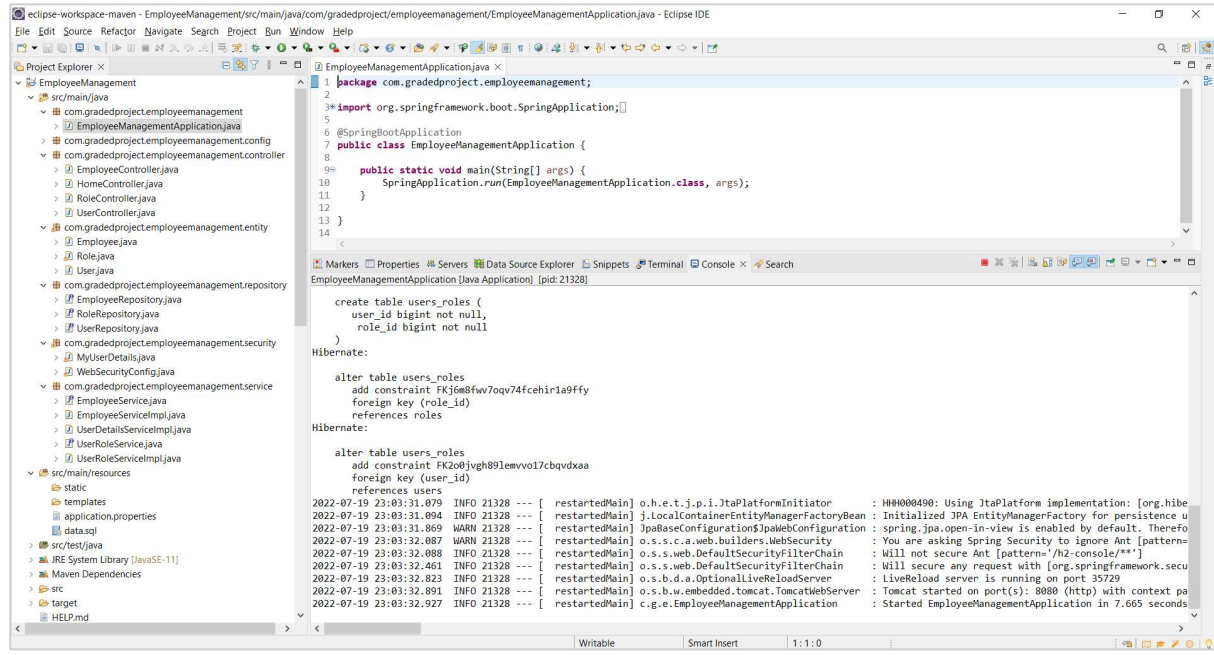


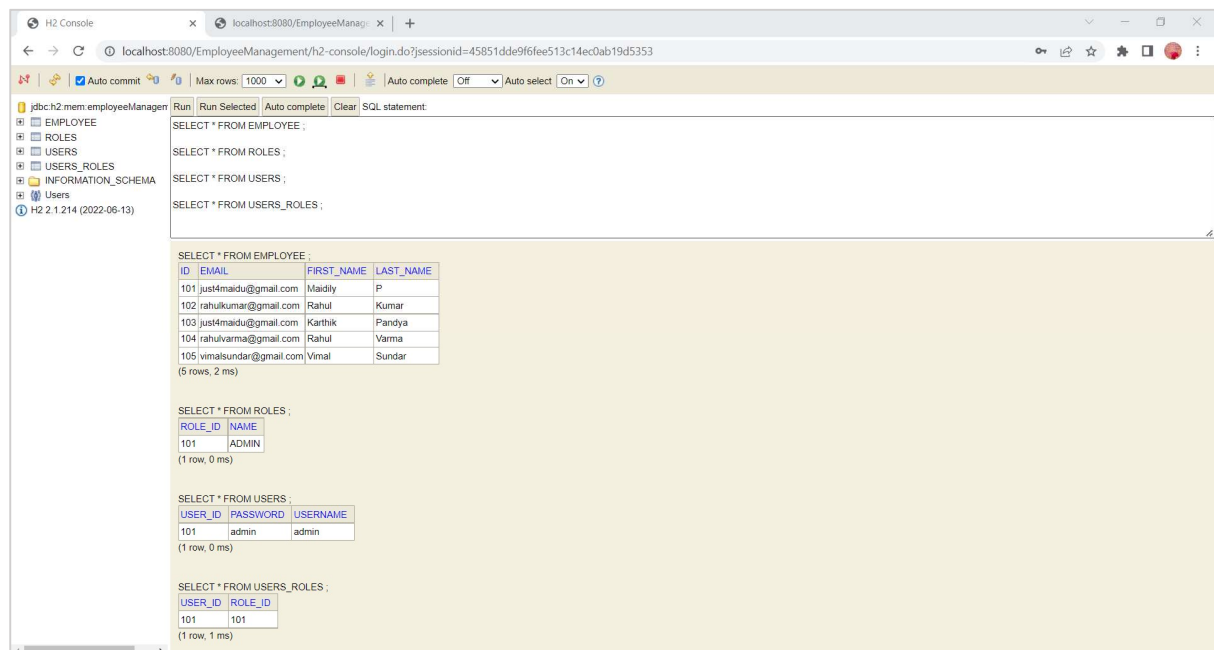
# Graded Assignment 4 – Spring REST

## Screenshots of Solution

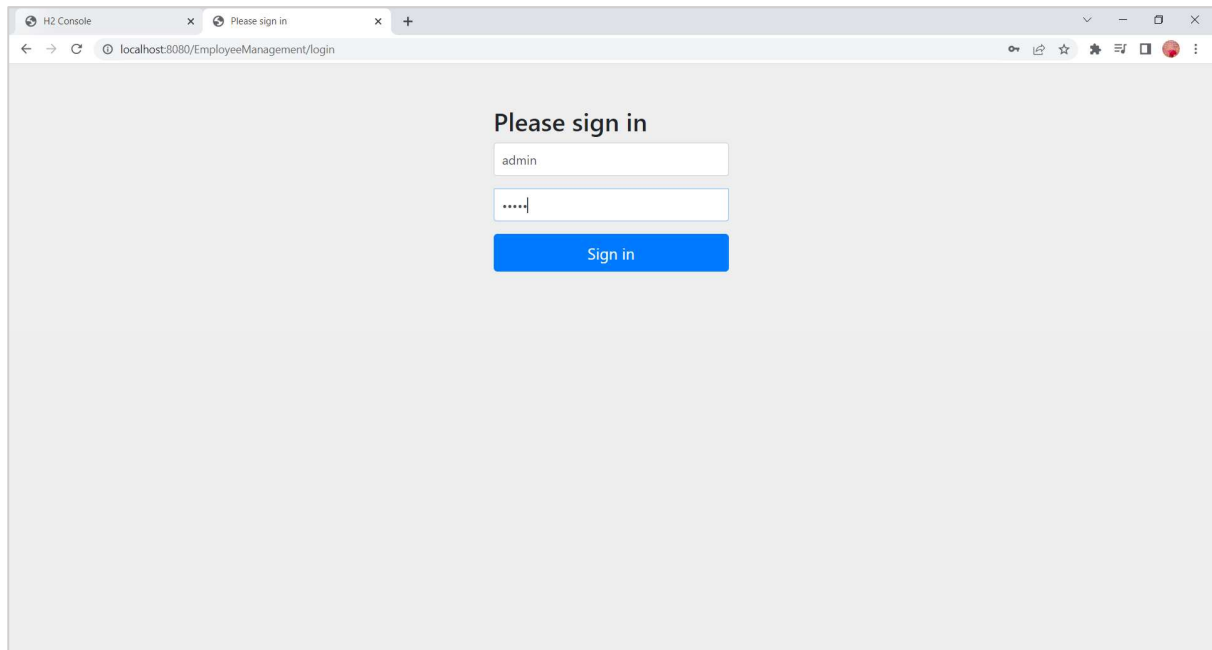
### Project Structure – Server started & application running:



### Initial entries inserted in H2-console: (during application start)

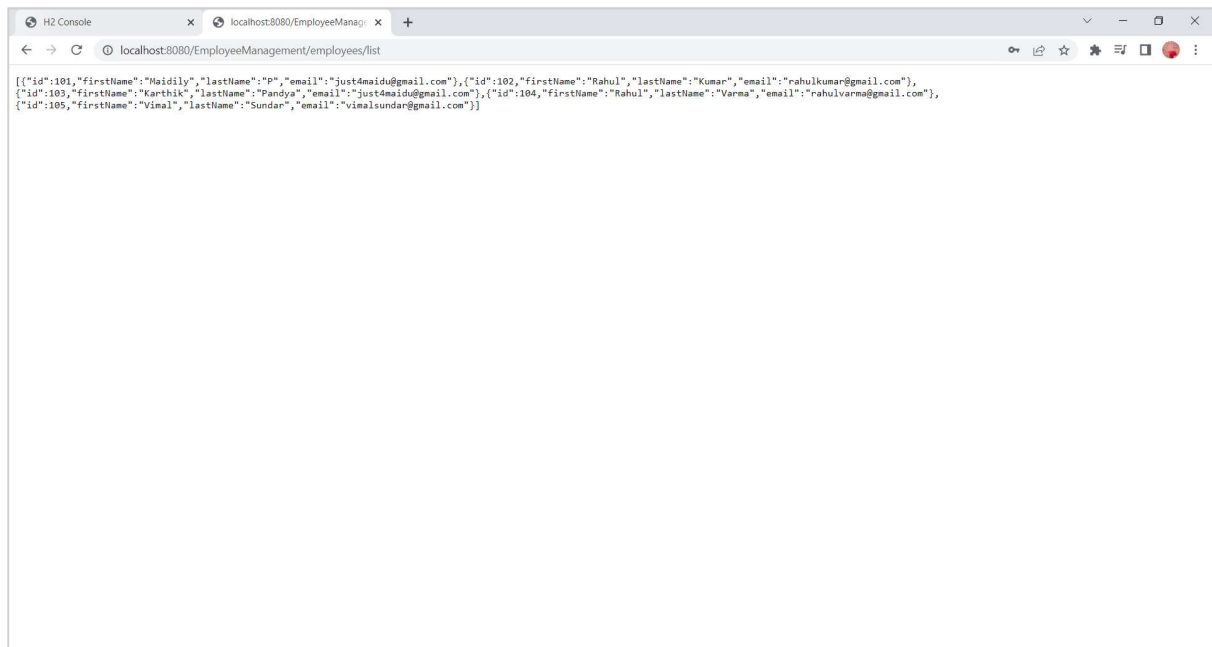


## Initial authentication using default ADMIN user:



The screenshot shows a web browser window with two tabs: 'H2 Console' and 'Please sign in'. The active tab is 'Please sign in', which displays a login form. The form has a title 'Please sign in' and two input fields: one for the username 'admin' and one for the password '\*\*\*\*'. Below the password field is a blue 'Sign in' button. The browser's address bar shows 'localhost:8080/EmployeeManagement/login'.

