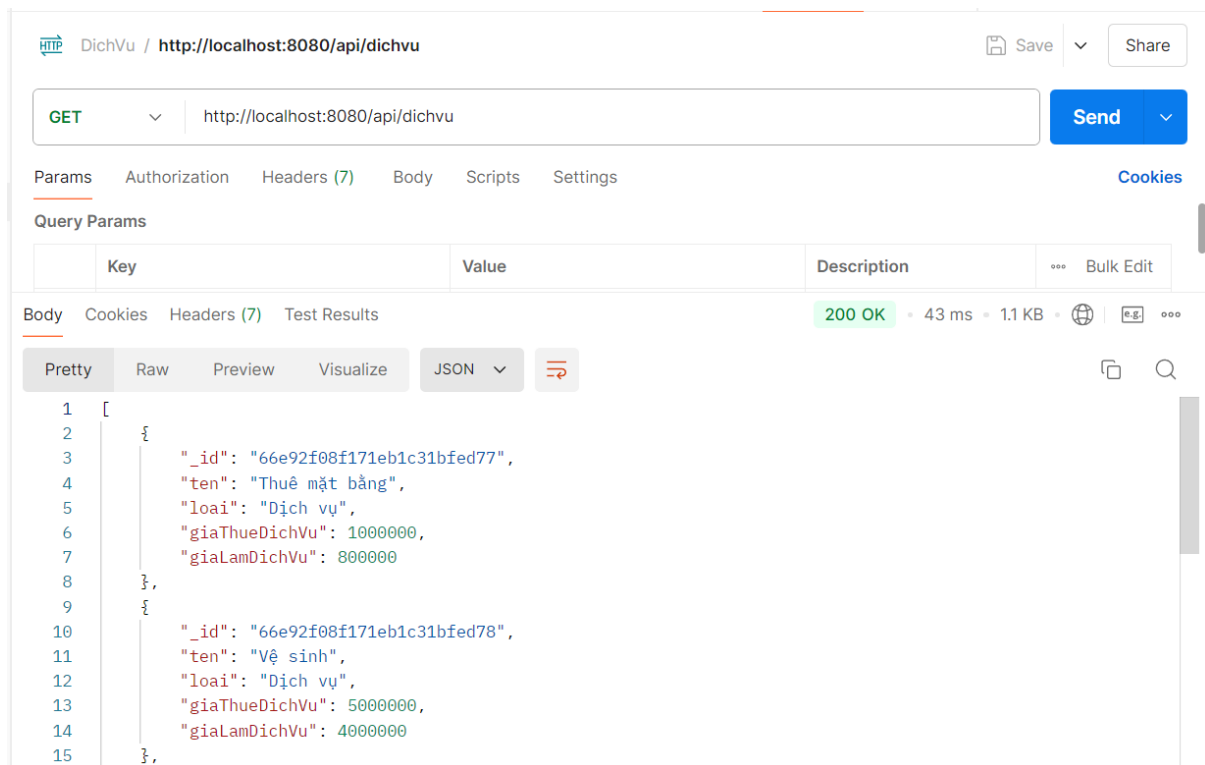
### 3. Các API liên quan tới Dịch Vụ

Hệ thống API cho Dịch Vụ bao gồm tổng cộng bốn endpoint để thực hiện các tác vụ CRUD (Create, Read, Update, Delete) cho dữ liệu dịch vụ

Dưới đây là các endpoint cụ thể:

Lấy danh sách tất cả các dịch vụ

URL: GET <http://localhost:8080/api/dichvu>



The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/api/dichvu`
- Method:** GET
- Status:** 200 OK
- Response Time:** 43 ms
- Response Size:** 1.1 KB
- Response Body (JSON):**

```
[
  {
    "_id": "66e92f08f171eb1c31bfed77",
    "ten": "Thuê mặt bằng",
    "loai": "Dịch vụ",
    "giaThueDichVu": 1000000,
    "giaLamDichVu": 800000
  },
  {
    "_id": "66e92f08f171eb1c31bfed78",
    "ten": "Vệ sinh",
    "loai": "Dịch vụ",
    "giaThueDichVu": 5000000,
    "giaLamDichVu": 4000000
  }
]
```

Cập nhật thông tin một dịch vụ cụ thể

URL: PUT <http://localhost:8080/api/dichvu/id>

HTTP DichVu / <http://localhost:8080/api/dichvu/id> Save Share

**PUT** <http://localhost:8080/api/dichvu/66e92f08f171eb1c31bfed77> Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "ten": "Dịch vụ sửa chữa",
3   "donGiaCoBan": 600000
4 }
```

Body Cookies Headers (7) Test Results **200 OK** • 53 ms • 348 B • 🌐 🔍 ⋮

Pretty Raw Preview Visualize **JSON** 🔍

```
1 {
2   "_id": "66e92f08f171eb1c31bfed77",
3   "ten": "Dịch vụ sửa chữa",
4   "loai": "",
5   "giaThueDichVu": 0,
6   "giaLamDichVu": 0
7 }
```

Tạo mới một dịch vụ

URL: POST <http://localhost:8080/api/dichvu>

HTTP DichVu / <http://localhost:8080/api/dichvu> Save Share

**POST** <http://localhost:8080/api/dichvu> Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "ten": "Dịch vụ dọn dẹp1",
3   "loai": "Dọn dẹp",
4   "giaThueDichVu": 500000,
5   "giaLamDichVu": 400000
6 }
7
```

Body Cookies Headers (7) Test Results **200 OK** • 49 ms • 377 B • 🌐 🔍 ⋮

Pretty Raw Preview Visualize **JSON** 🔍

```
1 {
2   "ten": "Dịch vụ dọn dẹp1",
3   "loai": "Dọn dẹp",
4   "giaThueDichVu": 500000,
5   "giaLamDichVu": 400000,
6   "_id": "66ff5db9e830809162e7d5f6",
7   "__v": 0
8 }
```

Xóa một dịch vụ dựa trên ID

URL: DELETE <http://localhost:8080/api/dichvu/id>

The screenshot shows a REST client interface with the following components:

- URL Bar:** Displays the URL `http://localhost:8080/api/dichvu/id` and a `DELETE` method dropdown.
- Request Body:** Contains the ID `http://localhost:8080/api/dichvu/66ff5db9e830809162e7d5f6`.
- Response:** Shows a `200 OK` status with a response time of `49 ms` and a body of `293 B`.
- Response Body:** Displays the JSON response `{ "message": "Dịch vụ đã được xóa thành công!" }` in a pretty-printed format.

