Project Screen Shots

(Graded Coding Assignment 6)

Employee Management Rest API based Web application performs CRUD functionality with security.

About the project:	
Project Type:	Rest API based Web application
Application Default Username:	Admin
Default User Role:	ADMIN
Default User Password:	123456
RDBMS used:	H2
H2-Console Username:	admin
H2-Console Password:	
Application default port:	8080

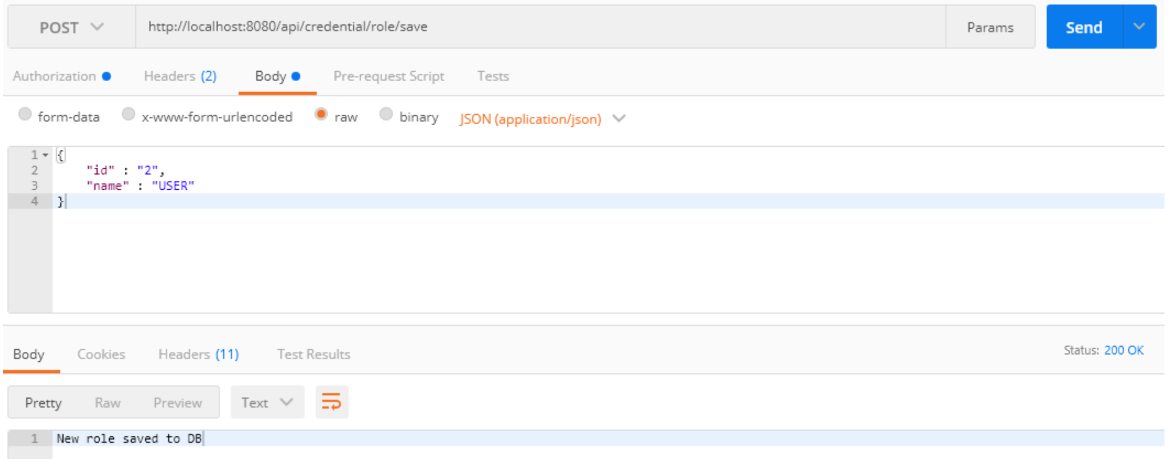
Screen recording:
https://drive.google.com/file/d/1e4b-kdgpNvjS8cWuPYSgcb8ION4rtjoB/view?usp=sharing

Authorizations Table	
Role: ADMIN	**Role: USER**
Credential Management: - Save new user - Update existing user - Save new role Employee Management: - Save new employee - Update existing employee data - Get employee by id - Delete Employee by id - List all employees - List employees by first name - List employees by sorted on first name – ASC & DESC	Credential Management: - Nil Employee Management: - Get employee by id - List all employees - List employees by first name - List employees by sorted on first name – ASC & DESC

Screen Shots:

Adding new Role

Option: Adding new role	Role: ADMIN	Status: Success



POST http://localhost:8080/api/credential/role/save

```
{
    "id" : "2",
    "name" : "USER"
}
```

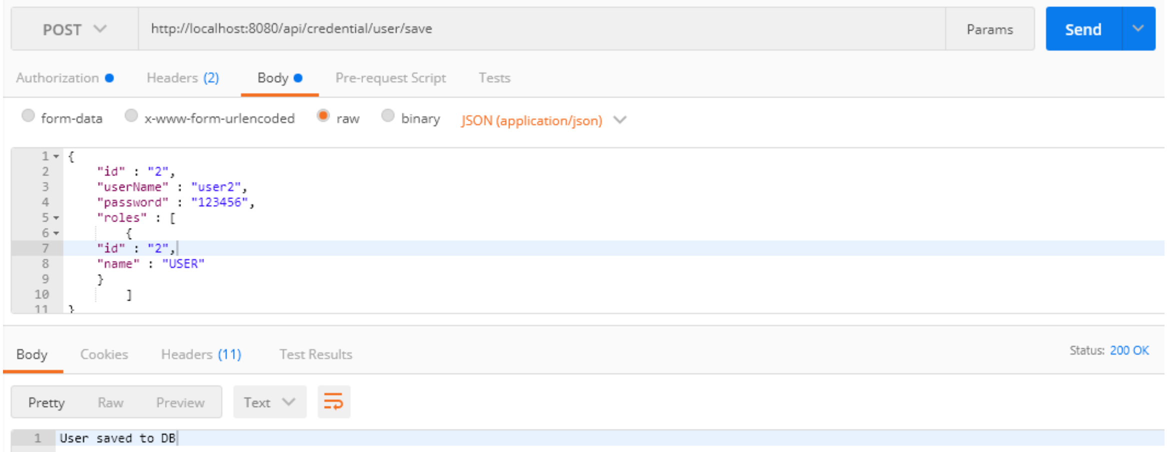
Status: 200 OK

```
New role saved to DB
```

Role ‘ADMIN’ only has authorization to add new role.

Adding new User

Option: Adding new user	Role: ADMIN	Status: Success



http://localhost:8080/api/credential/user/save

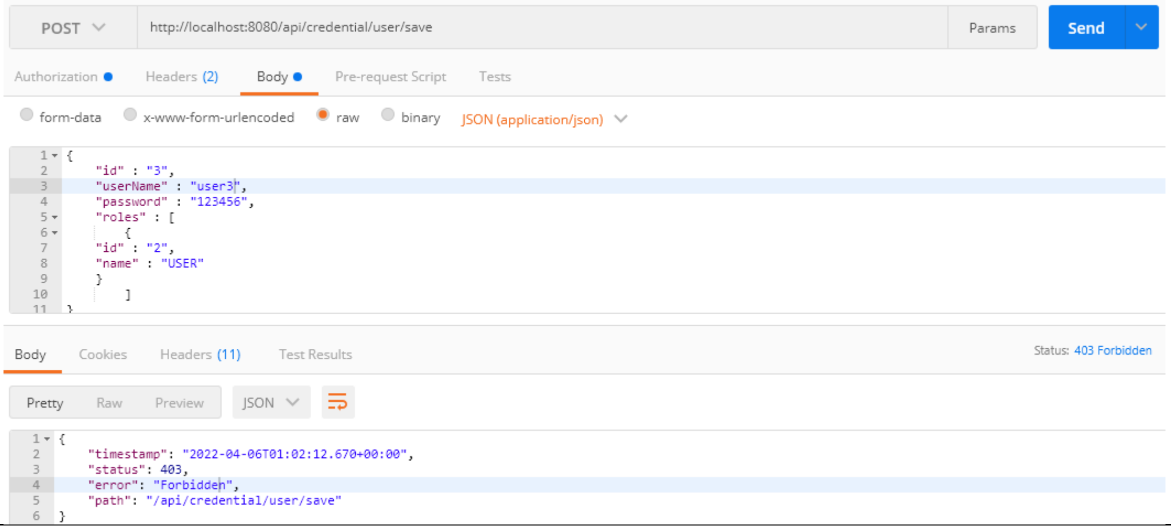
```
{
    "id" : "2",
    "userName" : "user2",
    "password" : "123456",
    "roles" : [
        {
    "id" : "2",
    "name" : "USER"
    }
        ]
}
```

Status: 200 OK

```
User saved to DB
```

Role ‘ADMIN’ only has authorization to add new user.