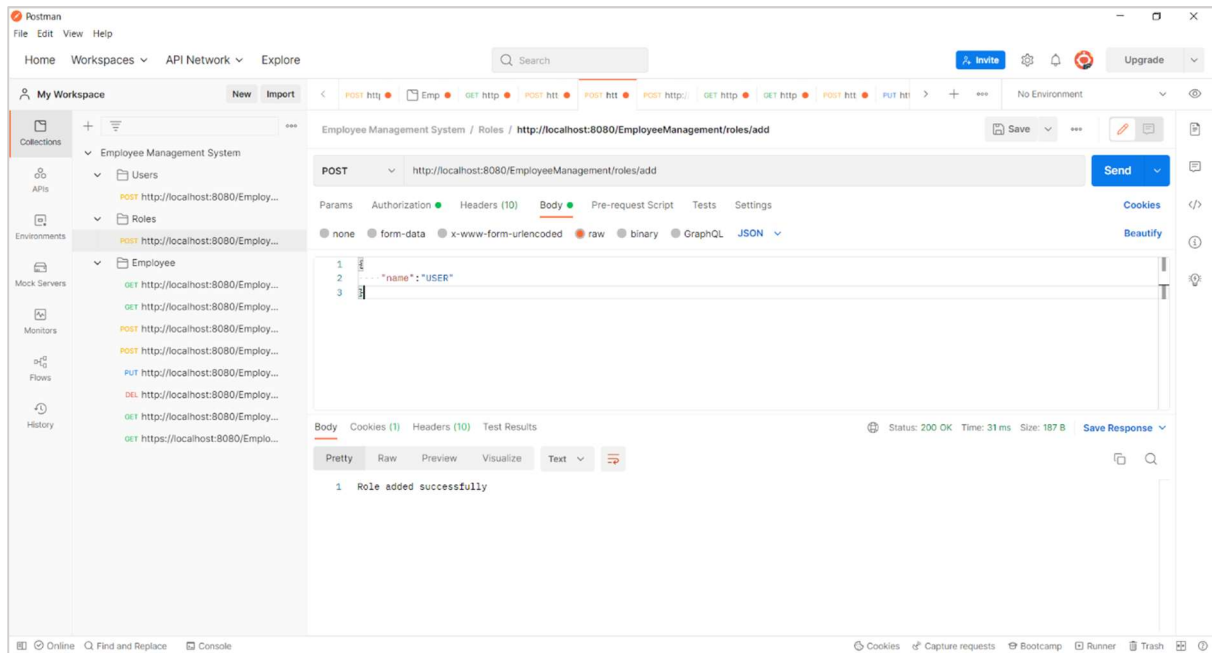
## Retrieving list of employees from browser:



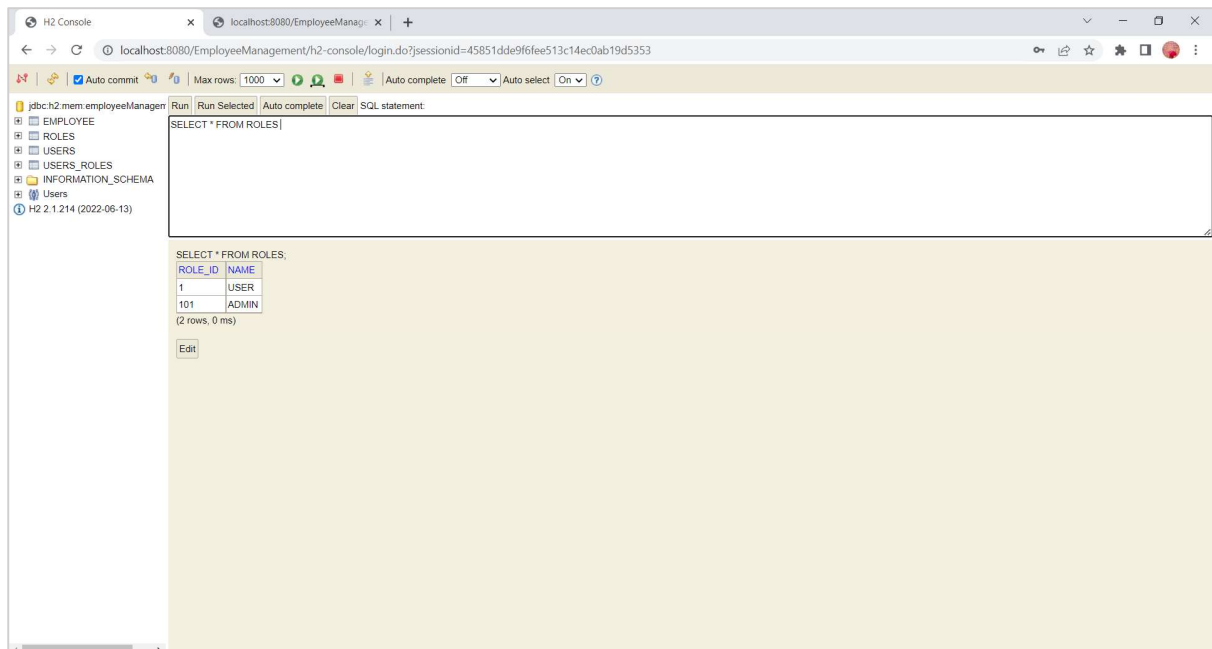
The screenshot shows a web browser window with two tabs: 'H2 Console' and 'localhost:8080/EmployeeManagement/employees/list'. The active tab is 'localhost:8080/EmployeeManagement/employees/list', which displays a JSON array of employee data. The browser's address bar shows 'localhost:8080/EmployeeManagement/employees/list'.

```
[{"id":101,"firstName":"Maidily","lastName":"P","email":"just4maidu@gmail.com"}, {"id":102,"firstName":"Rahul","lastName":"Kumar","email":"rahulkumar@gmail.com"}, {"id":103,"firstName":"Karthik","lastName":"Pandya","email":"just4maidu@gmail.com"}, {"id":104,"firstName":"Rahul","lastName":"Varma","email":"rahulvarma@gmail.com"}, {"id":105,"firstName":"Vimal","lastName":"Sundar","email":"vimalsundar@gmail.com"}]
```