Key	Value	Description
Key	Value	Description

#### 4. Các API liên quan tới CongTyDichVu

Hệ thống API cho tài nguyên CongTyDichVu bao gồm tổng cộng năm endpoint để thực hiện các tác vụ CRUD (Create, Read, Update, Delete) cho dữ liệu dịch vụ mà công ty sử dụng.

Dưới đây là các endpoint cụ thể:

Lấy danh sách tất cả các dịch vụ mà các công ty đang sử dụng

URL: GET <http://localhost:8080/api/congtydichvu>

HTTP CongTyDichVu / <http://localhost:8080/api/congtydichvu> Save Share

GET <http://localhost:8080/api/congtydichvu> Send

Params Authorization Headers (7) Body Scripts Settings Cookies

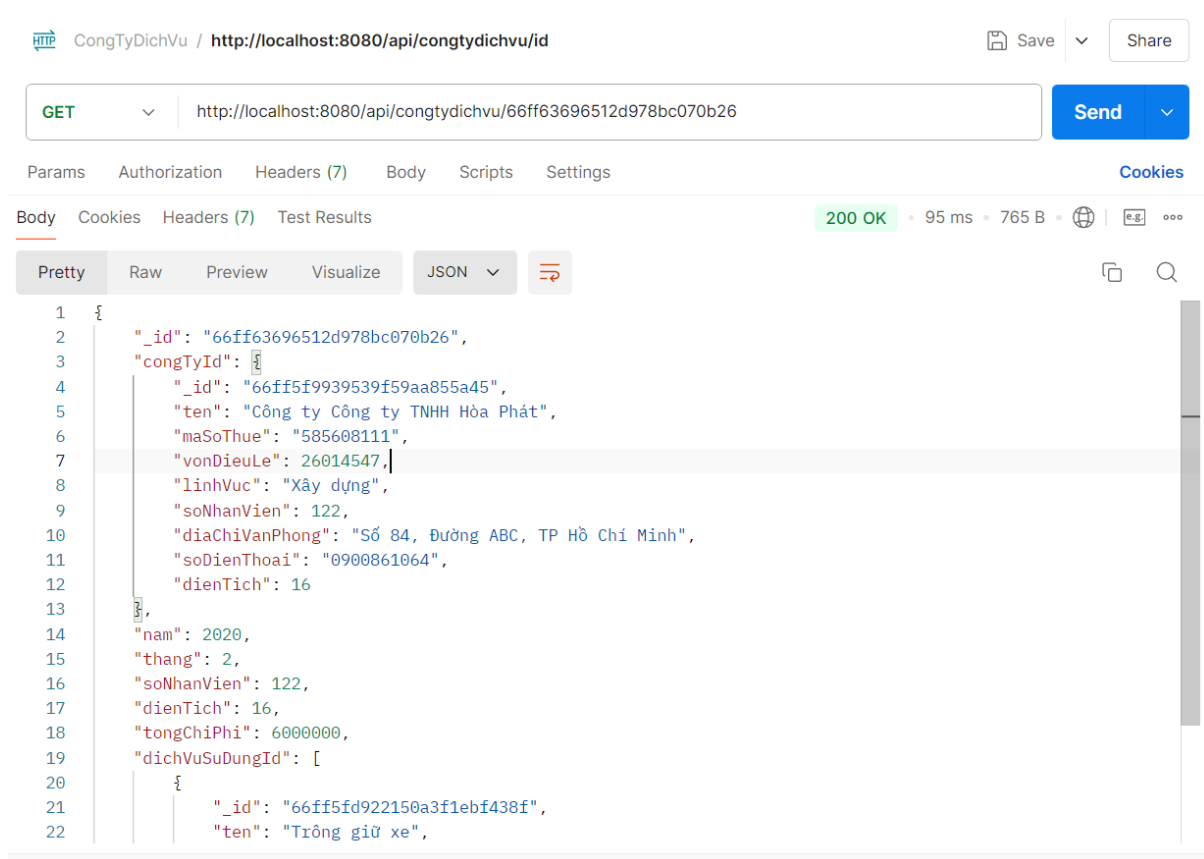
Body Cookies Headers (7) Test Results 200 OK • 35 ms • 1.07 KB •

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "_id": "66e932ac03c6fc31fd66fff8",
4     "congTyId": "66e92f99ad3dd9947765362e",
5     "nam": 2023,
6     "thang": 9,
7     "soNhanVien": 150,
8     "dienTich": 500,
9     "tongChiPhi": 2000000,
10    "dichVuSuDungId": [],
11    "__v": 0
12  },
13  {
14    "_id": "66e932c203c6fc31fd66fffc",
15    "congTyId": "66e53f2cdf97800ad11177a0",
16    "nam": 2023,
17    "thang": 9,
18    "soNhanVien": 150,
19    "dienTich": 500,
20    "tongChiPhi": 2000000,
21    "dichVuSuDungId": [
22      "66e5468045fc7a442b316f73",
```

Lấy thông tin về một dịch vụ mà công ty sử dụng dựa trên ID

URL: GET <http://localhost:8080/api/congtydichvu/id>



Cập nhật thông tin về dịch vụ mà công ty sử dụng dựa trên ID

URL: PUT <http://localhost:8080/api/congtydichvu/id>

CongTyDichVu / <http://localhost:8080/api/congtydichvu/id> Save Share

**PUT** <http://localhost:8080/api/congtydichvu/66ff63696512d978bc070b26> Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "soNhanVien": 60,
3   "dienTich": 250,
4   "tongChiPhi": 1200000
5 }
6
```

