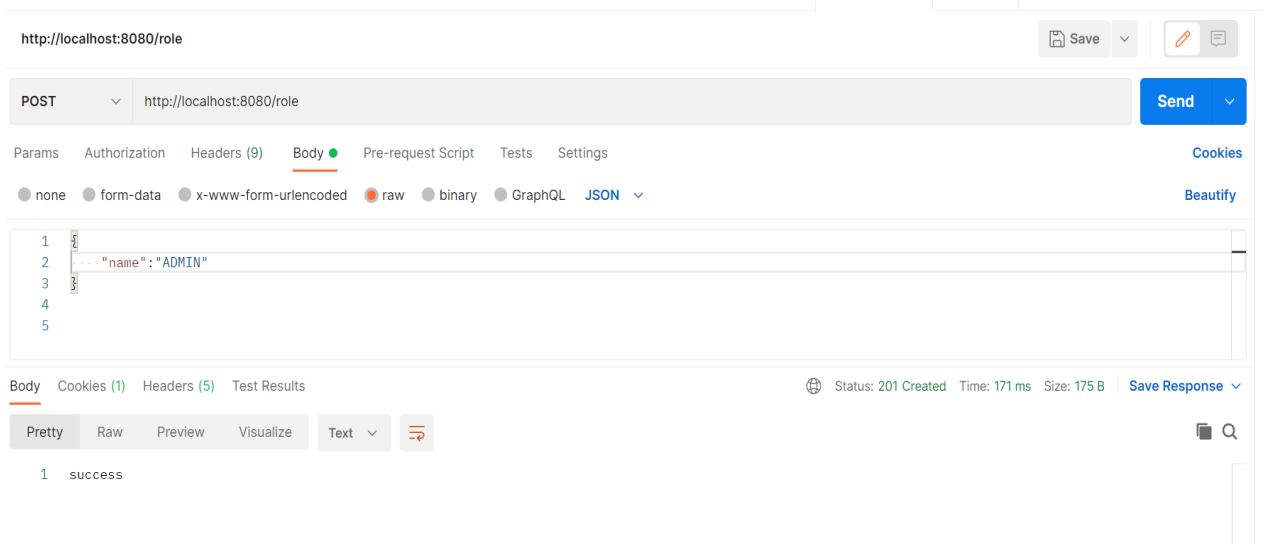
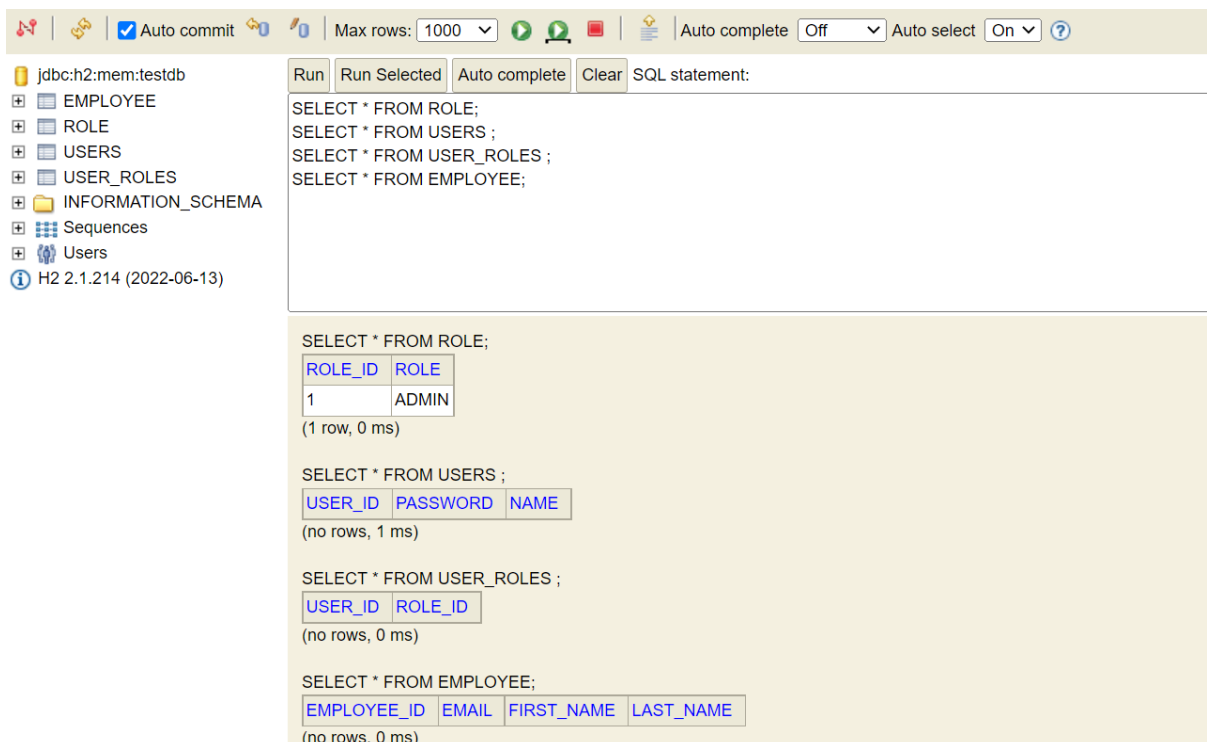