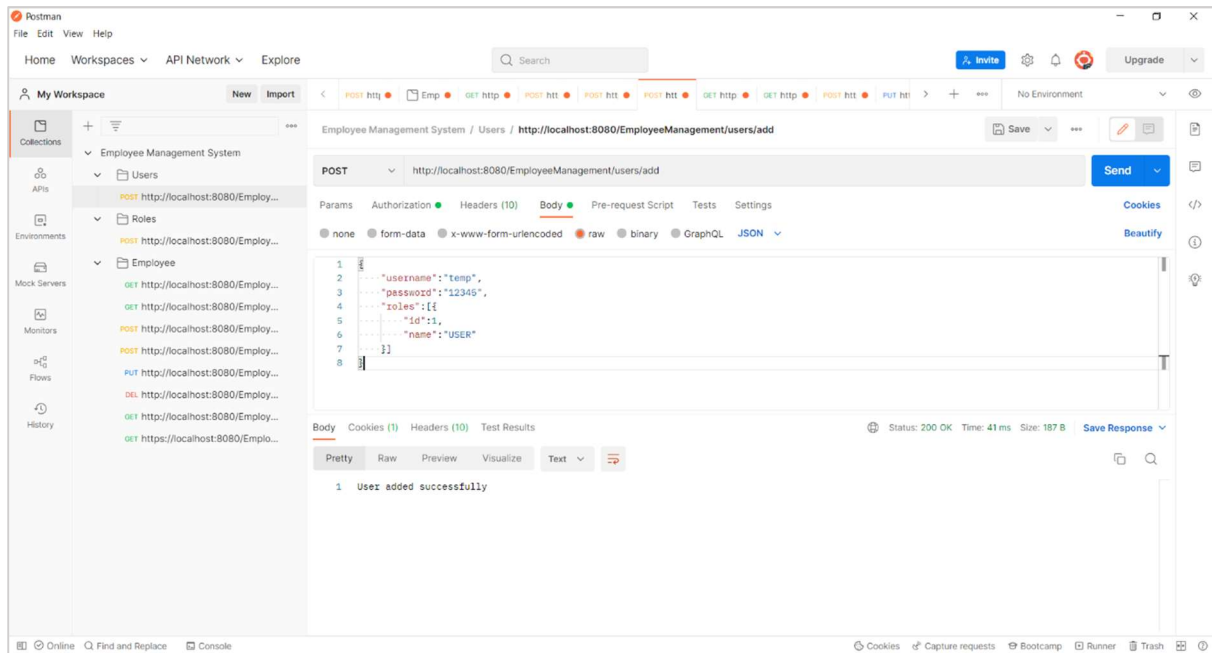
## In Postman – Add new Role: (USER)



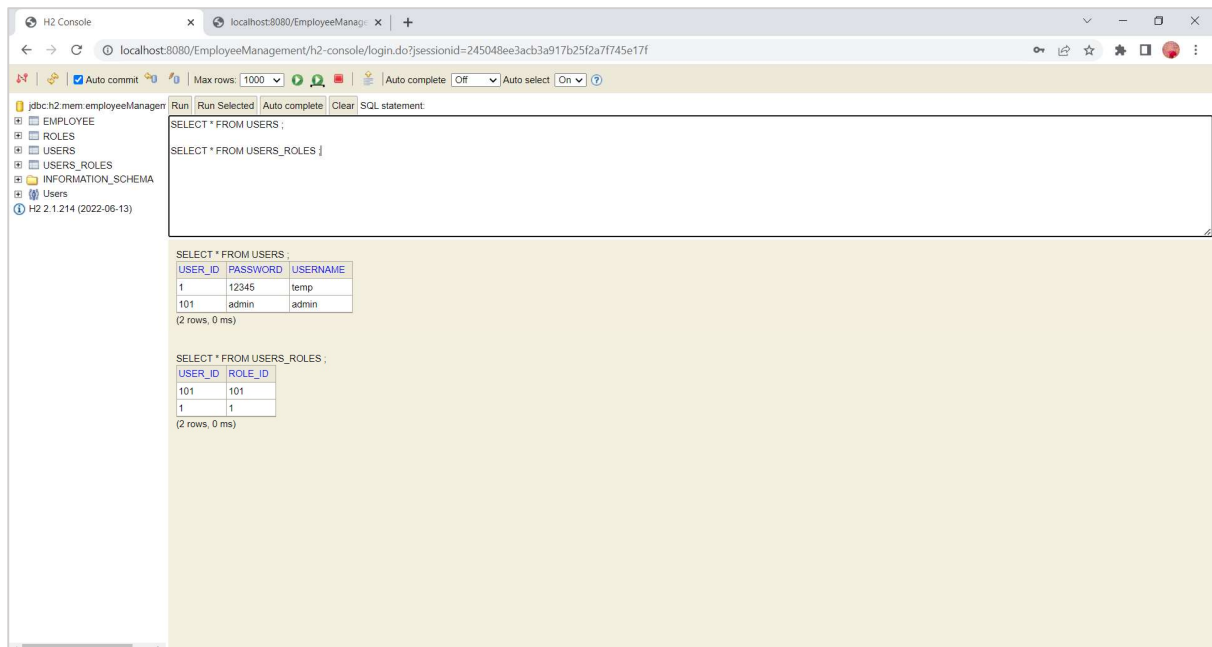
## New role addition reflecting in H2-console:



