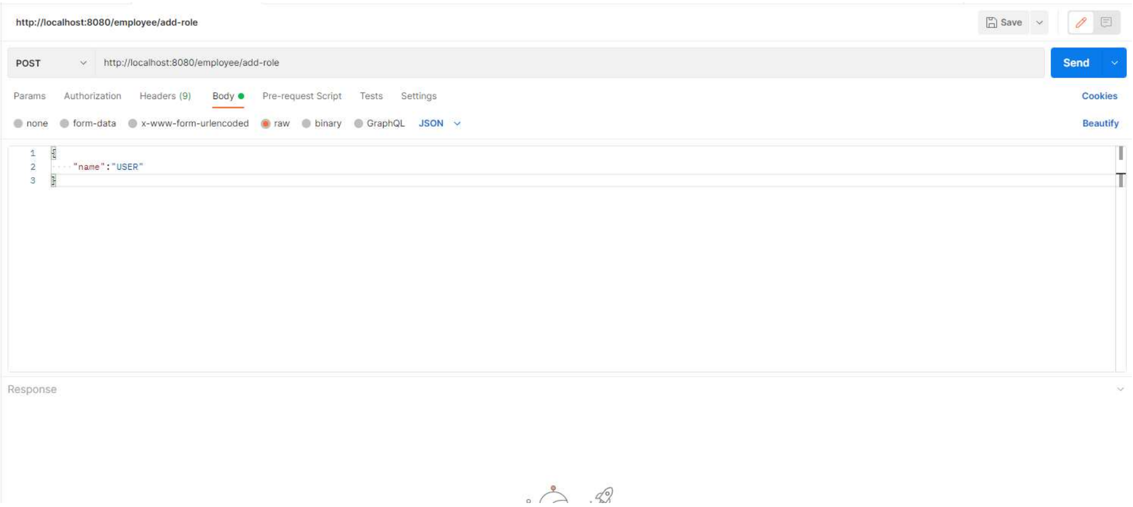
Employee Management Rest Api based Web application

Adding Roles to the database

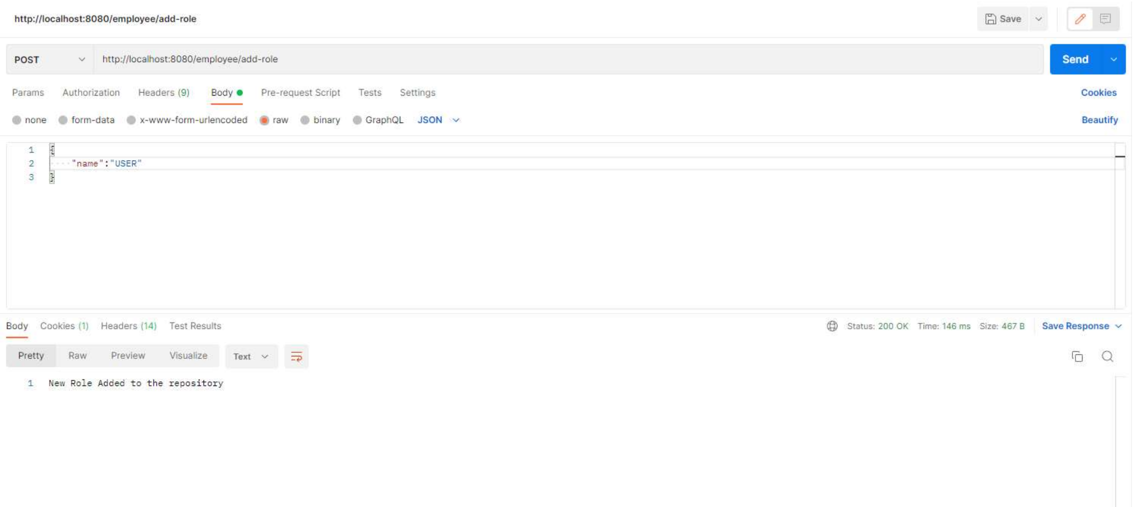


http://localhost:8080/employee/add-role

POST http://localhost:8080/employee/add-role

```
{
    "name":"USER"
}
```

Response from the application on adding the Role.