## Employee Management web application

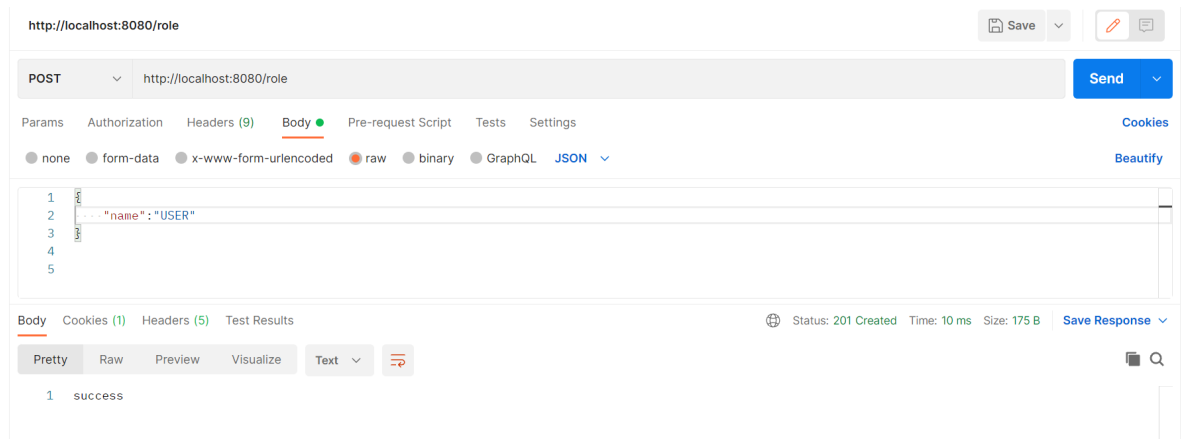
1. Creating Roles into a database.
  - a. Creating an Admin role into DB without any authentication.



Below is the screenshot of the H2 database after adding admin role



- b. Creating an User role into DB without any authentication.



Below is the screenshot of the H2 database after adding User role.

jdbc:h2:mem:testdb

- EMPLOYEE
- ROLE
- USERS
- USER\_ROLES
- INFORMATION\_SCHEMA
- Sequences
- Users
- H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

```
SELECT * FROM ROLE;
SELECT * FROM USERS ;
SELECT * FROM USER_ROLES ;
SELECT * FROM EMPLOYEE;
```

SELECT \* FROM ROLE;

ROLE_ID	ROLE
1	ADMIN
2	USER

(2 rows, 0 ms)

SELECT \* FROM USERS ;

USER_ID	PASSWORD	NAME
---------	----------	------

(no rows, 0 ms)

SELECT \* FROM USER\_ROLES ;

USER_ID	ROLE_ID
---------	---------

(no rows, 0 ms)

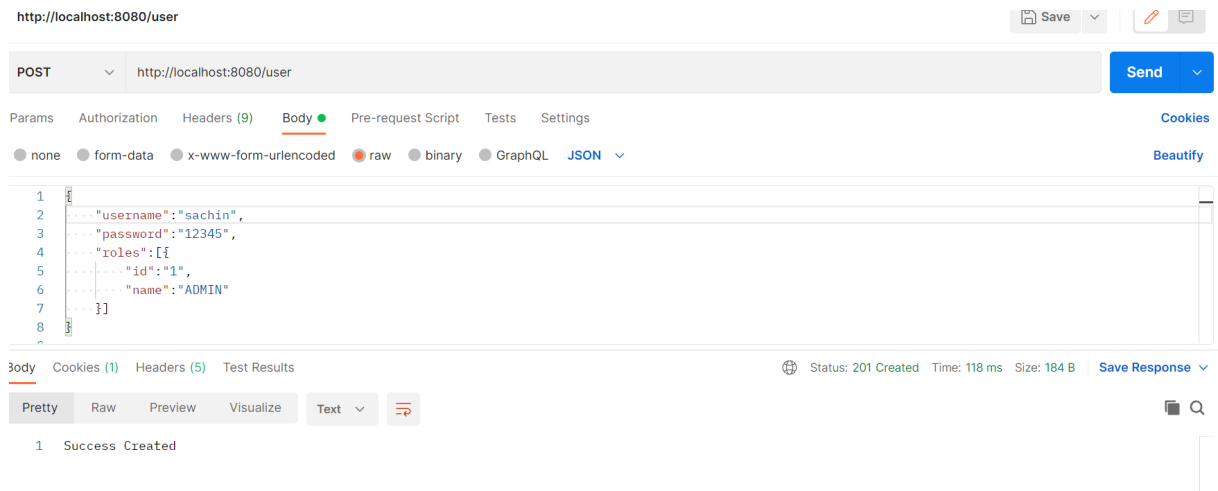
SELECT \* FROM EMPLOYEE;