## In Postman – Add new User Data for Authentication:

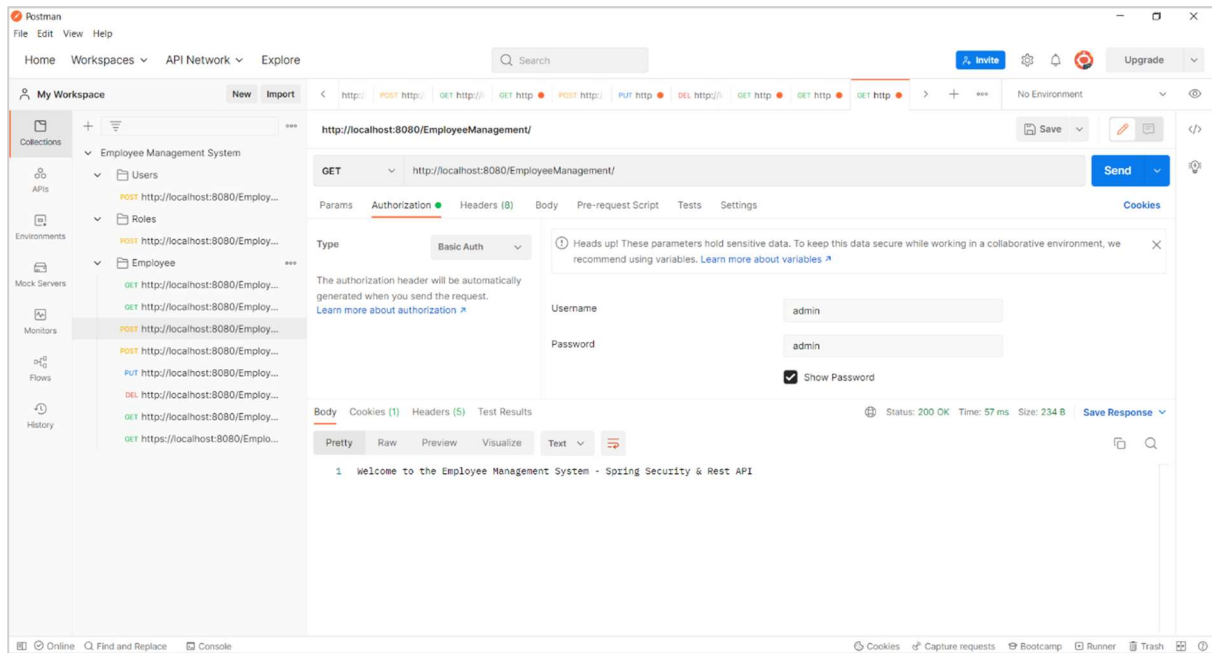


## New User data addition reflecting in H2-console:

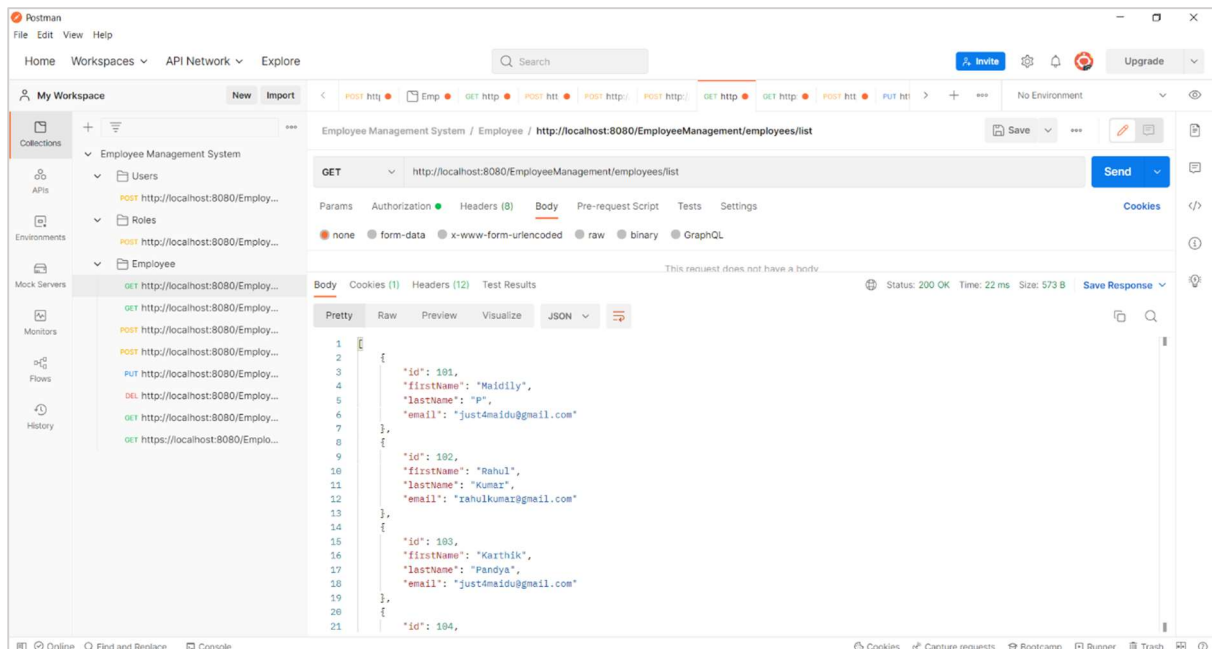


## CRUD Operations using REST API:

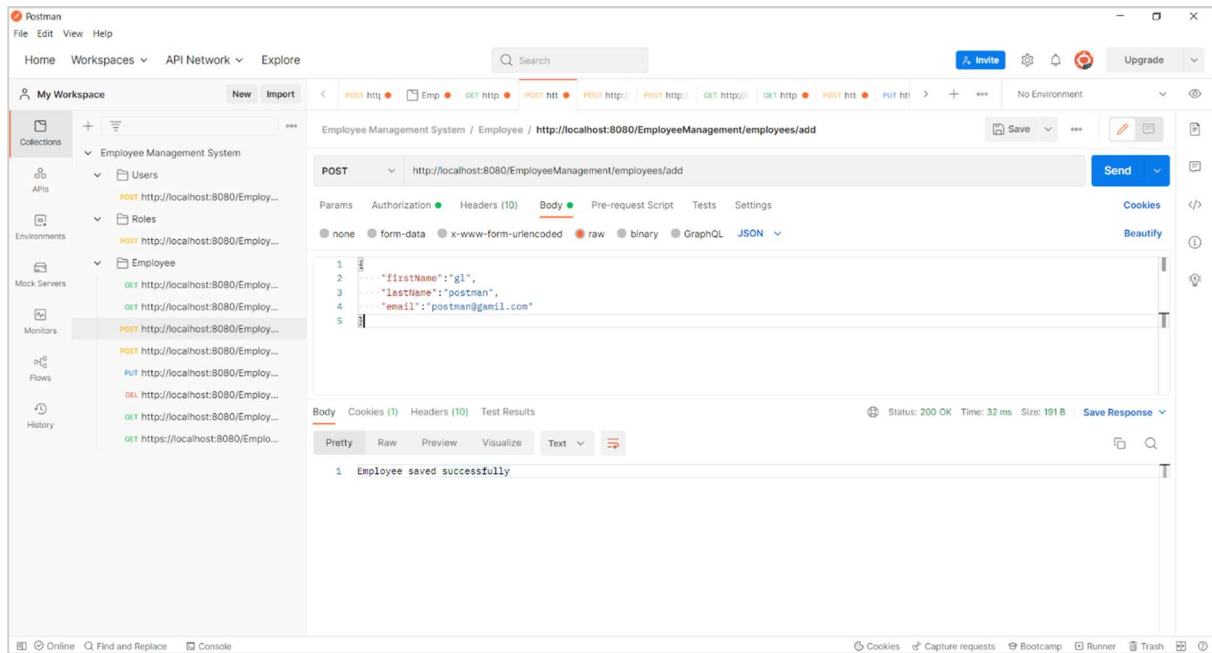