http://localhost:8080/employee/add-role

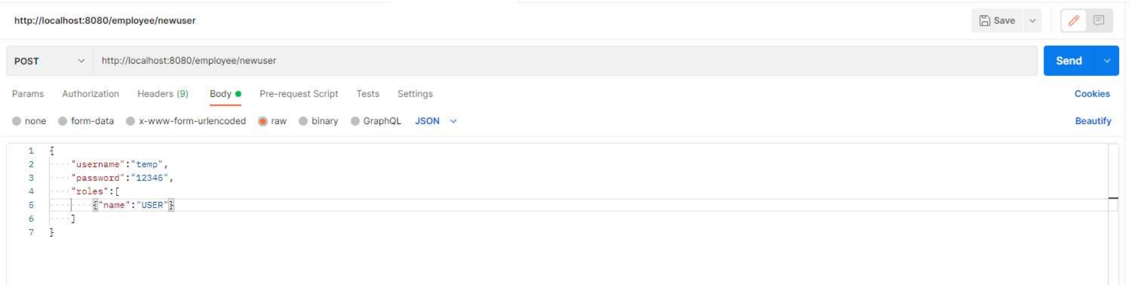
POST http://localhost:8080/employee/add-role

```
{
    "name":"USER"
}
```

Status: 200 OK Time: 146 ms Size: 467 B

```
New Role Added to the repository
```

Adding new User

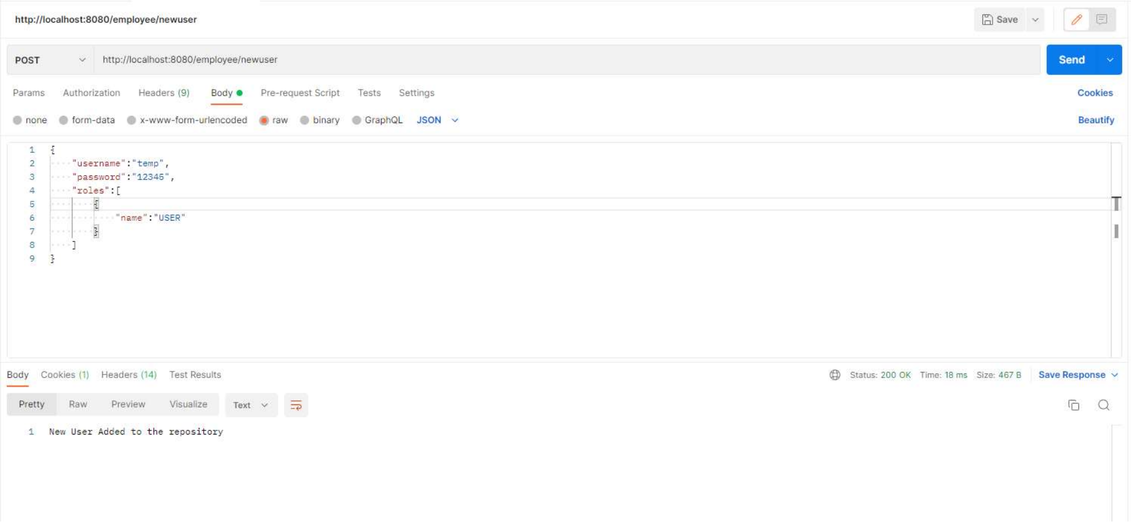


http://localhost:8080/employee/newuser

POST http://localhost:8080/employee/newuser

```
{
    "username":"temp",
    "password":"12345",
    "roles":[
        {"name":"USER"}
    ]
}
```