Option: Adding new user	Role: USER	Status: Forbidden



http://localhost:8080/api/credential/user/save

```
{
    "id" : "3",
    "userName" : "user3",
    "password" : "123456",
    "roles" : [
        {
    "id" : "2",
    "name" : "USER"
    }
        ]
}
```

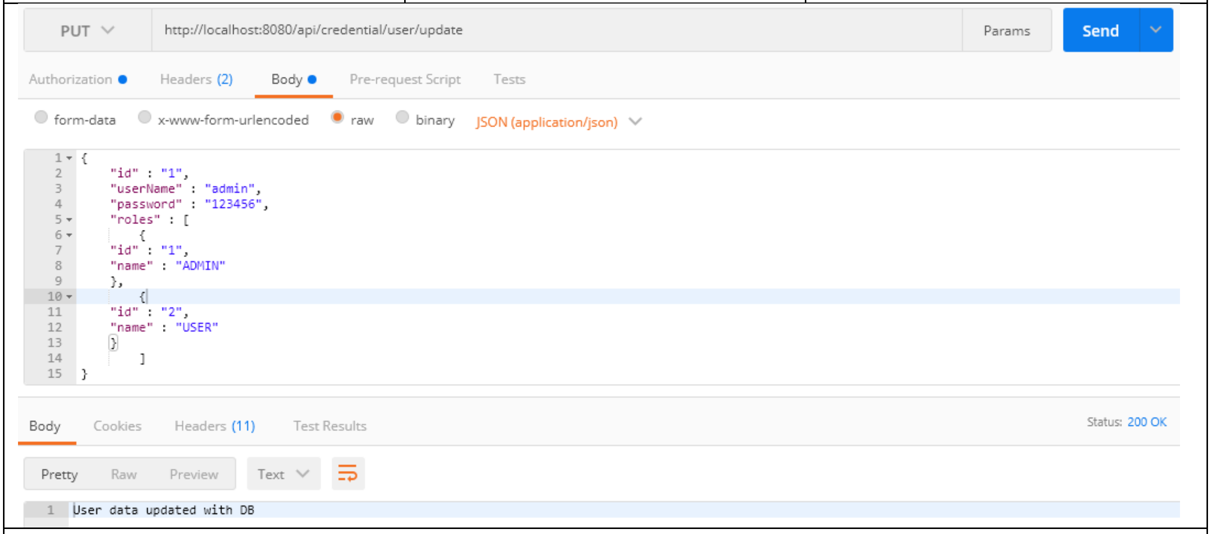
Status: 403 Forbidden

```
{
    "timestamp": "2022-04-06T01:02:12.670+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/api/credential/user/save"
}
```

Adding new user feature is forbidden for role: USER

Update user data

Option: Update user data	Role: ADMIN	Status: Success



PUT http://localhost:8080/api/credential/user/update

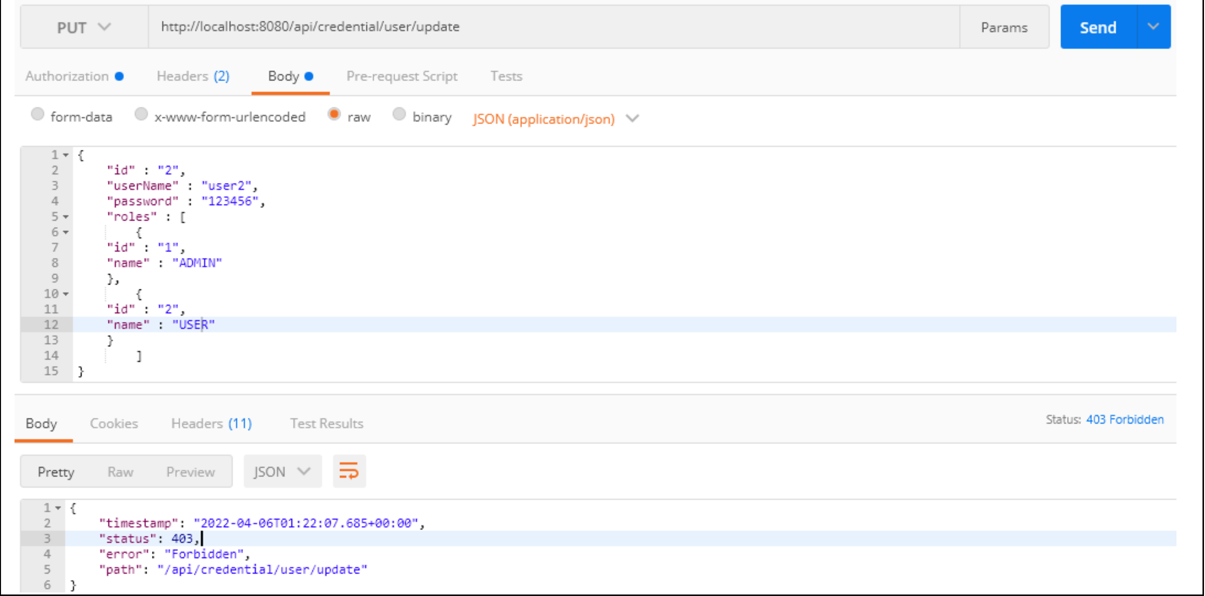
```
{
    "id" : "1",
    "userName" : "admin",
    "password" : "123456",
    "roles" : [
        {
    "id" : "1",
    "name" : "ADMIN"
    },
        {
    "id" : "2",
    "name" : "USER"
    }
        ]
}
```

Status: 200 OK

```
User data updated with DB
```

Role 'ADMIN' only has authorization to update user data.