### Hitting Welcome/Home endpoint:



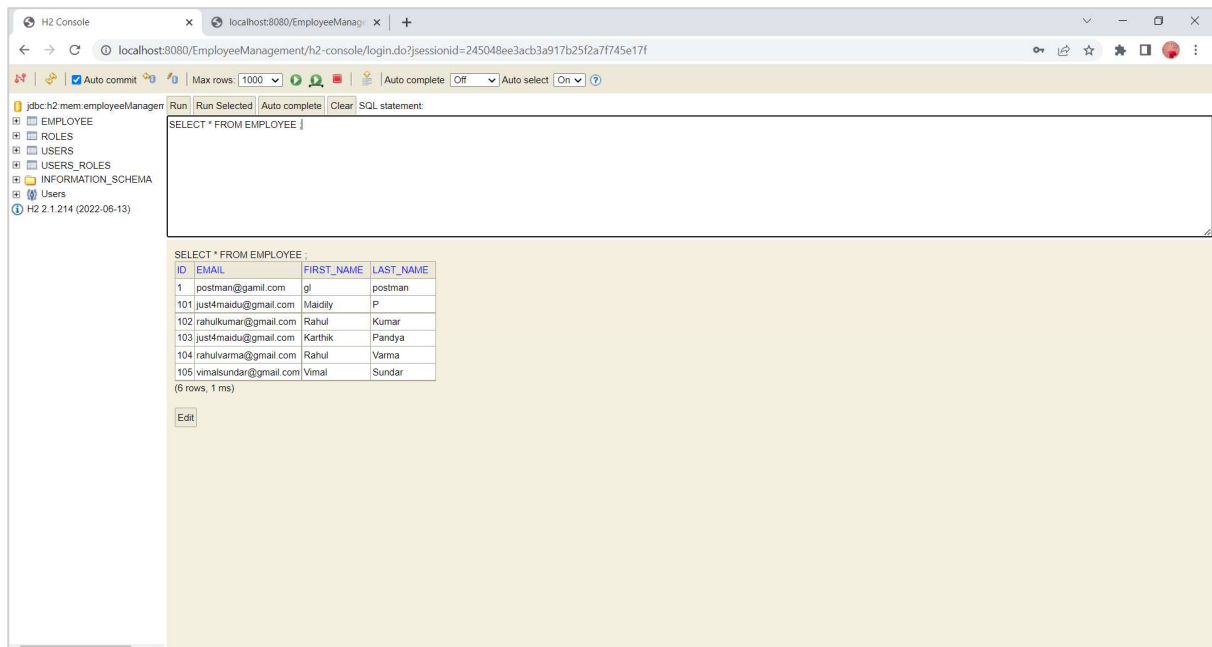
### Read/ List all Employee records stored in DB:



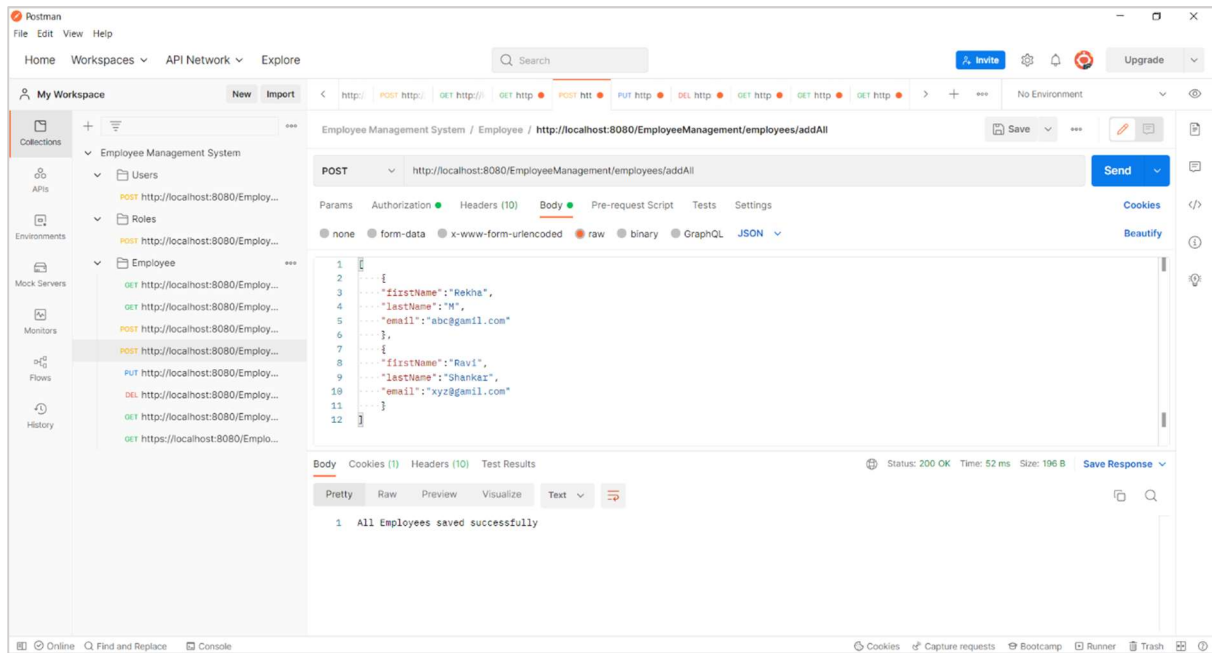
## Add new Employee record to DB: (ADMIN user)