Response on adding new user



http://localhost:8080/employee/newuser

POST http://localhost:8080/employee/newuser

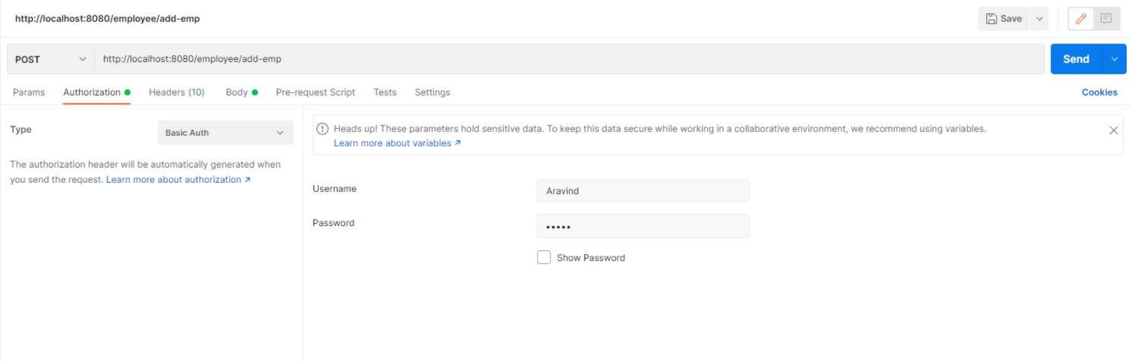
```
{
    "username":"temp",
    "password":"12345",
    "roles":[
        {
            "name":"USER"
        }
    ]
}
```

Status: 200 OK Time: 18 ms Size: 467 B

```
New User Added to the repository
```

Working with Application as Admin.

Setting up Authorization



http://localhost:8080/employee/add-emp

POST http://localhost:8080/employee/add-emp

Type: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

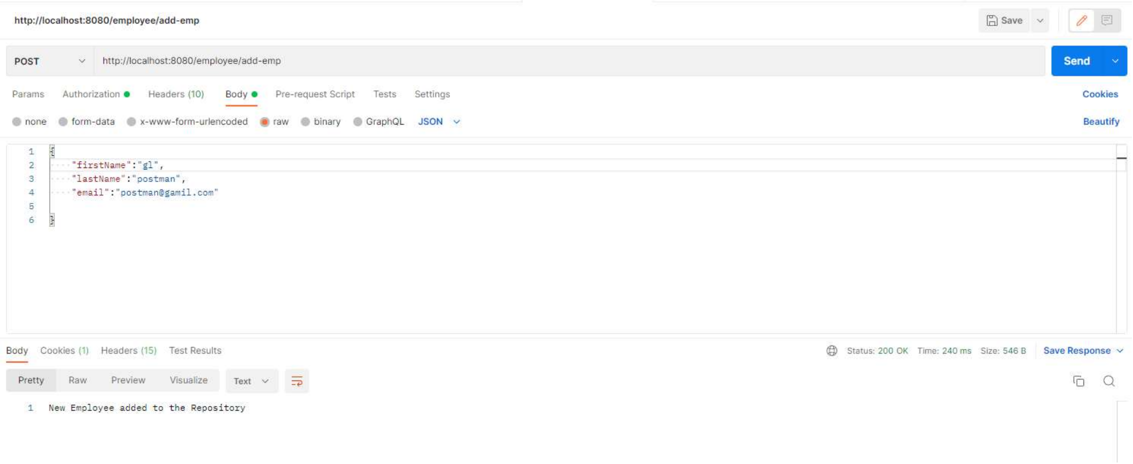
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Username: Aravind