**Body** Cookies Headers (7) Test Results 200 OK • 134 ms • 428 B • 🌐 🔍 ⋮

**Pretty** Raw Preview Visualize **JSON** 🔍

```
1 {
2   "_id": "66ff63696512d978bc070b26",
3   "congTyId": "66ff5f939539f59aa855a45",
4   "nam": 2020,
5   "thang": 2,
6   "soNhanVien": 60,
7   "dienTich": 250,
8   "tongChiPhi": 24500000,
9   "dichVuSuDungId": [
10    "66ff5fd922150a3f1ebf438f"
11  ]
12 }
```

Tạo mới thông tin dịch vụ mà công ty sử dụng

URL: POST <http://localhost:8080/api/congtydichvu>

HTTP CongTyDichVu / http://localhost:8080/api/congtydichvu

POST http://localhost:8080/api/congtydichvu

Params Authorization Headers (9) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "congTyId": "66ff5f9939539f59aa855a45", // Thay bằng ID thực của công ty
3   "nam": 2023,
4   "thang": 10,
5   "soNhanVien": 150,
6   "dienTich": 500,
7   "tongChiPhi": 2000000,
8   "dichVuSuDung": ["66ff5fd922150a3f1ebf438d", "66ff5fd922150a3f1ebf438c" ]
9 }
10
```

Body Cookies Headers (7) Test Results 200 OK • 81 ms • 412 B • 🌐 📄 🔍

Pretty Raw Preview Visualize JSON 🔍

```
1 {
2   "congTyId": "66ff5f9939539f59aa855a45",
3   "nam": 2023,
4   "thang": 10,
5   "soNhanVien": 150,
6   "dienTich": 500,
7   "tongChiPhi": 40000000,
8   "dichVuSuDung": [
9     "66ff5fd922150a3f1ebf438d",
10    "66ff5fd922150a3f1ebf438c"
11  ]
12 }
```

Xóa thông tin dịch vụ mà công ty sử dụng dựa trên ID

URL: DELETE <http://localhost:8080/api/congtydichvu/id>

HTTP CongTyDichVu / http://localhost:8080/api/congtydichvu/id

DELETE http://localhost:8080/api/congtydichvu/66ffae8840ff6c830d5396dd

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK • 60 ms • 302 B • 🌐 📄 🔍

Pretty Raw Preview Visualize JSON 🔍

```
1 {
2   "message": "Công ty dịch vụ đã được xóa thành công!"
3 }
```

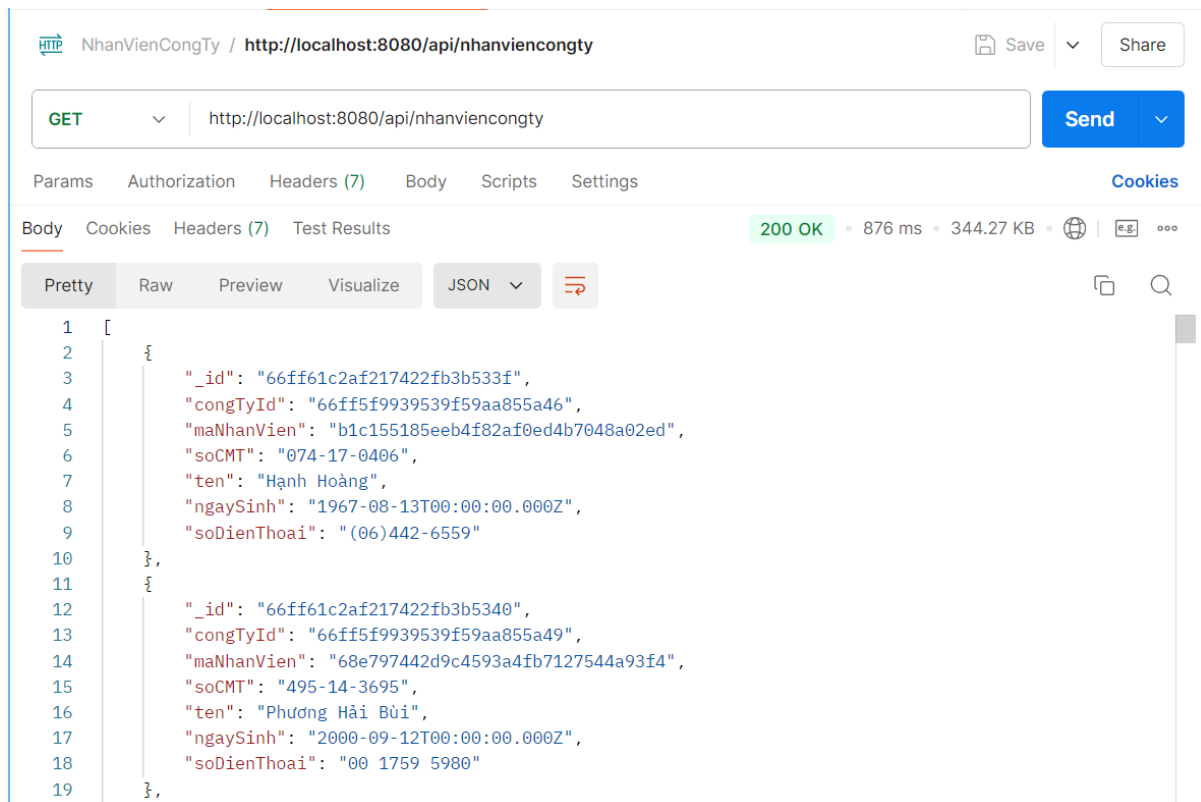
## 5. Các API liên quan tới API NhanVienCongTy

Các API được cung cấp cho NhanVienCongTy, bao gồm các phương thức để thực hiện các thao tác cơ bản như lấy danh sách nhân viên, lấy chi tiết một nhân viên, cập nhật, thêm mới và xóa:

### 1. Lấy danh sách tất cả nhân viên công ty

URL: GET /api/nhanviencongtty

Mô Tả: API này được sử dụng để lấy tất cả thông tin của các nhân viên thuộc công ty, bao gồm các thông tin như tên, mã nhân viên, ngày sinh, số điện thoại, v.v.



The screenshot shows a REST client interface with the following details:

- URL: `http://localhost:8080/api/nhanviencongtty`
- Method: `GET`
- Status: `200 OK` (876 ms, 344.27 KB)
- Response Body (JSON):