EMPLOYEE_ID	EMAIL	FIRST_NAME	LAST_NAME
-------------	-------	------------	-----------

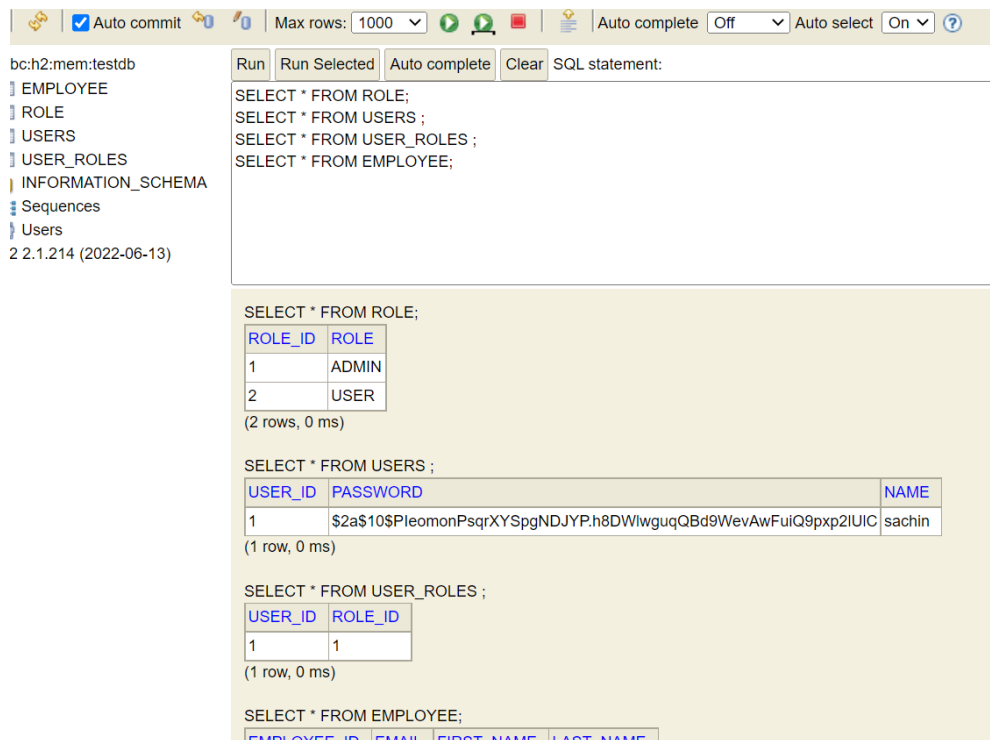
(no rows, 0 ms)

2. Creating User details into the database without authentication and password is storing in encoded format.

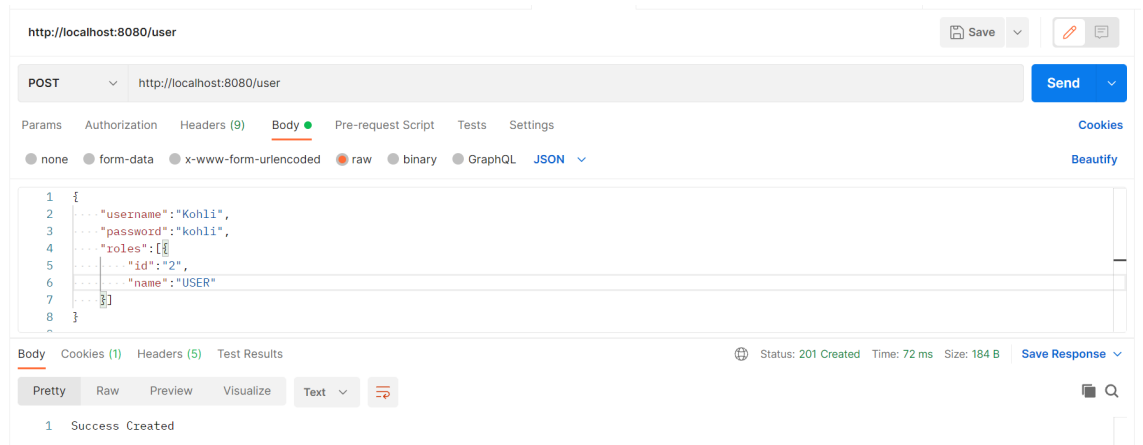
a. Adding User with ADMIN role.