Password: •••••

□ Show Password

Adding new Employee data



http://localhost:8080/employee/add-emp

POST http://localhost:8080/employee/add-emp

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

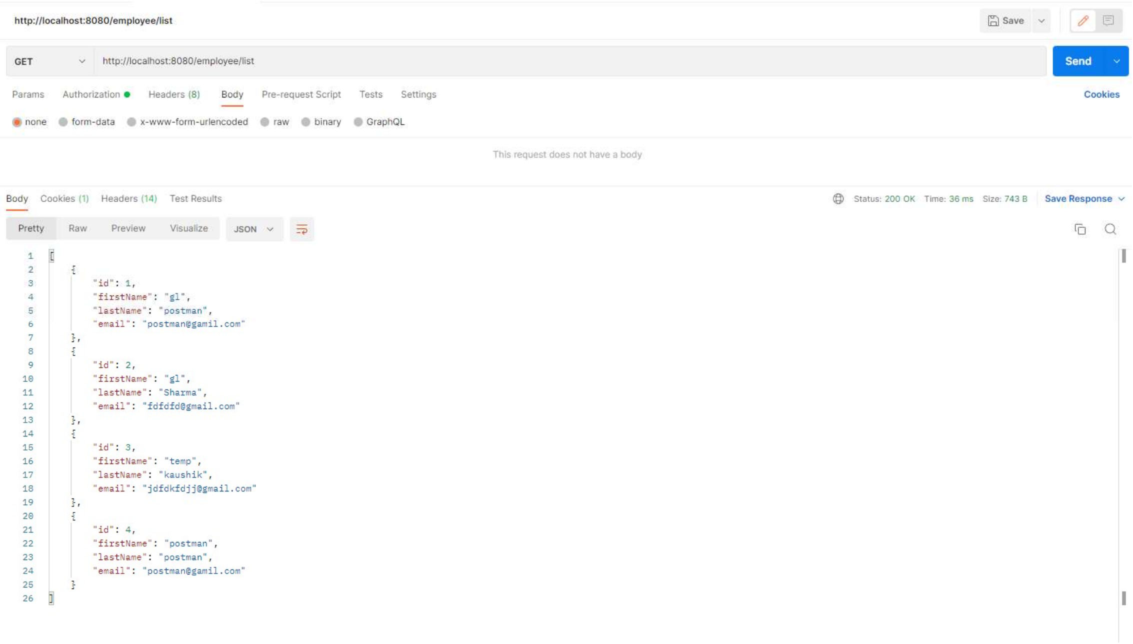
```
{
    "firstName":"gl",
    "lastName":"postman",
    "email":"postman@gamil.com"

}
```

Status: 200 OK Time: 240 ms Size: 546 B

Text

```
New Employee added to the Repository
```

List all Employee Data



http://localhost:8080/employee/list

GET http://localhost:8080/employee/list

● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

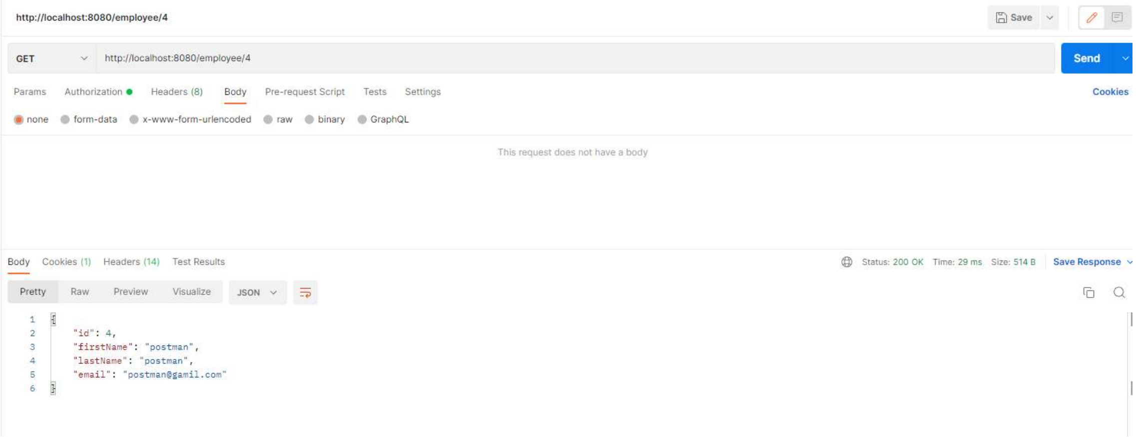
This request does not have a body

Status: 200 OK Time: 36 ms Size: 743 B

JSON

```
[
    {
        "id": 1,
        "firstName": "gl",
        "lastName": "postman",
        "email": "postman@gamil.com"
    },
    {
        "id": 2,
        "firstName": "gl",
        "lastName": "Sharma",
        "email": "fdfdfd@gmail.com"
    },
    {
        "id": 3,
        "firstName": "temp",
        "lastName": "kaushik",
        "email": "jdfdkfdjj@gmail.com"
    },
    {
        "id": 4,
        "firstName": "postman",
        "lastName": "postman",
        "email": "postman@gamil.com"
    }
]
```