```
[
  {
    "_id": "66ff61c2af217422fb3b533f",
    "congtTyId": "66ff5f9939539f59aa855a46",
    "maNhanVien": "b1c155185eeb4f82af0ed4b7048a02ed",
    "soCMT": "074-17-0406",
    "ten": "Hạnh Hoàng",
    "ngaySinh": "1967-08-13T00:00:00.000Z",
    "soDienThoai": "(06)442-6559"
  },
  {
    "_id": "66ff61c2af217422fb3b5340",
    "congtTyId": "66ff5f9939539f59aa855a49",
    "maNhanVien": "68e797442d9c4593a4fb7127544a93f4",
    "soCMT": "495-14-3695",
    "ten": "Phương Hải Bùi",
    "ngaySinh": "2000-09-12T00:00:00.000Z",
    "soDienThoai": "00 1759 5980"
  }
]
```

### 2. Lấy thông tin chi tiết của một nhân viên công ty cụ thể dựa vào id

URL: GET /api/nhanviencongtty/:id

Mô Tả: API này cho phép người dùng truy cập vào thông tin chi tiết của một nhân viên cụ thể bằng cách cung cấp id của nhân viên. Thông tin bao gồm tên, mã nhân viên, số CMT/CCCD, ngày sinh, địa chỉ, số điện thoại, và công ty mà nhân viên thuộc về.



GET http://localhost:8080/api/nhanviencongt/66ff61c2af217422fb3b533f Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (7) Test Results 200 OK • 73 ms • 706 B • [Globe] [E.g.] [More]

Pretty Raw Preview Visualize JSON [More]

```
1 {
2   "_id": "66ff61c2af217422fb3b533f",
3   "congTyId": {
4     "_id": "66ff5f9939539f59aa855a46",
5     "ten": "Công ty Công ty CP Bình Dương",
6     "maSoThue": "756607911",
7     "vonDieuLe": 21043466,
8     "linhVuc": "Giáo dục",
9     "soNhanVien": 105,
10    "diaChiVanPhong": "Số 62, Đường ABC, TP Hồ Chí Minh",
11    "soDienThoai": "0952946029",
12    "dienTich": 25
13  },
14 }
```

### 3. Cập nhật thông tin của một nhân viên cụ thể dựa vào id

URL: PUT /api/nhanviencongt/:id

Mô Tả: API này cho phép người dùng cập nhật thông tin của một nhân viên đã tồn tại trong hệ thống. Người dùng cần cung cấp id của nhân viên và dữ liệu mới để thực hiện việc cập nhật (ví dụ: cập nhật tên, địa chỉ, số điện thoại, v.v.).

NhanVienCongTy / http://localhost:8080/api/nhanviencongt/id Save Share

PUT http://localhost:8080/api/nhanviencongt/66ff61c2af217422fb3b533f Send

Params Authorization Headers (9) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "ten": "Nguyễn Văn A",
3   "soNhanVien": 150
4 }
5
```

Body Cookies Headers (7) Test Results 200 OK • 88 ms • 469 B • [Globe] [E.g.] [More]

Pretty Raw Preview Visualize JSON [More]

```
1 {
2   "_id": "66ff61c2af217422fb3b533f",
3   "congTyId": "66ff5f9939539f59aa855a46",
4   "maNhanVien": "b1c155185eeb4f82af0ed4b7048a02ed",
5   "soCMT": "074-17-0406",
6   "ten": "Nguyễn Văn A",
7   "ngaySinh": "1967-08-13T00:00:00.000Z",
8   "soDienThoai": "(06)442-6559"
9 }
```

#### 4. Thêm mới một nhân viên công ty vào hệ thống

URL: POST /api/nhanviencongtty

Mô Tả: API này cho phép thêm mới một nhân viên vào collection NhanVienCongTy. Dữ liệu cần thiết để tạo nhân viên mới bao gồm mã nhân viên, tên, ngày sinh, địa chỉ, và các thông tin cá nhân khác.

HTTP NhanVienCongTy / http://localhost:8080/api/nhanviencongtty

POST http://localhost:8080/api/nhanviencongtty

Params Authorization Headers (9) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "congTyId": "66ff5f9939539f59aa855a45",
3   "maNhanVien": "NV001",
4   "soCMT": "123456789",
5   "ten": "Nguyễn Văn Ánhhh",
6   "ngaySinh": "1990-01-01",
7   "soDienThoai": "0987654321"
}
```

Body Cookies Headers (7) Test Results 200 OK • 97 ms • 451 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔍

```
1 {
2   "congTyId": "66ff5f9939539f59aa855a45",
3   "maNhanVien": "NV001",
4   "soCMT": "123456789",
5   "ten": "Nguyễn Văn Ánhhh",
6   "ngaySinh": "1990-01-01T00:00:00.000Z",
7   "soDienThoai": "0987654321",
8   "_id": "66ffbca040ff6c830d5396ed",
9   "__v": 0
10 }
```

Postbot

#### 5. Xóa một nhân viên cụ thể dựa vào id

URL: DELETE /api/nhanviencongtty/:id

Mô Tả: API này được sử dụng để xóa thông tin của một nhân viên cụ thể khỏi hệ thống bằng cách cung cấp id của nhân viên.

HTTP NhanVienCongTy / <http://localhost:8080/api/nhanviencongtv/id> Save Share

DELETE <http://localhost:8080/api/nhanviencongtv/66ffbca040ff6c830d5396ed> Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK 42 ms 293 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Nhân viên đã được xóa thành công!"
3 }
```

## 6. Các API liên quan tới NhatKyRaVao

### 1. Lấy danh sách tất cả các bản ghi nhật ký ra vào

URL: GET <http://localhost:8080/api/nhatkyraovao>

HTTP NhatKyRaVao / <http://localhost:8080/api/nhatkyravao> Save Share

GET <http://localhost:8080/api/nhatkyravao> Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Body Cookies Headers (7) Test Results 200 OK 1188 ms 580.32 KB