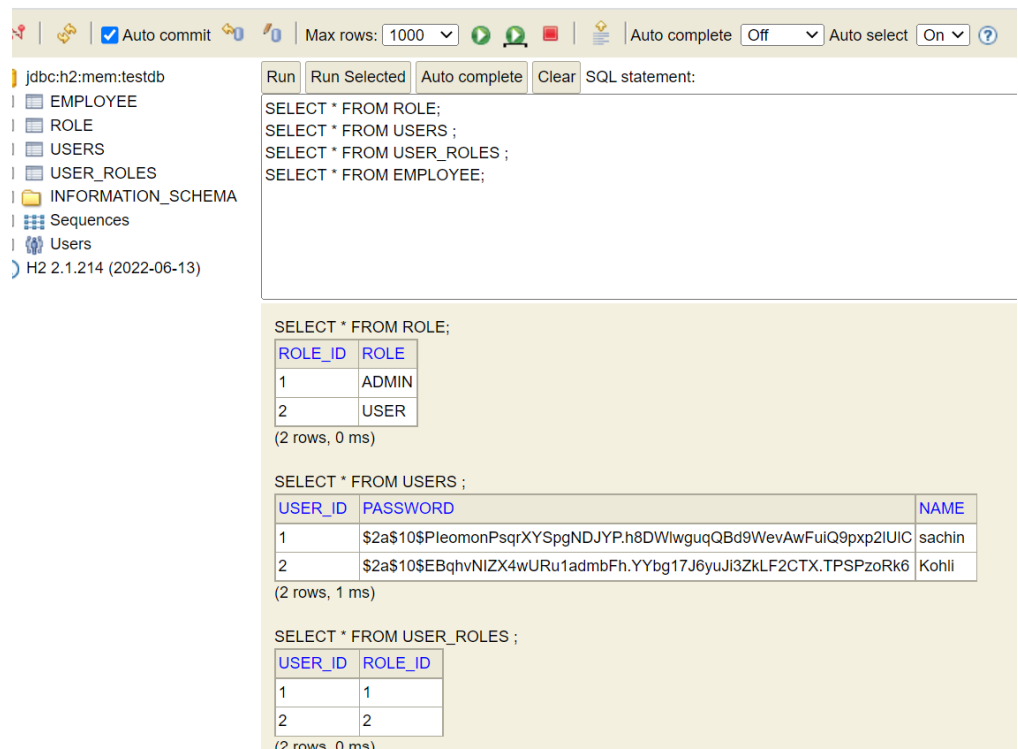
Below is the screenshot of the h2 database after adding user details.



b. creating a User with a USER role.

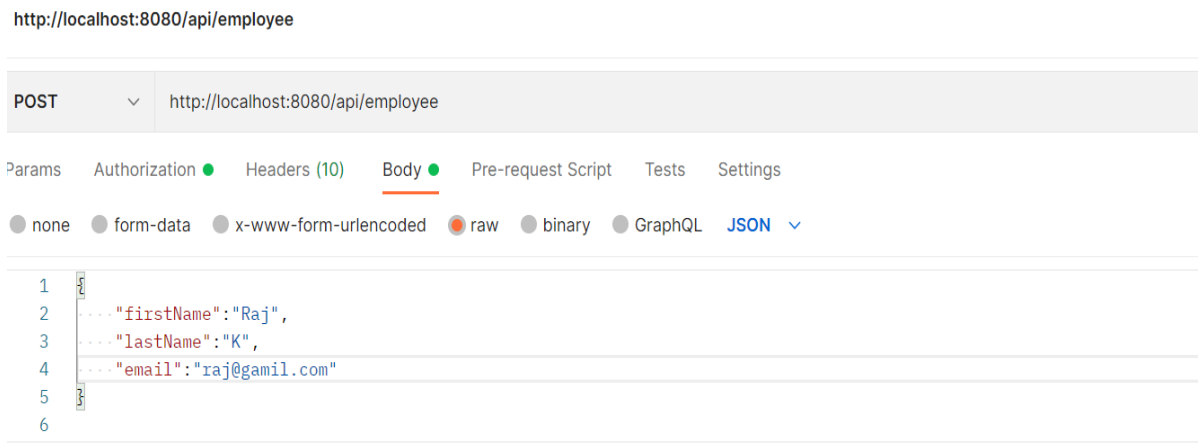


Below is the screenshot of the h2 database after adding the user contains USER role.

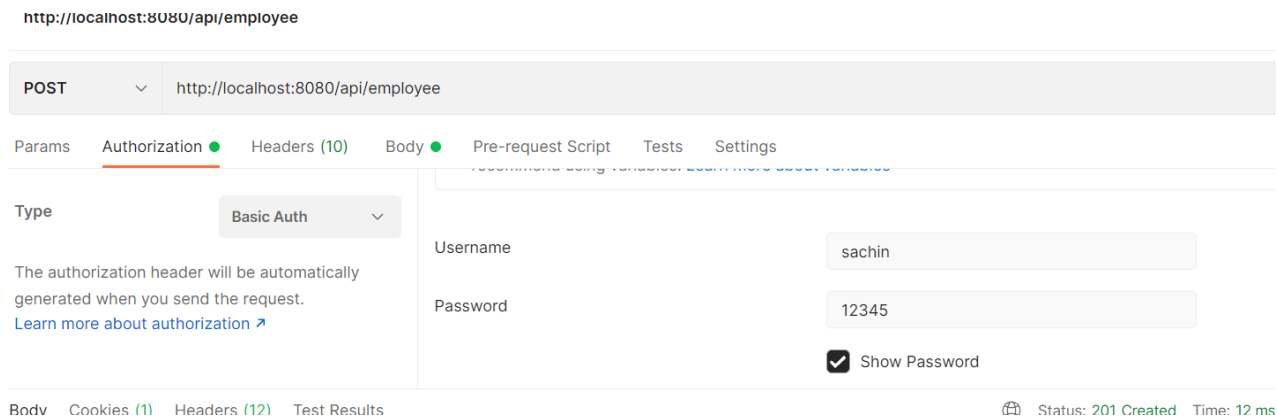


3. Creating Employee Details into the DB.

- a. Using ADMIN Role.
  - i. Screenshot of input json body.