Option: Update user data	Role: USER	Status: Forbidden



PUT http://localhost:8080/api/credential/user/update

```
{
    "id" : "2",
    "userName" : "user2",
    "password" : "123456",
    "roles" : [
        {
    "id" : "1",
    "name" : "ADMIN"
    },
        {
    "id" : "2",
    "name" : "USER"
    }
        ]
}
```

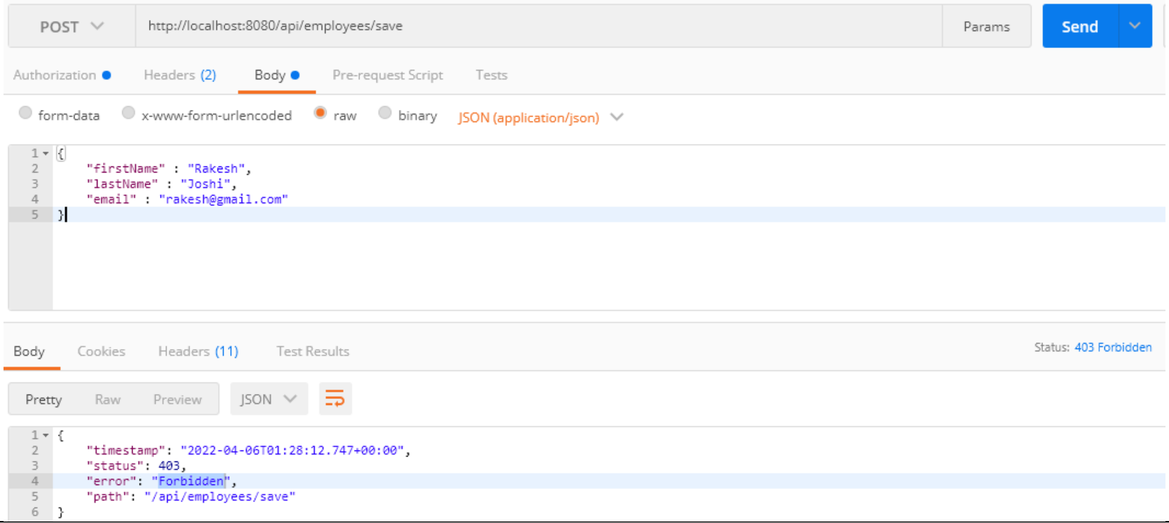
Status: 403 Forbidden

```
{
    "timestamp": "2022-04-06T01:22:07.685+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/api/credential/user/update"
}
```

Role 'ADMIN' only has authorization to update user data.

Adding new Employee

Option: Adding new employee	Role: USER	Status: Forbidden



POST http://localhost:8080/api/employees/save

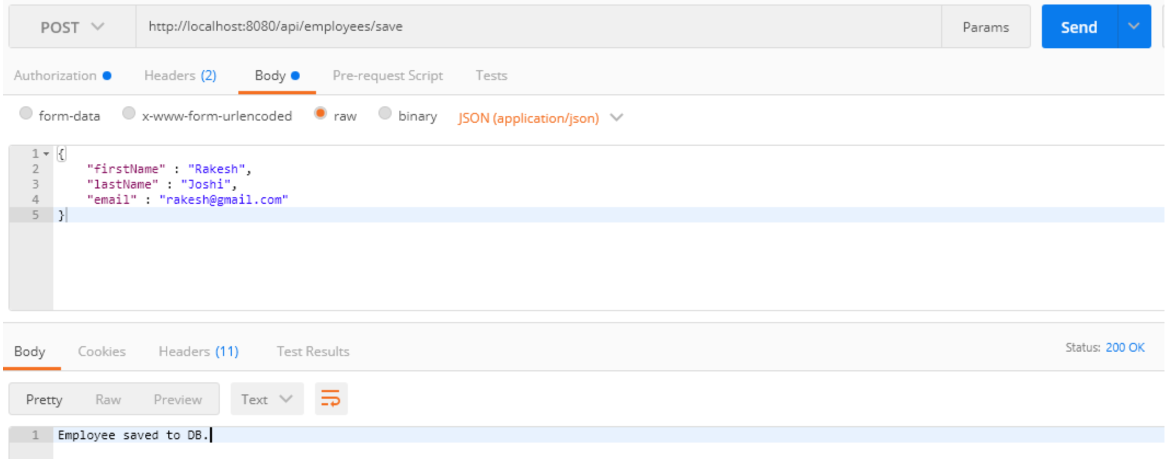
```
{
    "firstName" : "Rakesh",
    "lastName" : "Joshi",
    "email" : "rakesh@gmail.com"
}
```

Status: 403 Forbidden

```
{
    "timestamp": "2022-04-06T01:28:12.747+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/api/employees/save"
}
```

Role ‘ADMIN’ only has authorization to add new employee.

Option: Adding new employee	Role: ADMIN	Status: Success



POST http://localhost:8080/api/employees/save

```
{
    "firstName" : "Rakesh",
    "lastName" : "Joshi",
    "email" : "rakesh@gmail.com"
}
```

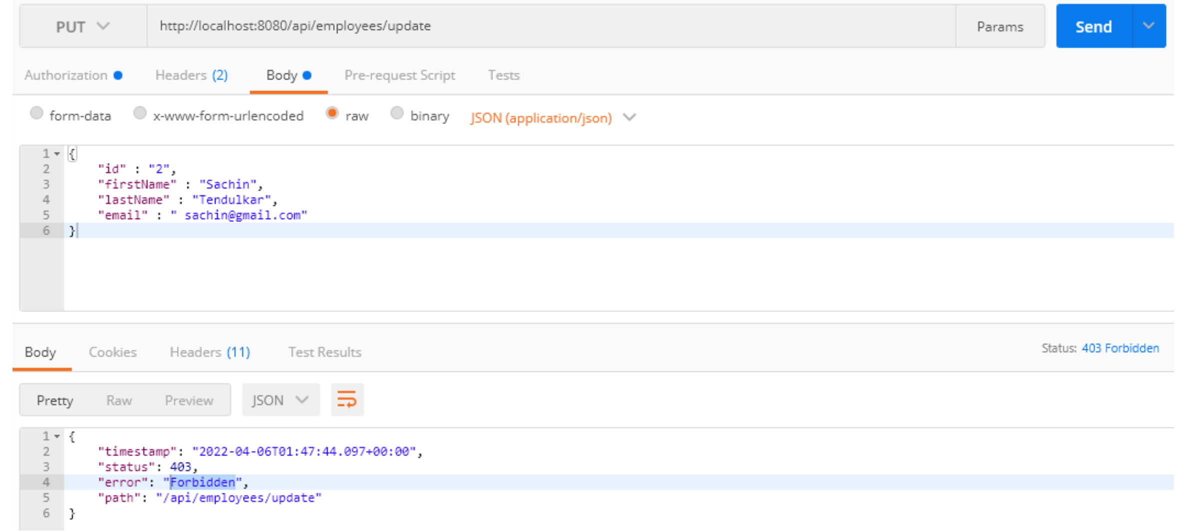
Status: 200 OK

```
Employee saved to DB.
```

Role ‘ADMIN’ only has authorization to add new employee.