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "_id": "66ffa58a9262b6c3587cb3df",
4     "nhanVienId": "66ff61c2af217422fb3b533f",
5     "thoiGian": "2024-07-17T02:34:44.000Z",
6     "viTri": "Tầng 2"
7   },
8   {
9     "_id": "66ffa58a9262b6c3587cb3e0",
10    "nhanVienId": "66ff61c2af217422fb3b533f",
11    "thoiGian": "2023-08-26T17:44:03.000Z",
12    "viTri": "Tầng 2"
13  },
14  {
15    "_id": "66ffa58a9262b6c3587cb3e1",
16    "nhanVienId": "66ff61c2af217422fb3b533f",
17    "thoiGian": "2023-09-04T05:46:27.000Z",
18    "viTri": "Tầng 2"
19  },
20 ]
```

### 2. Lấy thông tin chi tiết của một bản ghi ra vào cụ thể

URL: GET <http://localhost:8080/api/nhatkyraovao/:id>

HTTP NhatKyRaVao / <http://localhost:8080/api/nhatkyravao/id> Save Share

GET <http://localhost:8080/api/nhatkyravao/66ffa58a9262b6c3587cb3e6> Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK • 71 ms • 570 B • 🌐 🔍 ⋮

Pretty Raw Preview Visualize JSON 🔍

```
1 {
2   "_id": "66ffa58a9262b6c3587cb3e6",
3   "nhanVienId": {
4     "_id": "66ff61c2af217422fb3b5341",
5     "congTyId": "66ff5f9939539f59aa855a4c",
6     "maNhanVien": "9e83fa768c4a4b3886bd56216de30859",
7     "soCMT": "463-24-2733",
8     "ten": "Bác Nam Vũ",
9     "ngaySinh": "1993-02-11T00:00:00.000Z",
10    "soDienThoai": "03 7829832"
11  },
12  "thoiGian": "2023-09-28T21:46:44.000Z",
13  "viTri": "Tầng 1"
14 }
```

### 3. Lấy danh sách các bản ghi ra vào của một nhân viên cụ thể

URL: GET <http://localhost:8080/api/nhatkyraovao/nhanvien/:nhanVienId>

HTTP NhatKyRaVao / <http://localhost:8080/api/nhatkyraovao/nhanvien/<nhanVienId>> Save Share

GET <http://localhost:8080/api/nhatkyravao/nhanvien/66ff61c2af217422fb3b533f> Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Body Cookies Headers (7) Test Results 200 OK • 67 ms • 634 B • 🌐 🔍 ⋮

Pretty Raw Preview Visualize JSON 🔍

```
1 [
2   {
3     "_id": "66ffa58a9262b6c3587cb3df",
4     "nhanVienId": "66ff61c2af217422fb3b533f",
5     "thoiGian": "2024-07-17T02:34:44.000Z",
6     "viTri": "Tầng 2"
7   },
8   {
9     "_id": "66ffa58a9262b6c3587cb3e0",
10    "nhanVienId": "66ff61c2af217422fb3b533f",
11    "thoiGian": "2023-08-26T17:44:03.000Z",
12    "viTri": "Tầng 2"
13  },
14  {
15    "_id": "66ffa58a9262b6c3587cb3e1",
16    "nhanVienId": "66ff61c2af217422fb3b533f",
17    "thoiGian": "2023-09-04T05:46:27.000Z",
18    "viTri": "Tầng 2"
19  }
20 ]
```

#### 4. Cập nhật thông tin của một bản ghi ra vào dựa vào id

URL: PUT <http://localhost:8080/api/nhatkyraovao/id>

The screenshot shows a REST client interface for a PUT request. The URL is `http://localhost:8080/api/nhatkyraovao/66ffa58a9262b6c3587cb3df`. The request body is a JSON object: `{ "viTri": "Tầng 1" }`. The response status is `200 OK` with a response time of 64 ms and a body size of 367 B. The response body is a JSON object: `{ "_id": "66ffa58a9262b6c3587cb3df", "nhanVienId": "66ff61c2af217422fb3b533f", "thoiGian": "2024-07-17T02:34:44.000Z", "viTri": "Tầng 1" }`.

```
1 {
2   | "viTri": "Tầng 1"
3   | }
4 }
```

Body Cookies Headers (7) Test Results **200 OK** • 64 ms • 367 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔽 ⚙️

```
1 {
2   | " _id": "66ffa58a9262b6c3587cb3df",
3   | "nhanVienId": "66ff61c2af217422fb3b533f",
4   | "thoiGian": "2024-07-17T02:34:44.000Z",
5   | "viTri": "Tầng 1"
6   | }
```

#### 5. Thêm mới một bản ghi nhật ký ra vào

URL: POST <http://localhost:8080/api/nhatkyraovao>

The screenshot shows a REST client interface for a POST request. The URL is `http://localhost:8080/api/nhatkyraovao`. The request body is a JSON object: `{ "nhanVienId": "66ff61c2af217422fb3b533f", "viTri": "Tầng 1" }`. The response status is `201 Created` with a response time of 69 ms and a body size of 380 B. The response body is a JSON object: `{ "nhanVienId": "66ff61c2af217422fb3b533f", "thoiGian": "2024-10-04T10:06:38.530Z", "viTri": "Tầng 1", "_id": "66ffbe2e40ffc830d5396f9", "__v": 0 }`.

```
1 {
2   | "nhanVienId": "66ff61c2af217422fb3b533f",
3   | "viTri": "Tầng 1"
4   | }
5 }
```

Body Cookies Headers (7) Test Results **201 Created** • 69 ms • 380 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔽 ⚙️

```
1 {
2   | "nhanVienId": "66ff61c2af217422fb3b533f",
3   | "thoiGian": "2024-10-04T10:06:38.530Z",
4   | "viTri": "Tầng 1",
5   | "_id": "66ffbe2e40ffc830d5396f9",
6   | "__v": 0
7   | }
```

6. Xóa một bản ghi nhật ký ra vào cụ thể dựa vào id

URL: DELETE <http://localhost:8080/api/nhatkyraovao/:id>

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/api/nhatkyraovao/id`
- Method:** DELETE
- Path:** `http://localhost:8080/api/nhatkyraovao/66ffbe2e40ff6c830d5396f9`
- Query Params:** A table with 4 columns: Key, Value, Description, and Bulk Edit. The first row contains 'Key', 'Value', 'Description', and a 'Bulk Edit' button.
- Response:** 200 OK, 42 ms, 300 B. The response body is a JSON object: 