Fetching Employee record specifically based on the id of the employee



http://localhost:8080/employee/4

GET http://localhost:8080/employee/4

● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

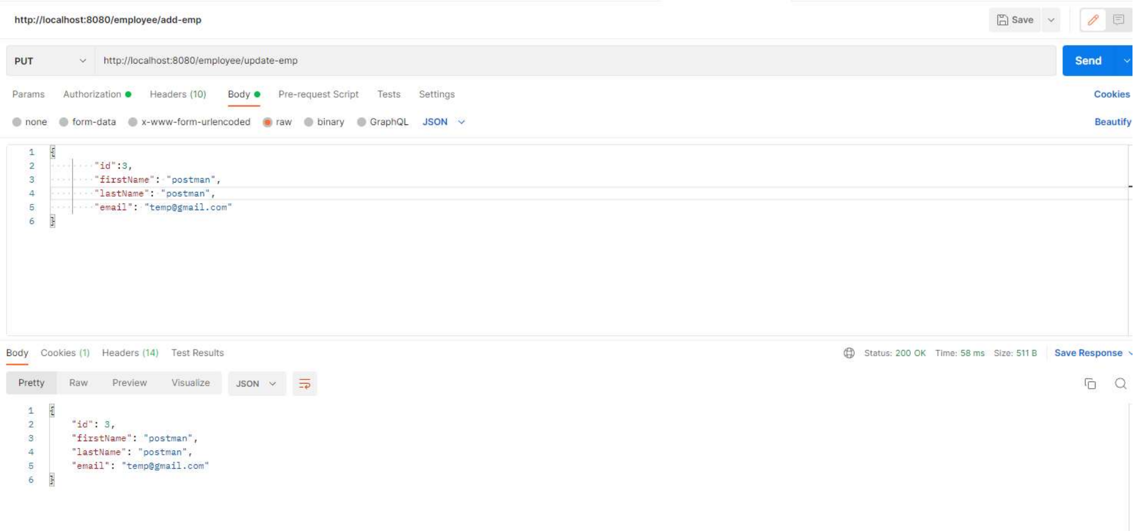
This request does not have a body

Status: 200 OK Time: 29 ms Size: 514 B

JSON

```
{
    "id": 4,
    "firstName": "postman",
    "lastName": "postman",
    "email": "postman@gamil.com"
}
```

Updating Existing employee record with the given updated json object.



http://localhost:8080/employee/add-emp

PUT http://localhost:8080/employee/update-emp

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

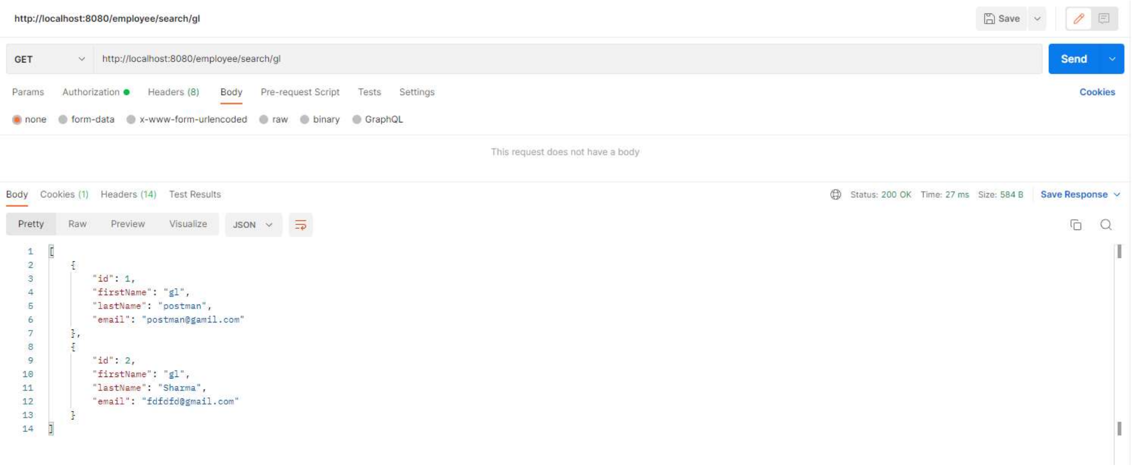
```
{
    "id":3,
    "firstName": "postman",
    "lastName": "postman",
    "email": "temp@gmail.com"
}
```