- ii. Screenshot of the authorization passing.



- iii. Upon executing, the record has been created in the database. Below is the screenshot of the response

http://localhost:8080/api/employee

POST http://localhost:8080/api/employee

Params Authorization Headers (10) Body Pre-request Script Tests Settings

Type Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Username sachin

Password 12345

☒ Show Password

Body Cookies (1) Headers (13) Test Results

Status: 201 Created Time: 412 ms Size: 451 B Save Response

Pretty Raw Preview Visualize Text

1

Run Run Selected Auto complete Clear SQL statement:

```
SELECT * FROM ROLE;
SELECT * FROM USERS ;
SELECT * FROM USER_ROLES ;
SELECT * FROM EMPLOYEE;
```

(2 rows, 0 ms)

SELECT \* FROM USERS ;

USER_ID	PASSWORD	NAME
1	\$2a\$10\$PleomonPsqrXYSpqNDJYP.h8DWlwguqQBd9WevAwFuiQ9pxp2lUIC	sachin
2	\$2a\$10\$EBqhvNIZX4wURu1admbFh.YYbg17J6yuJi3ZkLF2CTX.TPSPzoRk6	Kohli

(2 rows, 0 ms)

SELECT \* FROM USER\_ROLES ;

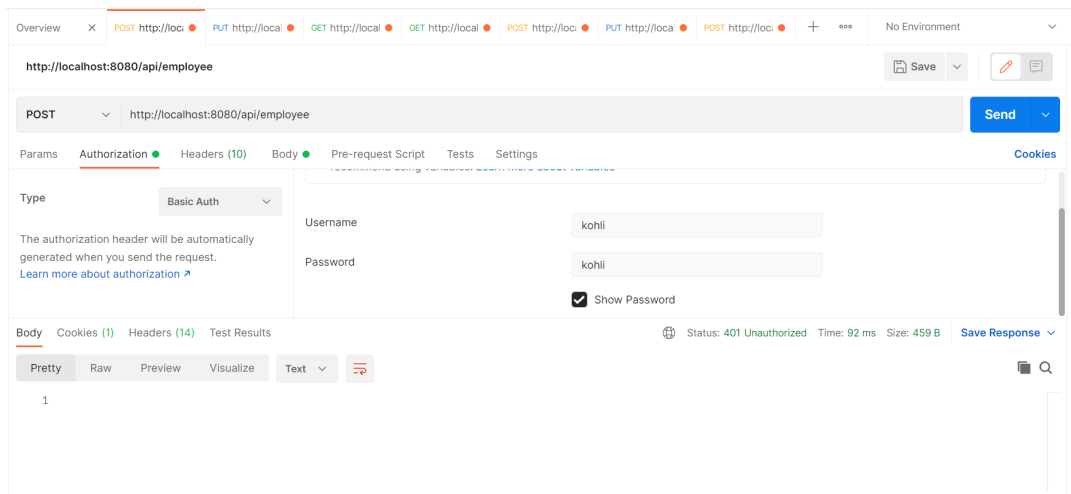
USER_ID	ROLE_ID
1	1
2	2

(2 rows, 1 ms)

SELECT \* FROM EMPLOYEE;