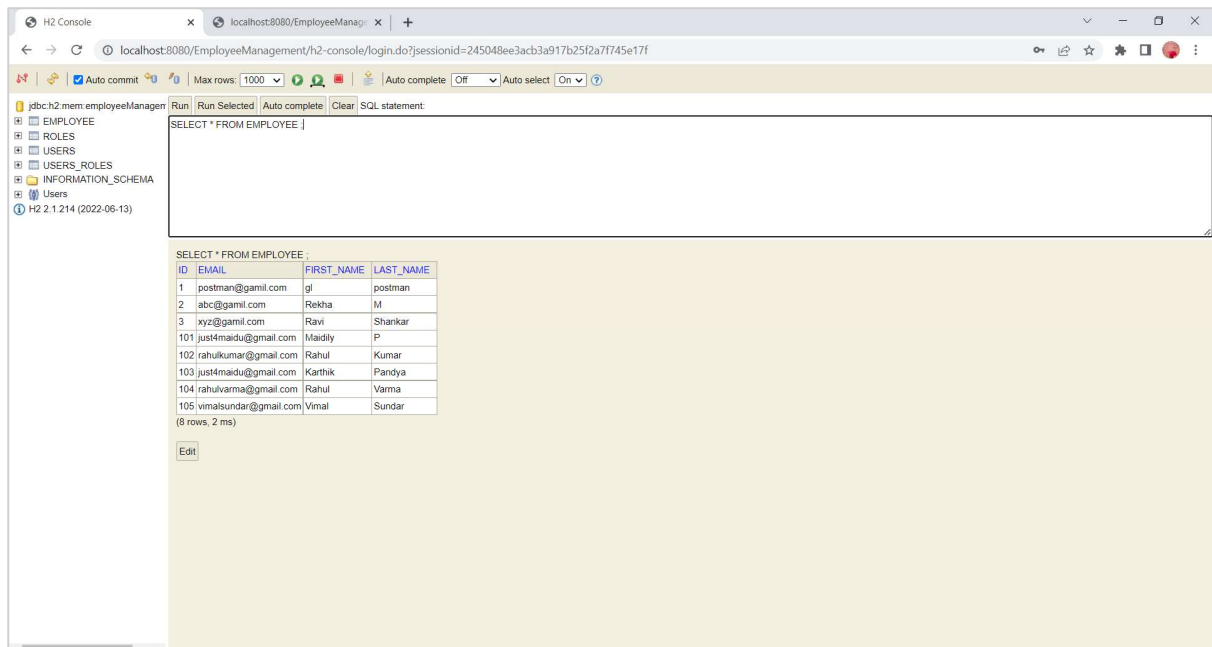
## Create action - New Employee record addition – reflecting in H2-console: [id = 1]



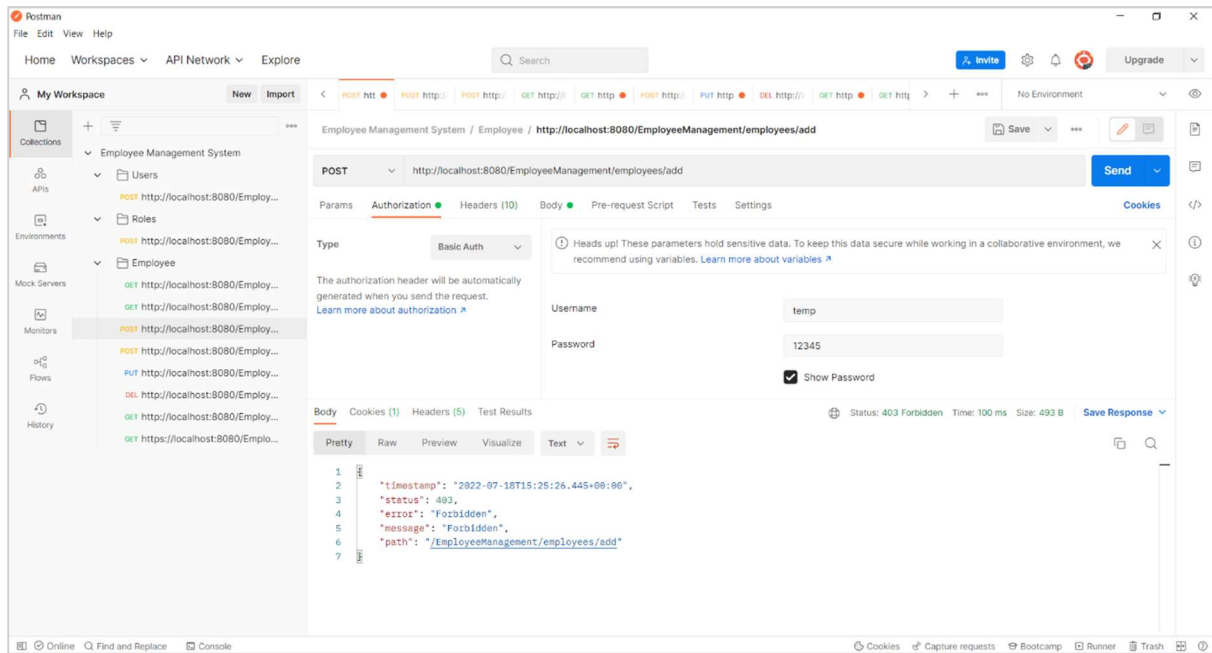
## Add many Employee records to DB: (ADMIN user)