Status: 200 OK Time: 58 ms Size: 511 B

JSON

```
{
    "id": 3,
    "firstName": "postman",
    "lastName": "postman",
    "email": "temp@gmail.com"
}
```

Find the Employee record by first name



http://localhost:8080/employee/search/gl

GET http://localhost:8080/employee/search/gl

● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

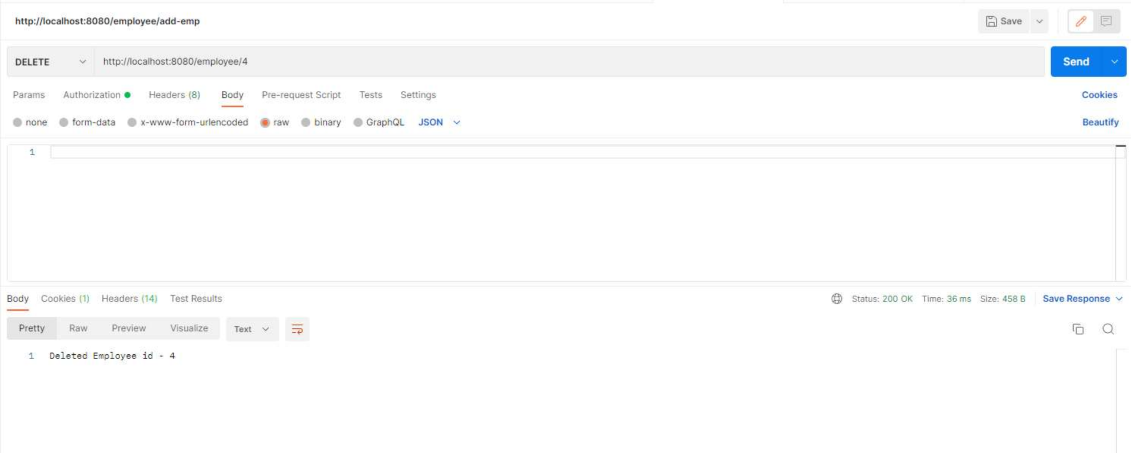
This request does not have a body

Status: 200 OK Time: 27 ms Size: 584 B

JSON

```
[
    {
        "id": 1,
        "firstName": "gl",
        "lastName": "postman",
        "email": "postman@gamil.com"
    },
    {
        "id": 2,
        "firstName": "gl",
        "lastName": "Sharma",
        "email": "fdfdfd@gmail.com"
    }
]
```

deleting an existing employee record based on the id of the employee.