EMPLOYEE_ID	EMAIL	FIRST_NAME	LAST_NAME
1	raj@gamil.com	Raj	K

## b. With USER Role.

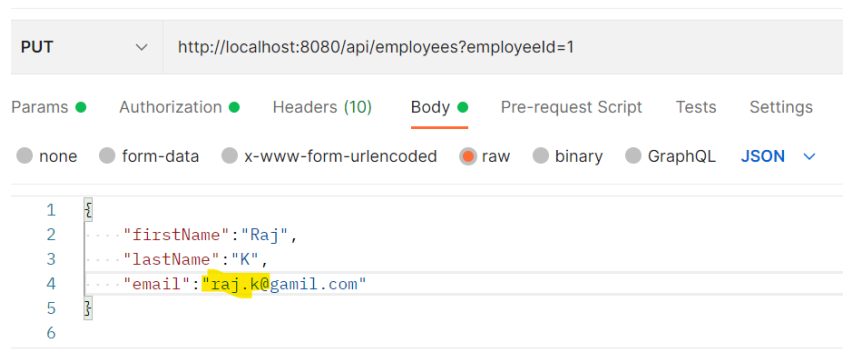


## 4. Updating the Record

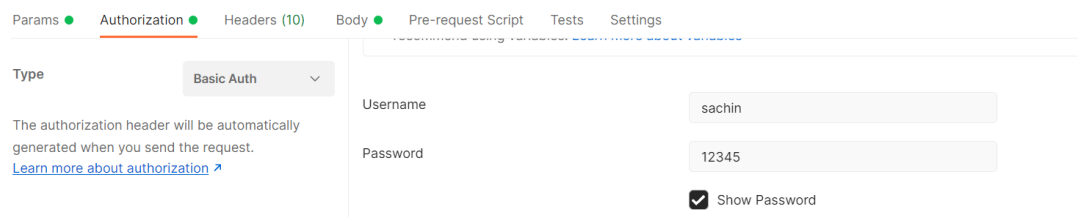
### a. with ADMIN Role

#### i. Input json (modified the email from raj to raj.k)

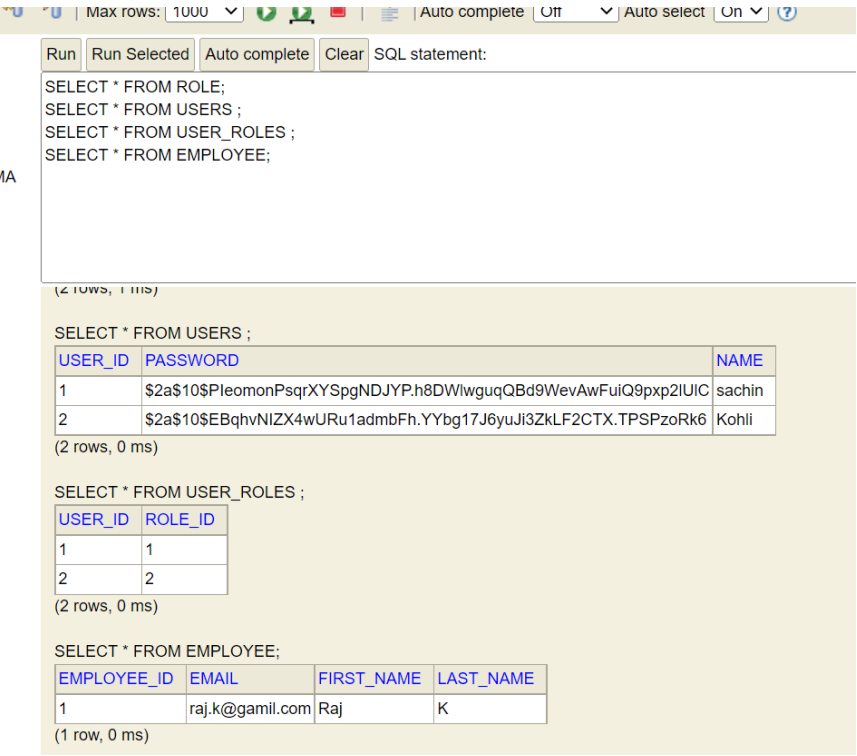
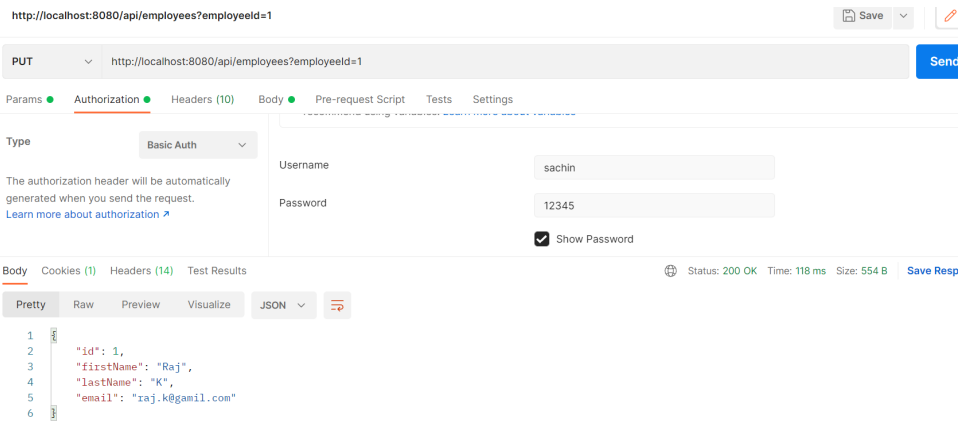
`http://localhost:8080/api/employees?employeeId=1`



#### ii. Authorization input screenshot.

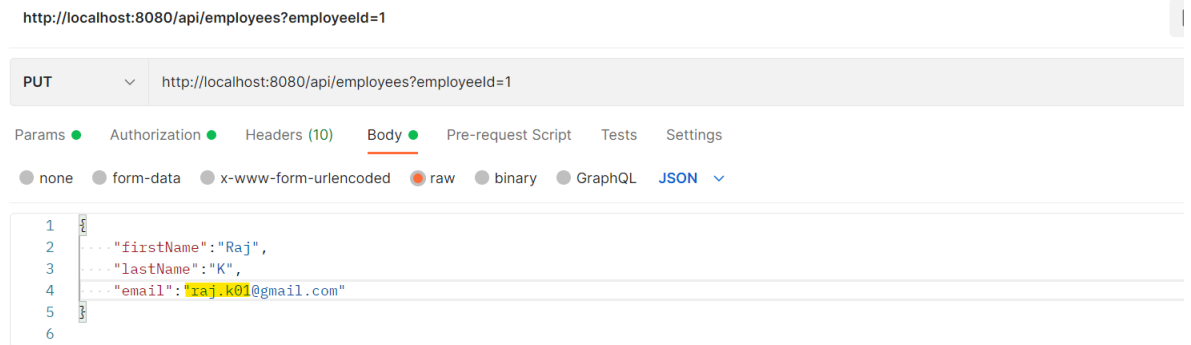


Upon executing the API, below screenshot of the response.