Update Employee Data

Option: Update Employee Data	Role: USER	Status: Forbidden



http://localhost:8080/api/employees/update

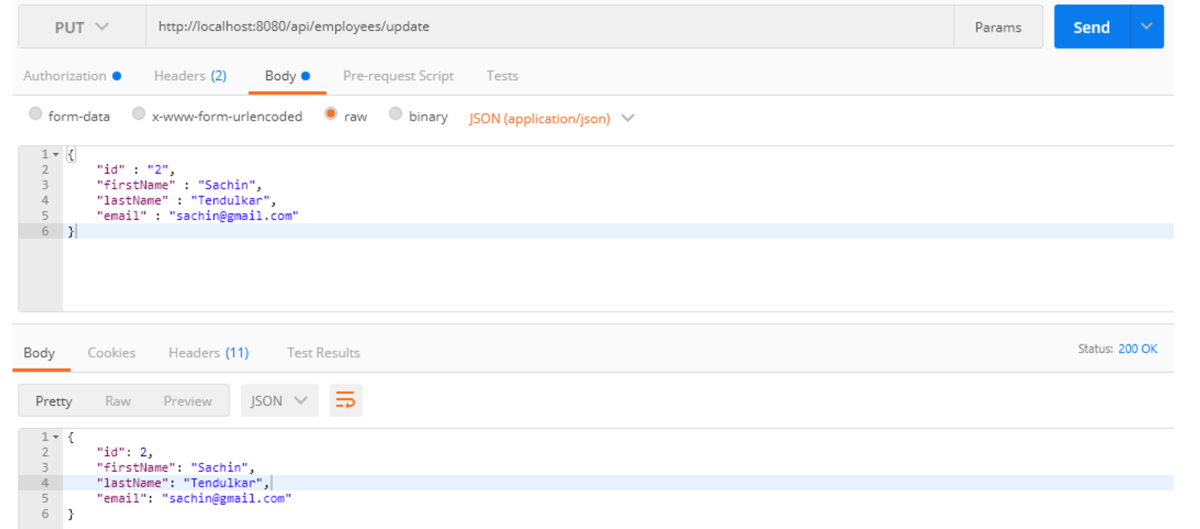
```
{
    "id" : "2",
    "firstName" : "Sachin",
    "lastName" : "Tendulkar",
    "email" : " sachin@gmail.com"
}
```

Status: 403 Forbidden

```
{
    "timestamp": "2022-04-06T01:47:44.097+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/api/employees/update"
}
```

Role ‘ADMIN’ only has authorization to update employee data.

Option: Update Employee Data	Role: ADMIN	Status: Success



http://localhost:8080/api/employees/update

```
{
    "id" : "2",
    "firstName" : "Sachin",
    "lastName" : "Tendulkar",
    "email" : "sachin@gmail.com"
}
```

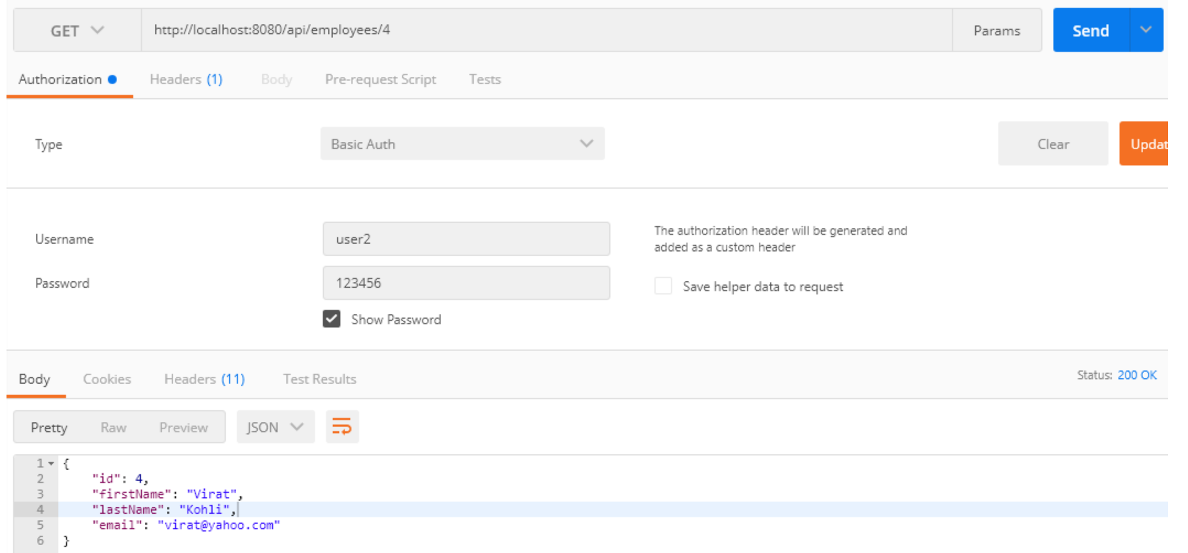
Status: 200 OK

```
{
    "id": 2,
    "firstName": "Sachin",
    "lastName": "Tendulkar",
    "email": "sachin@gmail.com"
}
```

Role ‘ADMIN’ only has authorization to update employee data.