```
{  "message": "Nhật ký ra vào đã được xóa thành công!"}
```

## 7. Các API liên quan tới NhanVienToaNha

Dưới đây là các API được cung cấp cho NhanVienToaNha, bao gồm các phương thức để thực hiện các thao tác cơ bản như lấy danh sách nhân viên, lấy chi tiết một nhân viên, cập nhật, thêm mới và xóa:

### 1. Lấy danh sách tất cả nhân viên tòa nhà

URL: GET `http://localhost:8080/api/nhanvientoanha`

Mô Tả: API này được sử dụng để lấy tất cả thông tin của các nhân viên tòa nhà, bao gồm tên, mã nhân viên, ngày sinh, địa chỉ, và số điện thoại.

HTTP NhanVienToaNha / http://localhost:8080/api/nhanvientoanha

GET http://localhost:8080/api/nhanvientoanha

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
-----	-------	-------------	-----------

Body Cookies Headers (7) Test Results 200 OK 47 ms 13.44 KB

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "_id": "66ff6e63e3eaaa7efe9f7167",
4     "maNhanVien": "623d9f06540644c7bdfd4bddb39653f7",
5     "ten": "Anh Tấn Hoàng",
6     "ngaySinh": "2003-10-17T00:00:00.000Z",
7     "diaChi": "65 Đặng Đường\nQuận JaneQuận, 556442",
8     "soDienThoai": "(07)915-2086",
9     "bac": "Mid",
10    "viTri": "Luật sư"
11  },
12  {
13    "_id": "66ff6e63e3eaaa7efe9f7168",
14    "maNhanVien": "a6ecbe48daf7498f91153f7561806a23",
15    "ten": "Vũ Phú Phạm",
16    "ngaySinh": "1993-03-01T00:00:00.000Z",
17    "diaChi": "11 Mai Khu\nThị xã JanePhường, 139249",
18    "soDienThoai": "+84-37-941 0998",
```

## 2. Lấy thông tin chi tiết của một nhân viên tòa nhà cụ thể dựa vào id

URL: GET http://localhost:8080/api/nhanvientoanha/:id

Mô Tả: API này cho phép người dùng truy cập vào thông tin chi tiết của một nhân viên tòa nhà cụ thể bằng cách cung cấp id của nhân viên. Thông tin bao gồm tên, mã nhân viên, số điện thoại, địa chỉ, và vị trí công việc của họ.

[HTTP](#) NhanVienToaNha / <http://localhost:8080/api/nhanvientoanha/id>

Save

Share

GET

http://localhost:8080/api/nhanvientoanha/66ff6e63e3eaaa7efe9f716c/

Send

ParamsAuthorizationHeaders (7)BodyScriptsSettingsCookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (7)Test Results200 OK • 34 ms • 515 B •

PrettyRawPreviewVisualizeJSON

```
1  {
2    "_id": "66ff6e63e3eaaa7efe9f716c",
3    "maNhanVien": "cf52d838207f44c8b92ef4f9f21e2e5a",
4    "ten": "Bác Tú Nguyễn",
5    "ngaySinh": "2004-02-19T00:00:00.000Z",
6    "diaChi": "17 Bùi Hẻm\nThị xã JohnQuận, 260059",
7    "soDienThoai": "(05) 8947 5932",
8    "bac": "Mid",
9    "viTri": "Quản lý sản xuất"
10 }
```

### 3. Cập nhật thông tin của một nhân viên tòa nhà dựa vào id

URL: PUT <http://localhost:8080/api/nhanvientoanha/:id>

Mô Tả: API này cho phép người dùng cập nhật thông tin của một nhân viên đã tồn tại trong hệ thống. Người dùng cần cung cấp id của nhân viên và các thông tin cần cập nhật (ví dụ: tên, địa chỉ, số điện thoại, vị trí công việc, v.v.).



The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/api/nhanvientoanha/66ff6e63e3eaaa7efe9f716c`. The request body is a JSON object with the following fields: `maNhanVien`, `ten`, `ngaySinh`, `diaChi`, `soDienThoai`, `bac`, and `viTri`. The response is a 200 OK status with a response time of 92 ms and a body size of 464 B. The response body is a JSON object that includes an `_id` field along with the original request data.

```
1 {
2   "maNhanVien": "NV001",
3   "ten": "Nguyen Thị Ái",
4   "ngaySinh": "1990-01-01T00:00:00.000Z",
5   "diaChi": "123 Đường ABC, TP HCM",
6   "soDienThoai": "0123456789",
7   "bac": "Trưởng phòng",
8   "viTri": "Quản lý"
9 }
```

200 OK • 92 ms • 464 B • [Globe] [R.E.] [More]

```
1 {
2   "_id": "66ff6e63e3eaaa7efe9f716c",
3   "maNhanVien": "NV001",
4   "ten": "Nguyen Thị Ái",
5   "ngaySinh": "1990-01-01T00:00:00.000Z",
6   "diaChi": "123 Đường ABC, TP HCM",
7   "soDienThoai": "0123456789",
8   "bac": "Trưởng phòng",
9   "viTri": "Quản lý"
10 }
```

#### 4. Thêm mới một nhân viên tòa nhà vào hệ thống

URL: POST `http://localhost:8080/api/nhanvientoanha`

Mô Tả: API này cho phép thêm mới một nhân viên vào collection `NhanVienToaNha`. Dữ liệu cần thiết để tạo nhân viên mới bao gồm mã nhân viên, tên, ngày sinh, địa chỉ, và thông tin vị trí công việc.