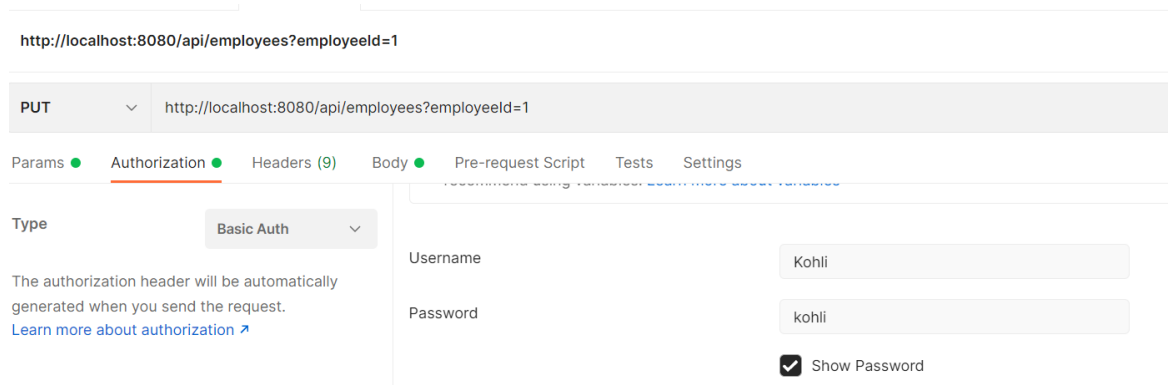




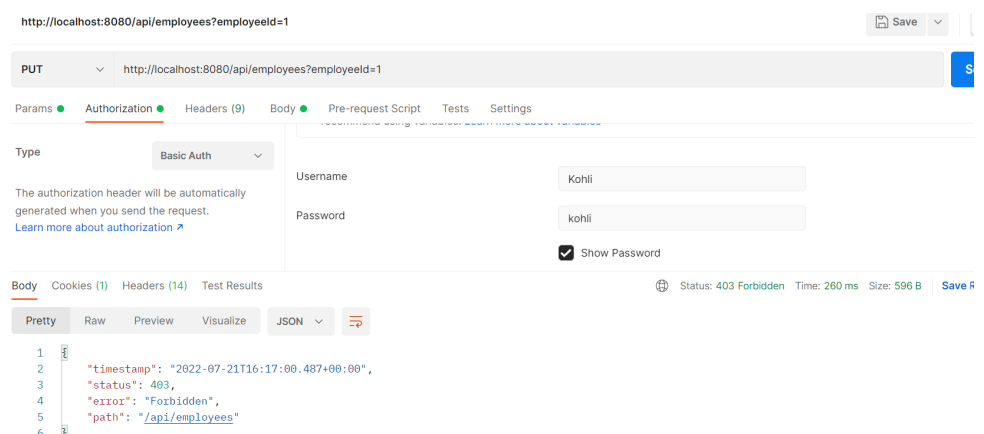
- b. With USER role.
- Input json (modified the email from raj to raj.k01)



- Authorization input screenshot.



- Upon executing the API, below screenshot of the response.



5. Getting all Employee records.

a. With ADMIN role.

http://localhost:8080/api/employees

Save

GET

http://localhost:8080/api/employees

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Type

Basic Auth

The authorization header will be automatically generated when you send the request.  
[Learn more about authorization](#)

Username

sachin

Password

12345

☒ Show Password

Body

Cookies (1)

Headers (13)

Test Results

200 OK

Time: 25 ms

Size: 481 B

Sa

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": 1,
3   "firstName": "Raj",
4   "lastName": "K",
5   "email": "raj.k@gmail.com"
6 }
```

Find and Replace

Console

Find and Replace

Console

b. With USER role.

http://localhost:8080/api/employees

Save

GET

http://localhost:8080/api/employees

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Type

Basic Auth

The authorization header will be automatically

Username

Kohli

Password

kohli

Body

Cookies (1)

Headers (14)

Test Results

200 OK

Time: 218 ms

Size: 624 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": 1,
3   "firstName": "Raj",
4   "lastName": "K",
5   "email": "raj@gmail.com"
6 },
7 {
8   "id": 2,
9   "firstName": "Benny",
10 }
```

6. Getting employee record based on employeeid

a. Using ADMIN role.