Get Employee by ID

<u>Option</u>: Get Employee by ID	<u>Role</u>: USER	<u>Status</u>: Success



GET http://localhost:8080/api/employees/4

Type: Basic Auth

Username: user2

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

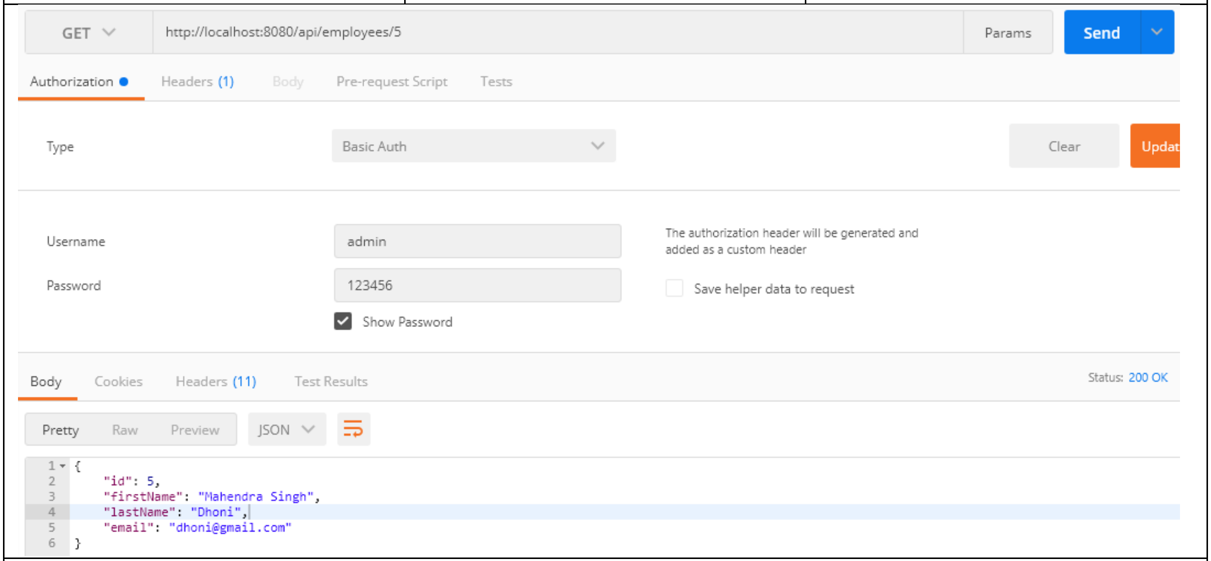
□ Save helper data to request

Status: 200 OK

```
{
    "id": 4,
    "firstName": "Virat",
    "lastName": "Kohli",
    "email": "virat@yahoo.com"
}
```

Both roles ‘ADMIN’ & ‘USER’ has authorization to get Employee by ID

<u>Option</u>: Get Employee by ID	<u>Role</u>: ADMIN	<u>Status</u>: Success



GET http://localhost:8080/api/employees/5

Type: Basic Auth

Username: admin

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

□ Save helper data to request

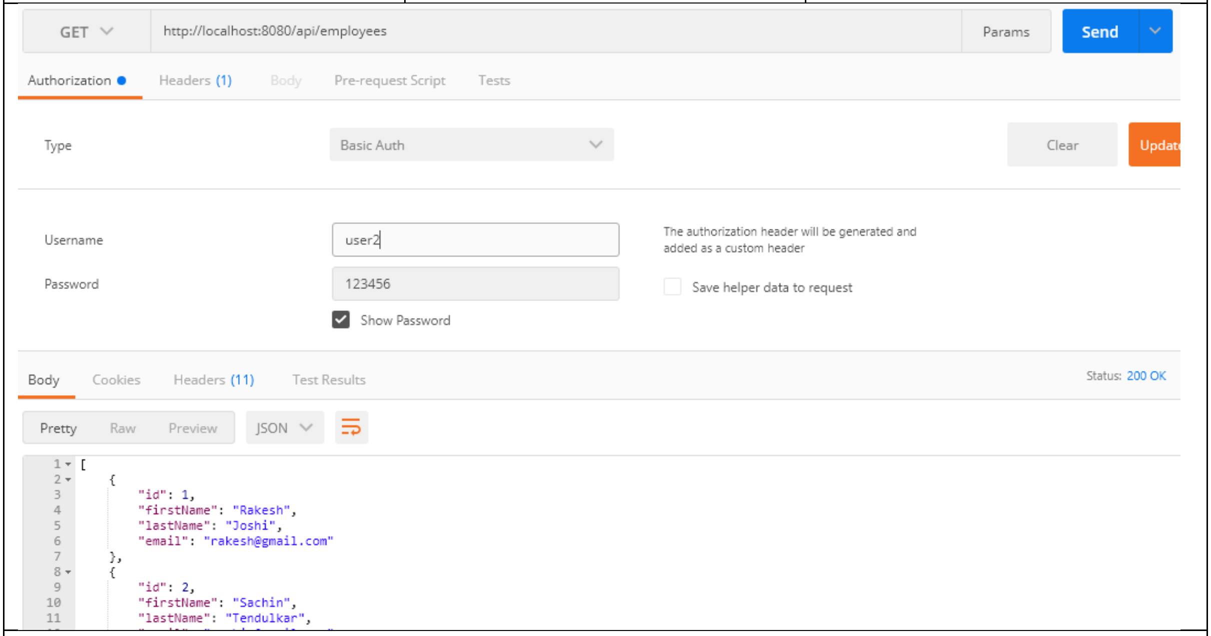
Status: 200 OK

```
{
    "id": 5,
    "firstName": "Mahendra Singh",
    "lastName": "Dhoni",
    "email": "dhoni@gmail.com"
}
```