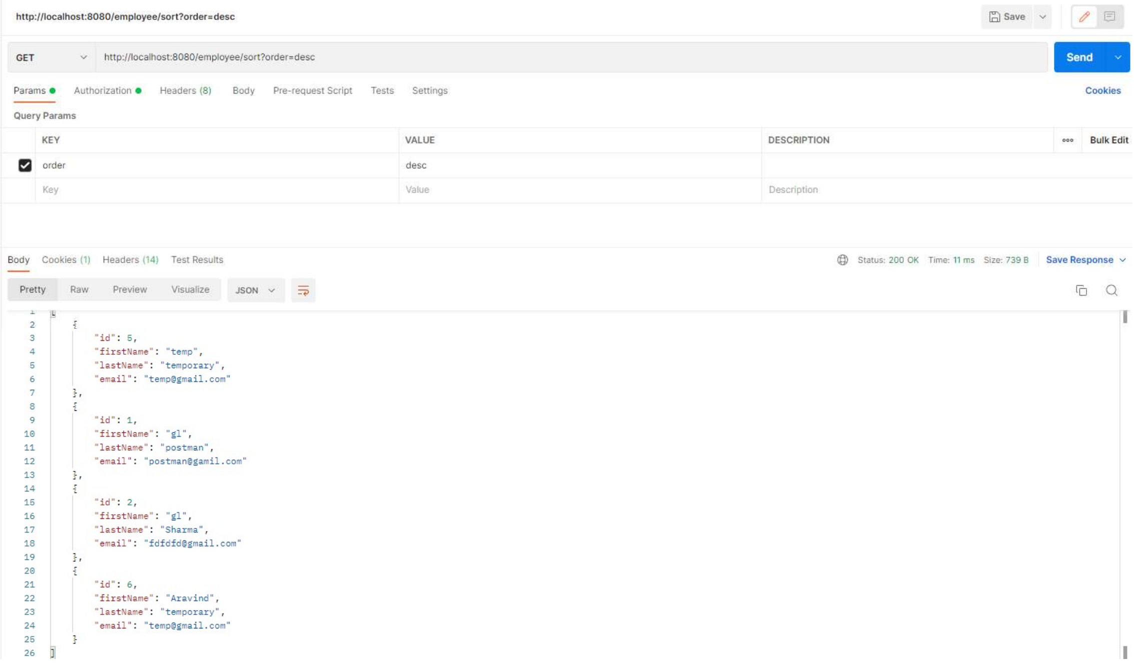
http://localhost:8080/employee/add-emp

DELETE http://localhost:8080/employee/4

Status: 200 OK Time: 36 ms Size: 458 B

```
Deleted Employee id - 4
```

list all employee records sorted on their first name in either ascending order or descending order



http://localhost:8080/employee/sort?order=desc

GET http://localhost:8080/employee/sort?order=desc

Query Params

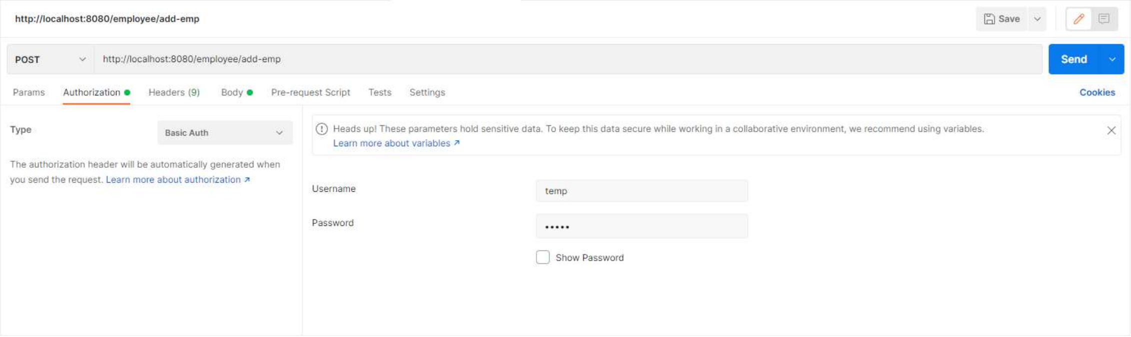
	KEY	VALUE	DESCRIPTION
☑	order	desc	
	Key	Value	Description

Status: 200 OK Time: 11 ms Size: 739 B

```
[
    {
        "id": 5,
        "firstName": "temp",
        "lastName": "temporary",
        "email": "temp@gmail.com"
    },
    {
        "id": 1,
        "firstName": "gl",
        "lastName": "postman",
        "email": "postman@gamil.com"
    },
    {
        "id": 2,
        "firstName": "gl",
        "lastName": "Sharma",
        "email": "fdfdfd@gmail.com"
    },
    {
        "id": 6,
        "firstName": "Aravind",
        "lastName": "temporary",
        "email": "temp@gmail.com"
    }
]
```

Working with Application as User.