Overview | POST http://loc: | PUT http://local: | GET http://local: | GET http://local: | POST http://loc: | PUT http://loca: | POST http://loc: | + | No Environment

http://localhost:8080/api/employees/1 | Save |

GET | http://localhost:8080/api/employees/1 | Send

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings

Type | Basic Auth |  
The authorization header will be automatically generated when you send the request.  
[Learn more about authorization](#) | Username: sachin | Password: 12345 | ☒ Show Password

Body | Cookies (1) | Headers (13) | Test Results | Status: 200 OK | Time: 37 ms | Size: 479 B | Save Res

Pretty | Raw | Preview | Visualize | JSON |  
1 | {  
2 | "id": 1,  
3 | "firstName": "Raj",  
4 | "lastName": "K",  
5 | "email": "raj.k@gmail.com"  
6 | }

b. With USER Role.

http://localhost:8080/api/employees/2 | Save |

GET | http://localhost:8080/api/employees/2 | Send

Params | Authorization | Headers (7) | Body | Pre-request Script | Tests | Settings | Cook

Type | Basic Auth | Username: Kohli | Password: kohli

Body | Cookies (1) | Headers (13) | Test Results | Status: 200 OK | Time: 44 ms | Size: 481 B | Save Response

Pretty | Raw | Preview | Visualize | JSON |  
1 | {  
2 | "id": 2,  
3 | "firstName": "Benny",  
4 | "lastName": "K",  
5 | "email": "benny@gmail.com"  
6 | }

## 7. Getting the record list based on the sort direction

### a. With ADMIN role.

http://localhost:8080/api/employees/sort?order=asc

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

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Type Basic Auth Username sachin Password .....

The authorization header will be automatically

Body Cookies (1) Headers (14) Test Results Status: 200 OK Time: 149 ms Size: 626 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 2,
4     "firstName": "Albin",
5     "lastName": "B",
6     "email": "albin@gmail.com"
7   },
8   {
9     "id": 1,
10    "firstName": "Raj",
11    "lastName": "K"
```

http://localhost:8080/api/employees/sort?order=desc

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

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Type Basic Auth Username sachin Password .....

The authorization header will be automatically

Body Cookies (1) Headers (13) Test Results Status: 200 OK Time: 13 m

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "firstName": "Raj",
5     "lastName": "K",
6     "email": "raj.k@gmail.com"
7   },
8   {
9     "id": 2,
10    "firstName": "Albin",
11    "lastName": "B"
```

## b. With User Role.

http://localhost:8080/api/employees/sort?order=asc

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

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Type Basic Auth

The authorization header will be automatically

Username Kohli

Password kohli

Status: 200 OK Time: 65 ms Size: 549 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 2,
4     "firstName": "Benny",
5     "lastName": "K",
6     "email": "benny@gmail.com"
7   },
8   {
9     "id": 1,
10    "firstName": "Raj",
```

http://localhost:8080/api/employees/sort?order=desc

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

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Type Basic Auth

The authorization header will be automatically

Username Kohli

Password kohli

Status: 200 OK Time: 14 ms Size: 549 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "firstName": "Raj",
5     "lastName": "K",
6     "email": "raj@gmail.com"
7   },
8   {
9     "id": 2,
10    "firstName": "Benny",
```

## 8. Search the employee record with firstName.

### a. ADMIN role.

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

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

Authorization: Basic Auth

Username: sachin

Password: .....

Status: 200 OK Time: 14 ms Size: 483 B

```
1 {
2   "id": 2,
3   "firstName": "Albin",
4   "lastName": "B",
5   "email": "albin@gmail.com"
6 }
7
8
```

### b. USER role.

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

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

Authorization: Basic Auth

Username: Kohli

Password: kohli

Status: 200 OK Time: 30 ms Size: 483 B

```
1 {
2   "id": 2,
3   "firstName": "Benny",
4   "lastName": "K",
5   "email": "benny@gmail.com"
6 }
7
8
```

9. Deleting the Record.

a. ADMIN role.

http://localhost:8080/api/employees/2

Save

DELETE

http://localhost:8080/api/employees/2

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Type

Basic Auth

Username

sachin

Password

.....

The authorization header will be automatically

Body

Cookies (1)

Headers (11)

Test Results

Status: 204 No Content

Time: 22 ms

Size: 360 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1

Run

Run Selected

Auto complete

Clear

SQL statement:

SELECT \* FROM ROLE;  
SELECT \* FROM USERS ;  
SELECT \* FROM USER\_ROLES ;  
SELECT \* FROM EMPLOYEE ;

(2 rows, 1 ms)

SELECT \* FROM USERS ;

USER_ID	PASSWORD	NAME
1	\$2a\$10\$PleomonPsqrXYSpqNDJYP.h8DWlwguqQBd9WevAwFuiQ9pxp2IUIC	sachin
2	\$2a\$10\$EBqhvNIZX4wURu1admbFh.YYbg17J6yuJi3ZkLF2CTX.TPSPzoRk6	Kohli

(2 rows, 1 ms)

SELECT \* FROM USER\_ROLES ;

USER_ID	ROLE_ID
1	1
2	2

(2 rows, 2 ms)

SELECT \* FROM EMPLOYEE ;

EMPLOYEE_ID	EMAIL	FIRST_NAME	LAST_NAME
1	raj.k@gamil.com	Raj	K

(1 row, 1 ms)

b. USER Role.

http://localhost:8080/api/employees/1

Save

DELETE

http://localhost:8080/api/employees/1

Send

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettings

Type

Basic Auth

Username

Kohli

Password

.....

The authorization header will be automatically

Body

Cookies (1)

Headers (13)

Test Results

Status: 403 Forbidden

Time: 12 ms

Size: 523 B

Save Respor

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

path

timestamp": "2022-07-21T16:24:42.559+00:00",

status": 403,

error": "Forbidden",

path": "/api/employees/1"