POST http://localhost:8080/api/nhanvientoanha

Send

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "maNhanVien": "NV002",
3   "ten": "Nguyen Van A",
4   "ngaySinh": "1990-01-01T00:00:00.000Z",
5   "diaChi": "123 Đường ABC, TP HCM",
6   "soDienThoai": "0123456789",
7   "bac": "Trưởng phòng",
8   "viTri": "Quản lý"
9 }
```

Body Cookies Headers (7) Test Results 201 Created • 45 ms • 473 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔍

```
1 {
2   "maNhanVien": "NV002",
3   "ten": "Nguyen Van A",
4   "ngaySinh": "1990-01-01T00:00:00.000Z",
5   "diaChi": "123 Đường ABC, TP HCM",
6   "soDienThoai": "0123456789",
7   "bac": "Trưởng phòng",
8   "viTri": "Quản lý",
9   "_id": "66ffc06640ff6c830d539705",
10  "__v": 0
}
```

## 5. Xóa một nhân viên tòa nhà cụ thể dựa vào id

URL: DELETE http://localhost:8080/api/nhanvientoanha/:id

Mô Tả: API này được sử dụng để xóa thông tin của một nhân viên cụ thể khỏi hệ thống bằng cách cung cấp id của nhân viên.

🔗 NhanVienToaNha / http://localhost:8080/api/nhanvientoanha/id Save Share

DELETE http://localhost:8080/api/nhanvientoanha/66ffc06640ff6c830d539705 Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	⋮ Bulk Edit
Key	Value	Description	

Body Cookies Headers (7) Test Results 200 OK • 42 ms • 293 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔍

```
1 {
2   "message": "Nhân viên đã được xóa thành công!"
3 }
```

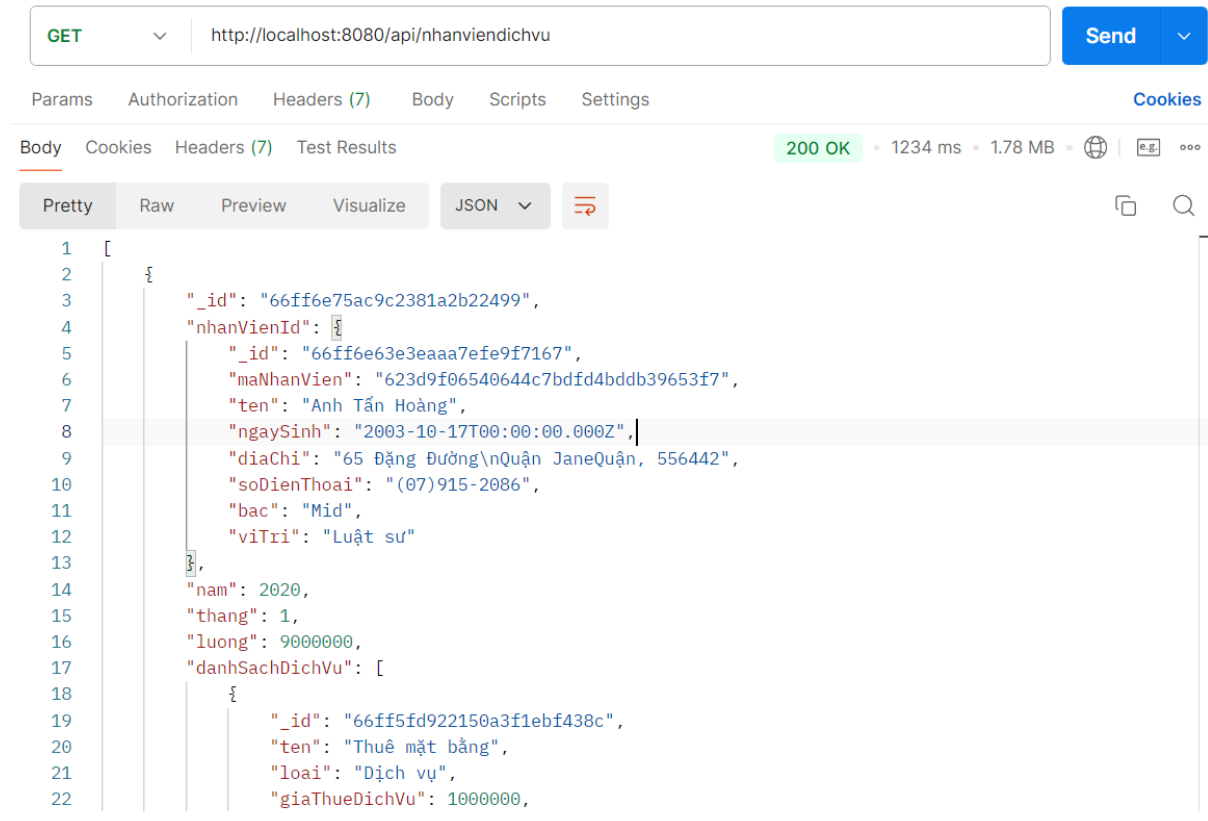
## 8. Các API liên quan tới NhanVienDichVu

Dưới đây là các API được cung cấp cho NhanVienDichVu, bao gồm các phương thức để thực hiện các thao tác cơ bản như lấy danh sách nhân viên dịch vụ, lấy chi tiết, cập nhật, thêm mới và xóa:

## 1. Lấy danh sách tất cả các bản ghi về nhân viên dịch vụ

URL: GET <http://localhost:8080/api/nhanviendichvu>

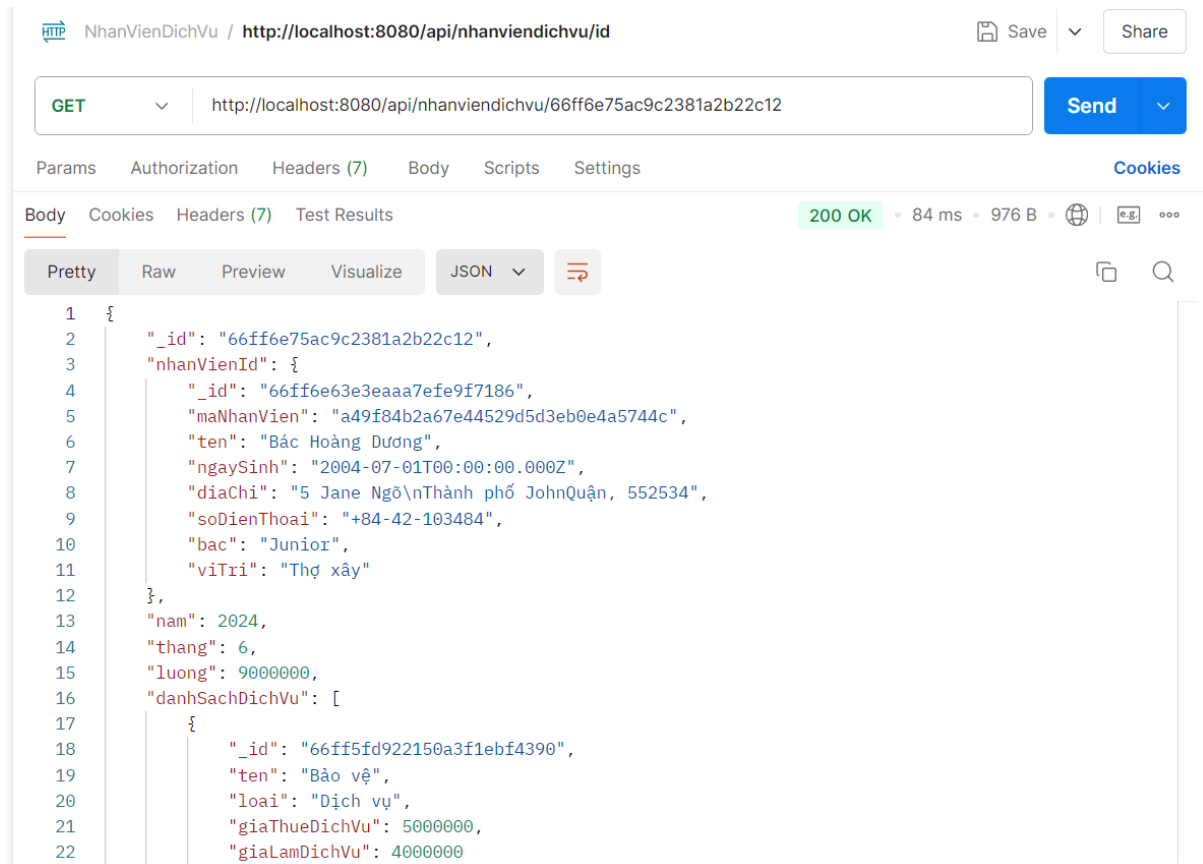
Mô Tả: API này được sử dụng để lấy thông tin của tất cả các nhân viên tòa nhà tham gia cung cấp các dịch vụ trong những tháng khác nhau. Thông tin bao gồm tên nhân viên, danh sách dịch vụ, năm, tháng và mức lương.



## 2. Lấy thông tin chi tiết của một nhân viên dịch vụ cụ thể dựa vào id

URL: GET <http://localhost:8080/api/nhanviendichvu/:id>

Mô Tả: API này cho phép người dùng truy cập vào thông tin chi tiết của một nhân viên dịch vụ cụ thể bằng cách cung cấp id của bản ghi đó. Thông tin bao gồm năm, tháng, lương, và các dịch vụ mà nhân viên tham gia.



### 3. Cập nhật thông tin của một nhân viên dịch vụ dựa vào id

URL: PUT `http://localhost:8080/api/nhanviendichvu/:id`

Mô Tả: API này cho phép người dùng cập nhật thông tin của một nhân viên dịch vụ đã tồn tại trong hệ thống. Người dùng cần cung cấp id của bản ghi và thông tin cần cập nhật (như lương, danh sách dịch vụ, v.v.).

PUT

http://localhost:8080/api/nhanviendichvu/66ff6e75ac9c2381a2b22c12

Send

Params

Authorization

Headers (9)

Body

Scripts

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   // "nhanVienId": "614b1c4a3f46d6d1c0a5c2b0", // ID của nhân viên
3   // "nam": 2023,
4   // "thang": 10,
5   "danhSachDichVu": [
6     "66ff5fd922150a3f1ebf4390" // ID của dịch vụ mới
7   ]
8 }
```