Setting up authorization



http://localhost:8080/employee/add-emp

POST http://localhost:8080/employee/add-emp

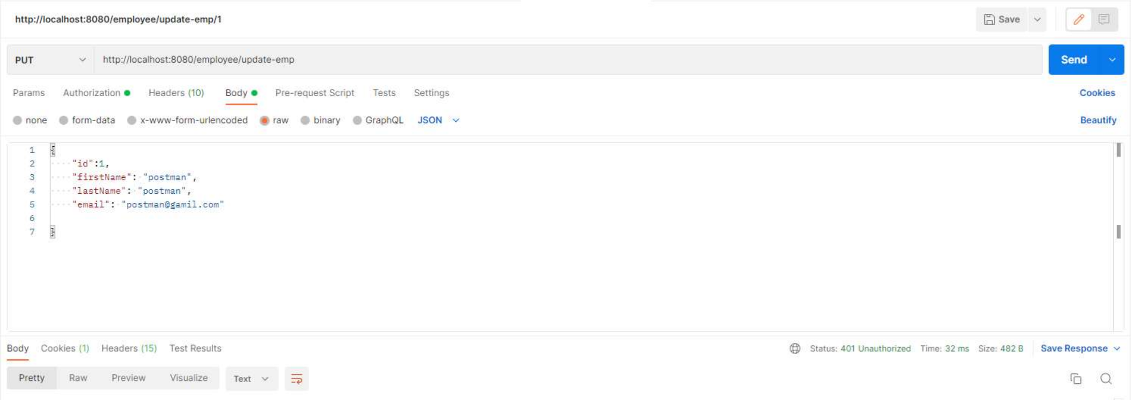
Type: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Username: temp

Password: •••••



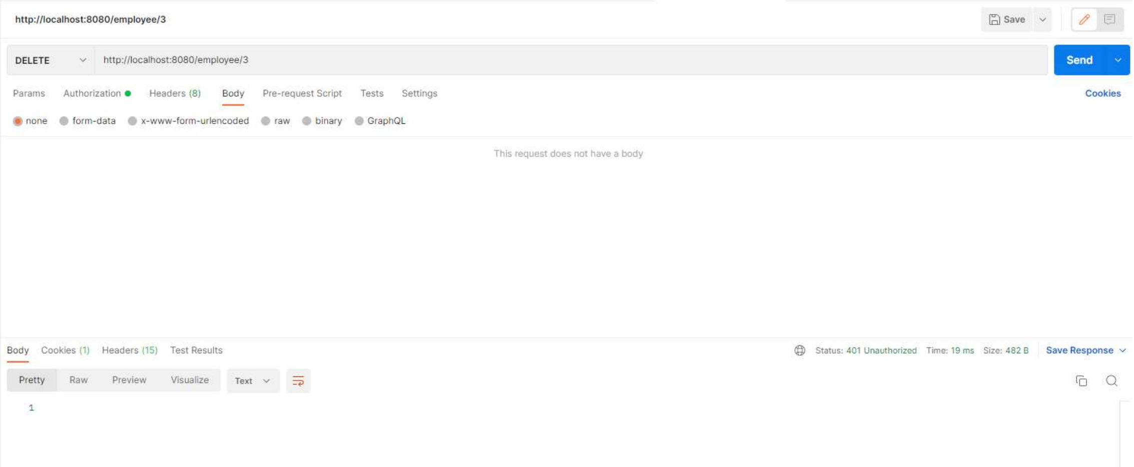
http://localhost:8080/employee/update-emp/1

PUT http://localhost:8080/employee/update-emp

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

```
{
    "id":1,
    "firstName": "postman",
    "lastName": "postman",
    "email": "postman@gamil.com"

}
```

Status: 401 Unauthorized Time: 32 ms Size: 482 B



http://localhost:8080/employee/3

DELETE http://localhost:8080/employee/3

● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

Status: 401 Unauthorized Time: 19 ms Size: 482 B

```
1
```

Receiving unauthorized message for user on functions add new employee record, update employee record, and delete the employee record.