Both roles ‘ADMIN’ & ‘USER’ has authorization to get Employee by ID

Listing all the Employees

Option: List all Employees	Role: USER	Status: Success



GET http://localhost:8080/api/employees

Type: Basic Auth

Username: user2

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

□ Save helper data to request

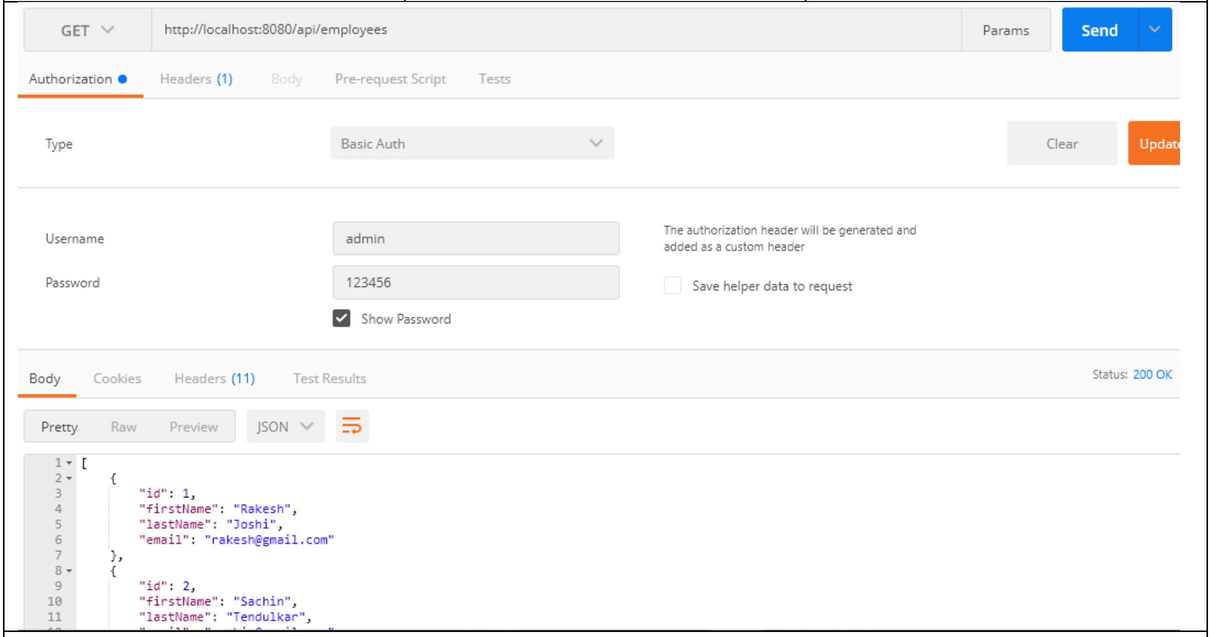
Status: 200 OK

JSON

```
[
    {
        "id": 1,
        "firstName": "Rakesh",
        "lastName": "Joshi",
        "email": "rakesh@gmail.com"
    },
    {
        "id": 2,
        "firstName": "Sachin",
        "lastName": "Tendulkar",
```

Both roles ‘ADMIN’ & ‘USER’ has authorization to List all Employees

Option: List all Employees	Role: ADMIN	Status: Success



GET http://localhost:8080/api/employees

Type: Basic Auth

Username: admin

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

□ Save helper data to request

Status: 200 OK

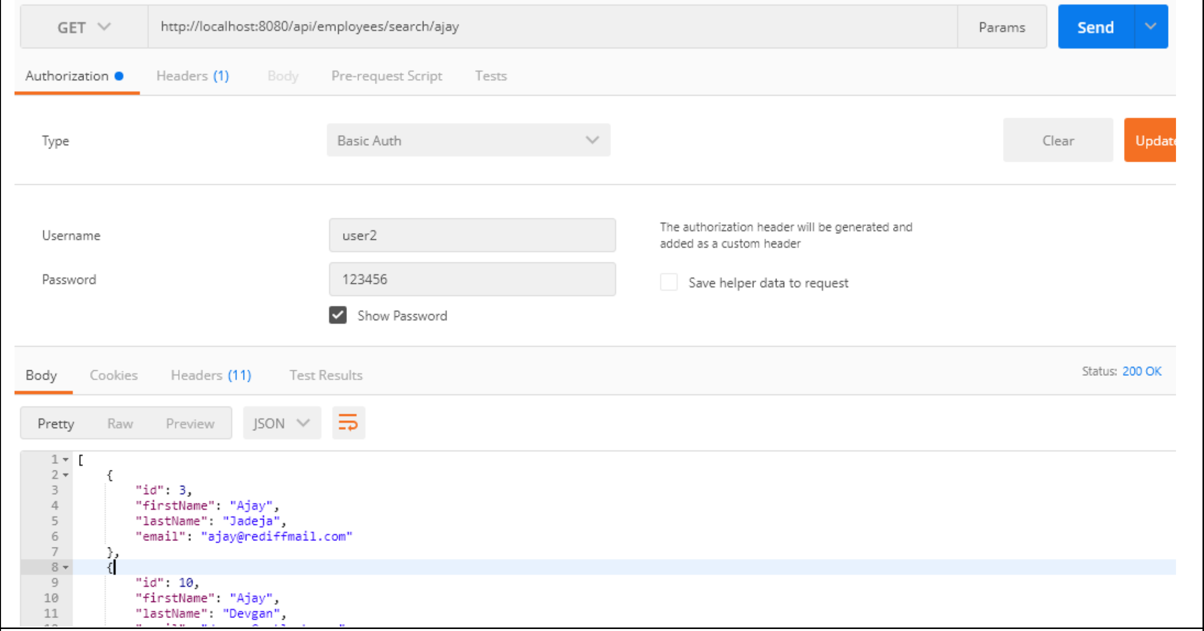
JSON

```
[
    {
        "id": 1,
        "firstName": "Rakesh",
        "lastName": "Joshi",
        "email": "rakesh@gmail.com"
    },
    {
        "id": 2,
        "firstName": "Sachin",
        "lastName": "Tendulkar",
```