Body

Cookies

Headers (7)

Test Results

200 OK

157 ms

401 B

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "_id": "66ff6e75ac9c2381a2b22c12",
3   "nhanVienId": "66ff6e63e3eaaa7efe9f7186",
4   "nam": 2024,
5   "thang": 6,
6   "luong": 4000000,
7   "danhSachDichVu": [
8     "66ff5fd922150a3f1ebf4390"
9   ],
10  "__v": 1
11 }
```

#### 4. Thêm mới một nhân viên dịch vụ vào hệ thống

URL: POST <http://localhost:8080/api/nhanviendichvu>

Mô Tả: API này cho phép thêm mới một bản ghi về nhân viên dịch vụ vào collection NhanVienDichVu. Dữ liệu cần thiết để tạo bao gồm nhanVienId, nam, thang, luong, và danh sách dịch vụ mà nhân viên tham gia.

POST http://localhost:8080/api/nhanviendichvu

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "nhanVienId": "66ff6e63e3eaaa7efe9f7186", // ID của nhân viên
3   "nam": 2025,
4   "thang": 1,
5   "danhSachDichVu": [
6     "66ff5fd922150a3f1ebf438d", // ID của dịch vụ 1
7     "66ff5fd922150a3f1ebf438e" // ID của dịch vụ 2
8   ]
9 }
```

Body Cookies Headers (7) Test Results 200 OK • 99 ms • 428 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔍

```
1 {
2   "nhanVienId": "66ff6e63e3eaaa7efe9f7186",
3   "nam": 2025,
4   "thang": 1,
5   "luong": 7000000,
6   "danhSachDichVu": [
7     "66ff5fd922150a3f1ebf438d",
8     "66ff5fd922150a3f1ebf438e"
9   ],
10  "_id": "66ffc1cd40ff6c830d53971e",
11  "v": 0
12 }
```

## 5. Xóa một bản ghi nhân viên dịch vụ cụ thể dựa vào id

URL: DELETE http://localhost:8080/api/nhanviendichvu/:id

Mô Tả: API này được sử dụng để xóa thông tin của một nhân viên dịch vụ cụ thể khỏi hệ thống bằng cách cung cấp id của bản ghi.

HTTP NhanVienDichVu / http://localhost:8080/api/nhanviendichvu/id Save Share

DELETE http://localhost:8080/api/nhanviendichvu/66ffc1cd40ff6c830d53971e Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK • 44 ms • 291 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON 🔍

```
1 {
2   "message": "Bản ghi đã được xóa thành công!"
3 }
```