Here's a detailed API documentation for the employee_operations endpoint, which includes instructions for each method.

Employee API Documentation
Base URL: https://api.example.com/

Endpoint: /employees/

Authentication
Method: Bearer Token
Header:
Authorization: Bearer <YOUR_API_KEY>
Endpoints
1. Get Employee(s)
Method: GET
Endpoint: /employees/ or /employees/{id}/
Description
Retrieves employee details. If an id is provided, returns a specific employee's details; otherwise, returns a paginated list of employees.

URL Parameters
id (optional): Unique identifier of the employee.
Query Parameters
department (optional): Filters employees by department.
role (optional): Filters employees by role.
Response (Single Employee)
Success (200 OK):
json

```
{
  "id": 1,
  "name": "John Doe",
  "department": "IT",
  "role": "Developer"
}
```
Error (404 Not Found):
json
```
{
  "error": "Employee not found"
}
```
Response (Paginated List of Employees)
Success (200 OK):
json
```
{
  "count": 50,
  "next": "https://api.example.com/employees/?page=2",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "John Doe",
      "department": "IT",
      "role": "Developer"
    },
    ...
  ]
}
```
2. Create an Employee
Method: POST
Endpoint: /employees/
Description
Creates a new employee record.

Request Body
name (string, required): Name of the employee.
department (string, required): Department where the employee works.
role (string, required): Employee’ s role.
Example Request
json
```
{
  "name": "John Doe",
  "department": "IT",
  "role": "Developer"
}
```
Response
Success (201 Created):
json
```
{
  "message": "Employee details created successfully",
  "data": {
    "id": 1,
    "name": "John Doe",
    "department": "IT",
    "role": "Developer"
  }
}
```
Error (400 Bad Request):
json
```
{
  "name": ["This field is required."]
}
```
3. Update an Employee
Method: PUT
Endpoint: /employees/{id}/
Description
Updates details of an existing employee.

URL Parameters
id (required): Unique identifier of the employee.
Request Body
name (string, optional): Updated name.
department (string, optional): Updated department.
role (string, optional): Updated role.
Example Request
json
```
{
  "name": "Jane Doe",
  "department": "HR",
  "role": "Manager"
}
```
Response
Success (200 OK):

json
```
{
  "message": "Employee details updated successfully",
  "data": {
    "id": 1,
    "name": "Jane Doe",
    "department": "HR",
    "role": "Manager"
  }
}
```
Error (404 Not Found):
json
```
{
  "error": "Employee not found"
}
```
Error (400 Bad Request):
json
```
{
  "name": ["This field is required."]
}
```
4. Delete an Employee
Method: DELETE
Endpoint: /employees/{id}/
Description
Deletes an employee record.

URL Parameters
id (required): Unique identifier of the employee.
Response
Success (204 No Content):
json
```
{
  "message": "Employee deleted successfully"
}
```
Error (404 Not Found):
json
```
{
  "error": "Employee not found"
}
```
5. Filtering
Method:GET
Endpoint: /employees/??department=HR
Description
Filtering the employess
Query Parameters for Filtering
department - Filters employees based on their department.

Type: String (case-insensitive)
Example: ?department=IT
role - Filters employees based on their role.

Type: String (case-insensitive)
Example: ?role=Developer

6. Filter by Department
http://127.0.0.1:8000/api/employees/??department=HR

```
Success (200):
{
    "count": 4,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 1,
            "name": "mani",
            "email": "manikantakotla1998@gmail.com",
            "department": "Hr",
            "role": "Manager",
            "date_joined": "2024-10-31"
        },
    ]
}
```

7. Filter by Role

To retrieve a list of employees with the Manager role:

```
Success (200):
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 5,
            "name": "m",
            "email": "m1998@gmail.com",
            "department": "Hr",
            "role": "Manager",
            "date_joined": "2024-10-31"
        }
    ]
}
```

8)Pagination in Employee API

Endpoint: GET /employees/

Overview

The API uses Page Number Pagination, where results are divided into pages. The default page size is 10 records per page, but this may be adjusted if needed.

Query Parameters for Pagination

page - Specifies which page to retrieve.

Type: Integer

Example: ?page=2

```
Success (200):
{
    "count": 12,
    "next": "http://127.0.0.1:8000/api/employees/?page=2",
    "previous": null,
    "results": [
        {
            "id": 1,
            "name": "mani",
        "email": "manikantakotla1998@gmail.com",
        "department": "Hr",
        "role": "dev",
        "date_joined": "2024-10-31"
    },
    {
        "id": 2,
        "name": "mani",
        "email": "manikantakotlaa1998@gmail.com",
        "department": "Engineering",
        "role": "dev",
        "date_joined": "2024-10-31"
    },
    {
        "id": 3,
        "name": "manikotla",
        "email": "manikantakotlla1998@gmail.com",
        "department": "Sales",
        "role": "dev",
        "date_joined": "2024-10-31"
    },
    {
        "id": 4,
        "name": "n",
        "email": "manikantaotlaaq1998@gmail.com",
        "department": "Hr",
        "role": "dev",
        "date_joined": "2024-10-31"
    },
    {
        "id": 5,
        "name": "m",
        "email": "m1998@gmail.com",
        "department": "Engineering",
        "role": "Manager",
        "date_joined": "2024-10-31"
    },
    {
        "id": 6,
        "name": "k",
        "email": "k1998@gmail.com",
        "department": "Sales",
        "role": "Sales",
        "date_joined": "2024-10-31"
    },
    {
        "id": 7,
        "name": "a",
        "email": "a1998@gmail.com",
        "department":"Sales",
        "role": "Developer",
        "date_joined": "2024-10-31"
    },
    {
        "id": 8,
        "name": "k",
            "email": "h1998@gmail.com",
            "department": "HR",
            "role": "Analyst",
            "date_joined": "2024-10-31"
        },
        {
            "id": 9,
            "name": "p",
            "email": "p1998@gmail.com",
            "department": "HR",
            "role": "Analyst",
            "date_joined": "2024-10-31"
        },
        {
            "id": 10,
            "name": "p",
            "email": "km1998@gmail.com",
            "department": "HR",
            "role": "Analyst",
            "date_joined": "2024-10-31"
        }
    ]
}
```
Error Codes
400 Bad Request: Invalid or missing parameters in the request.
401 Unauthorized: Invalid or missing authentication token.
404 Not Found: Resource not found.
500 Internal Server Error: Unexpected server error.
This documentation provides a complete reference for using the employee_operations endpoint effectively, covering all supported HTTP methods and typical request/response examples.