Both roles ‘ADMIN’ & ‘USER’ has authorization to List all Employees

List all Employees by First Name

Option: List Employees by First Name	Role: USER	Status: Success



http://localhost:8080/api/employees/search/ajay

Type: Basic Auth

Username: user2

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

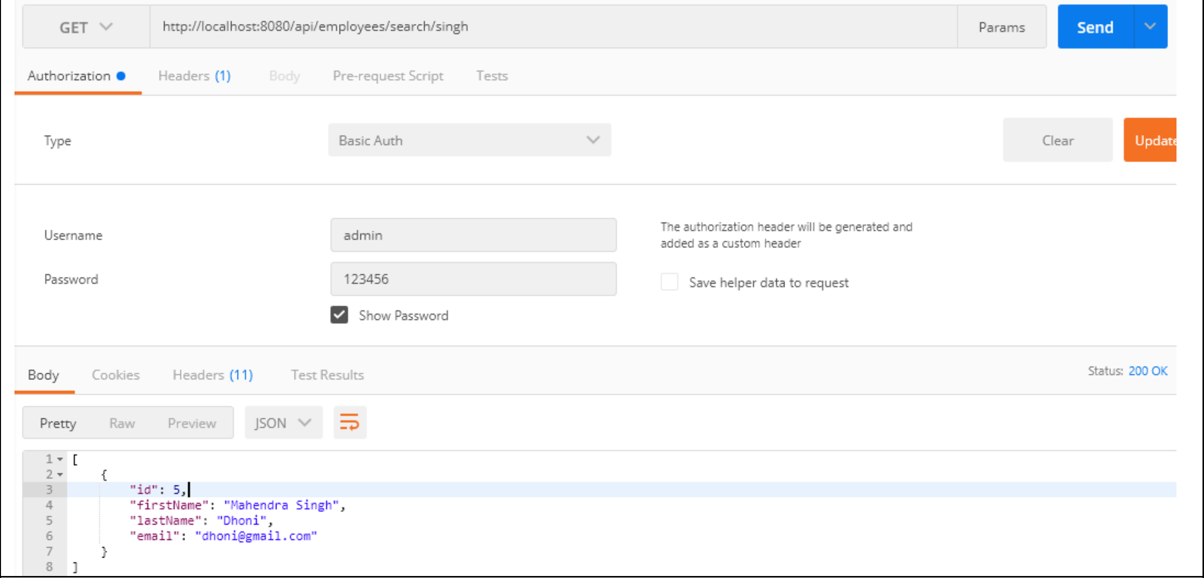
□ Save helper data to request

Status: 200 OK

```
[
    {
        "id": 3,
        "firstName": "Ajay",
        "lastName": "Jadeja",
        "email": "ajay@rediffmail.com"
    },
    {
        "id": 10,
        "firstName": "Ajay",
        "lastName": "Devgan",
```

Both roles 'ADMIN' & 'USER' has authorization to List Employees by First Name

Option: List Employees by First Name	Role: ADMIN	Status: Success



http://localhost:8080/api/employees/search/singh

Type: Basic Auth

Username: admin

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

□ Save helper data to request

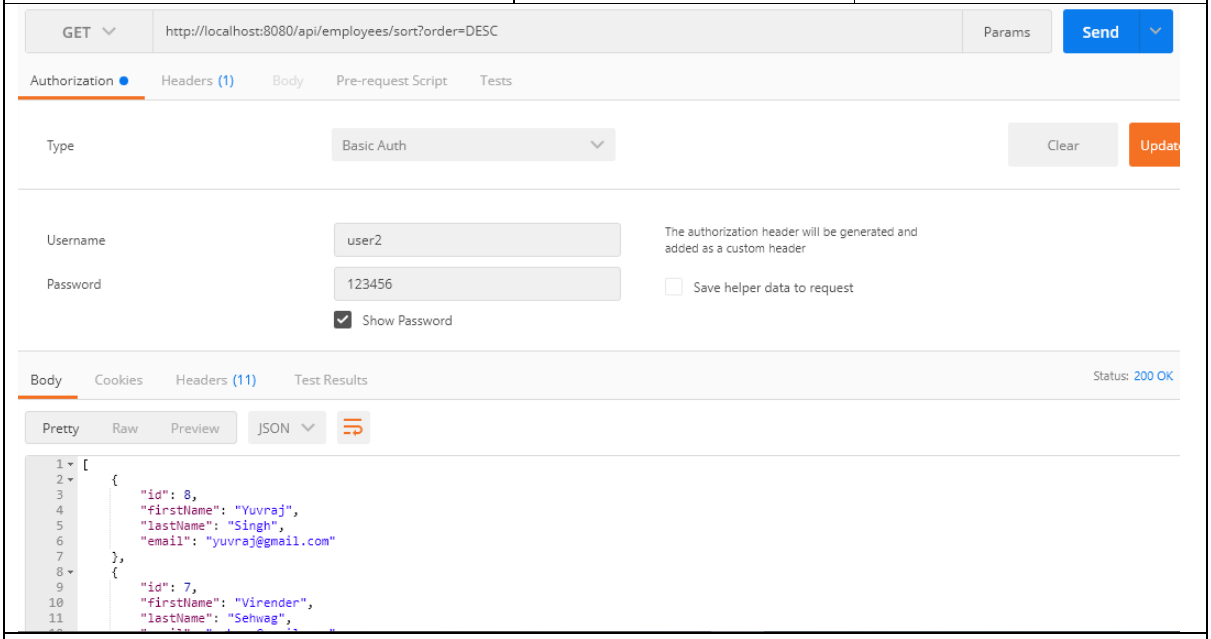
Status: 200 OK

```
[
    {
        "id": 5,
        "firstName": "Mahendra Singh",
        "lastName": "Dhoni",
        "email": "dhoni@gmail.com"
    }
]
```

Both roles 'ADMIN' & 'USER' has authorization to List Employees by First Name

List all Employees Sorted – ASC & DESC

Option: List all Employees Sorted	Role: USER	Status: Success



GET http://localhost:8080/api/employees/sort?order=DESC

Type: Basic Auth

Username: user2

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

□ Save helper data to request

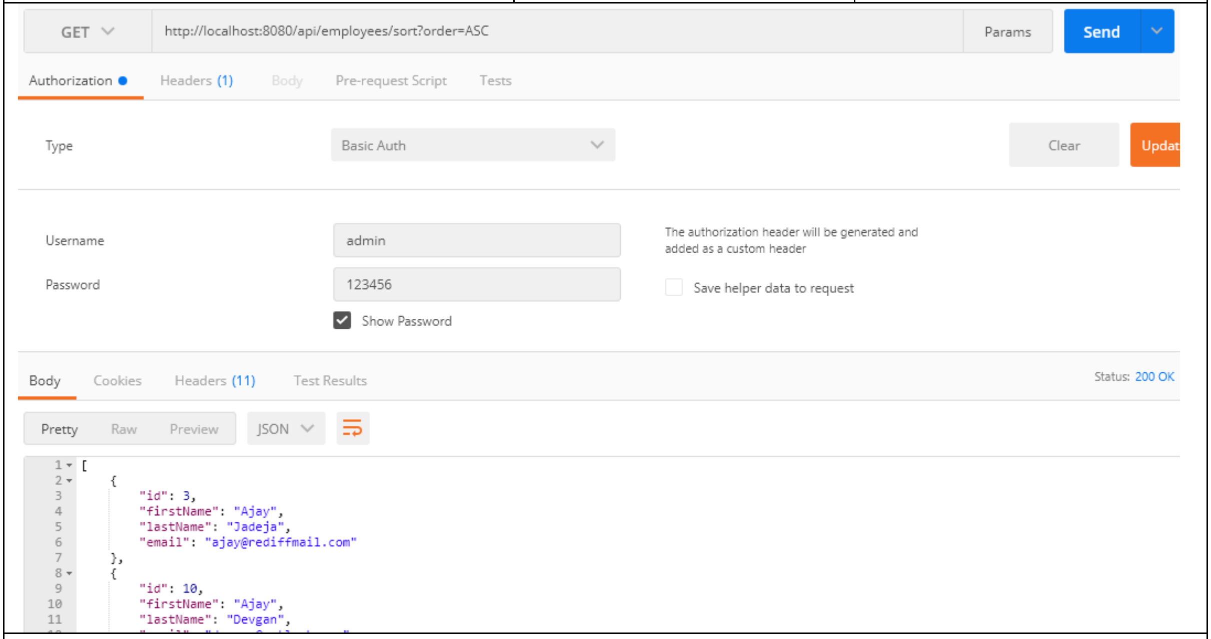
Status: 200 OK

JSON

```
[
    {
        "id": 8,
        "firstName": "Yuvraj",
        "lastName": "Singh",
        "email": "yuvraj@gmail.com"
    },
    {
        "id": 7,
        "firstName": "Virender",
        "lastName": "Sehwag",
```

Both roles 'ADMIN' & 'USER' has authorization to List all Employees Sorted

Option: List all Employees Sorted	Role: ADMIN	Status: Success



GET http://localhost:8080/api/employees/sort?order=ASC

Type: Basic Auth

Username: admin

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

□ Save helper data to request

Status: 200 OK

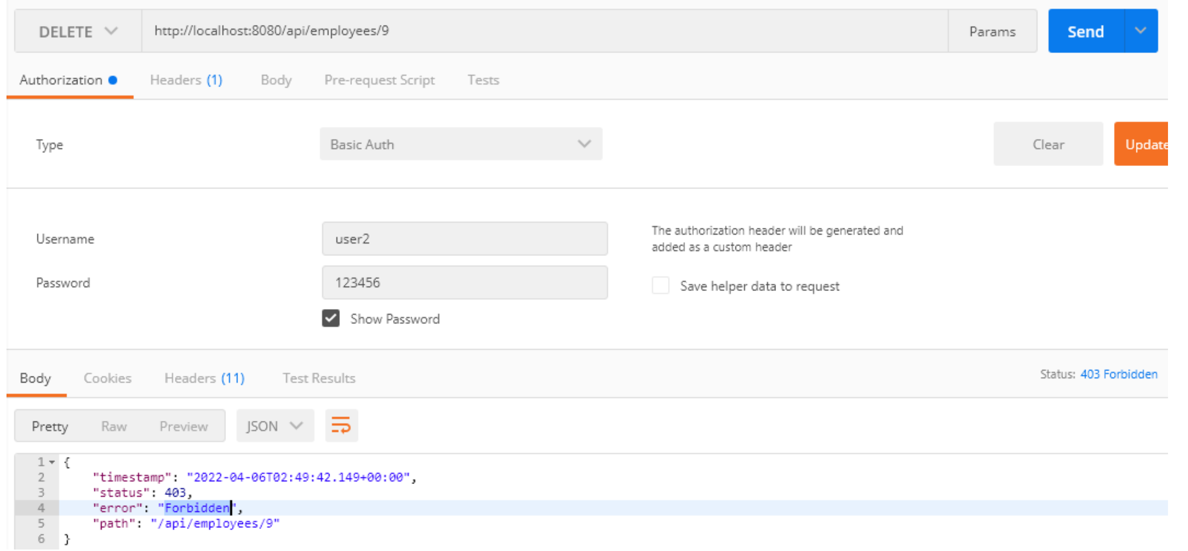
JSON

```
[
    {
        "id": 3,
        "firstName": "Ajay",
        "lastName": "Jadeja",
        "email": "ajay@rediffmail.com"
    },
    {
        "id": 10,
        "firstName": "Ajay",
        "lastName": "Devgan",
```

Both roles 'ADMIN' & 'USER' has authorization to List all Employees Sorted

Delete an Employee

Option: Delete an Employee	Role: USER	Status: Forbidden



DELETE http://localhost:8080/api/employees/9

Type: Basic Auth

Username: user2

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

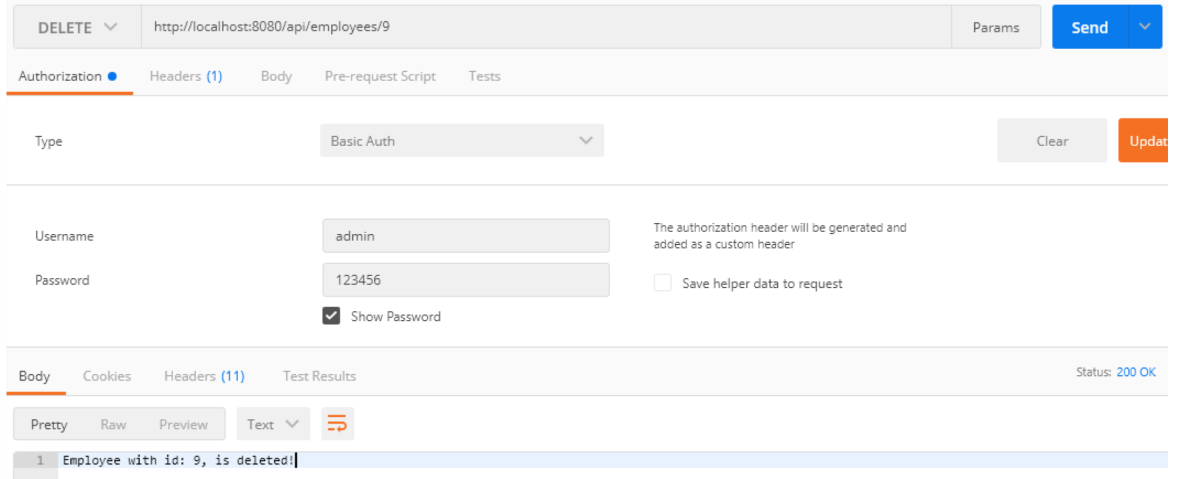
□ Save helper data to request

Status: 403 Forbidden

```
{
    "timestamp": "2022-04-06T02:49:42.149+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/api/employees/9"
}
```

Role ‘ADMIN’ only has authorization to Delete an Employee.

Option: Delete an Employee	Role: ADMIN	Status: Success



DELETE http://localhost:8080/api/employees/9

Type: Basic Auth

Username: admin

Password: 123456

☑ Show Password

The authorization header will be generated and added as a custom header

□ Save helper data to request

Status: 200 OK

```
Employee with id: 9, is deleted!
```

Role ‘ADMIN’ only has authorization to Delete an Employee.

Thank you.