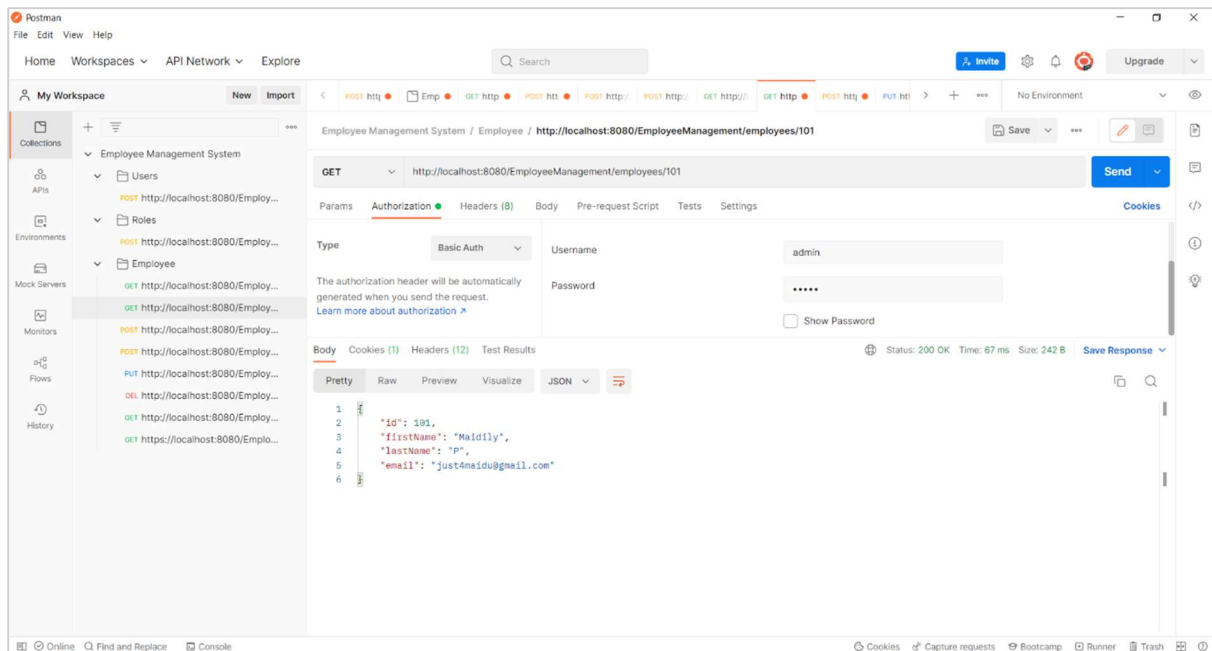
## Create action - Multiple Employee records – reflecting in H2-console: [id = 2, 3]



## Add new Employee record to DB: (USER user) *Forbidden authentication*

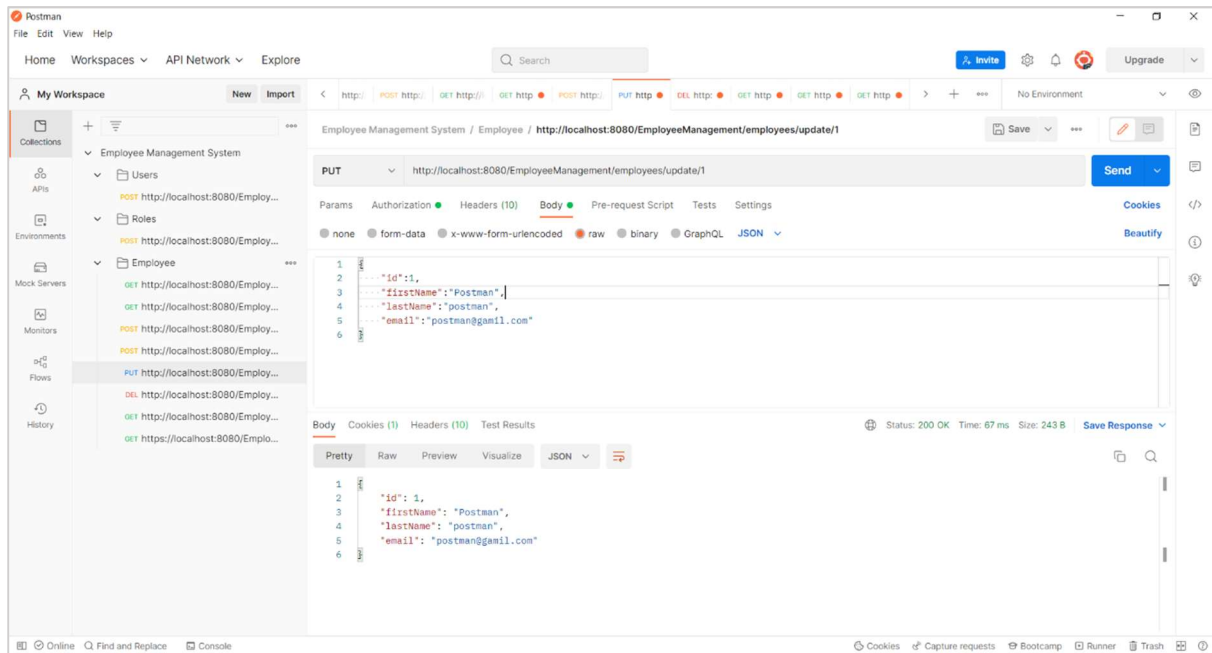


## Fetch an existing Employee record by "id": `[id = 101]`

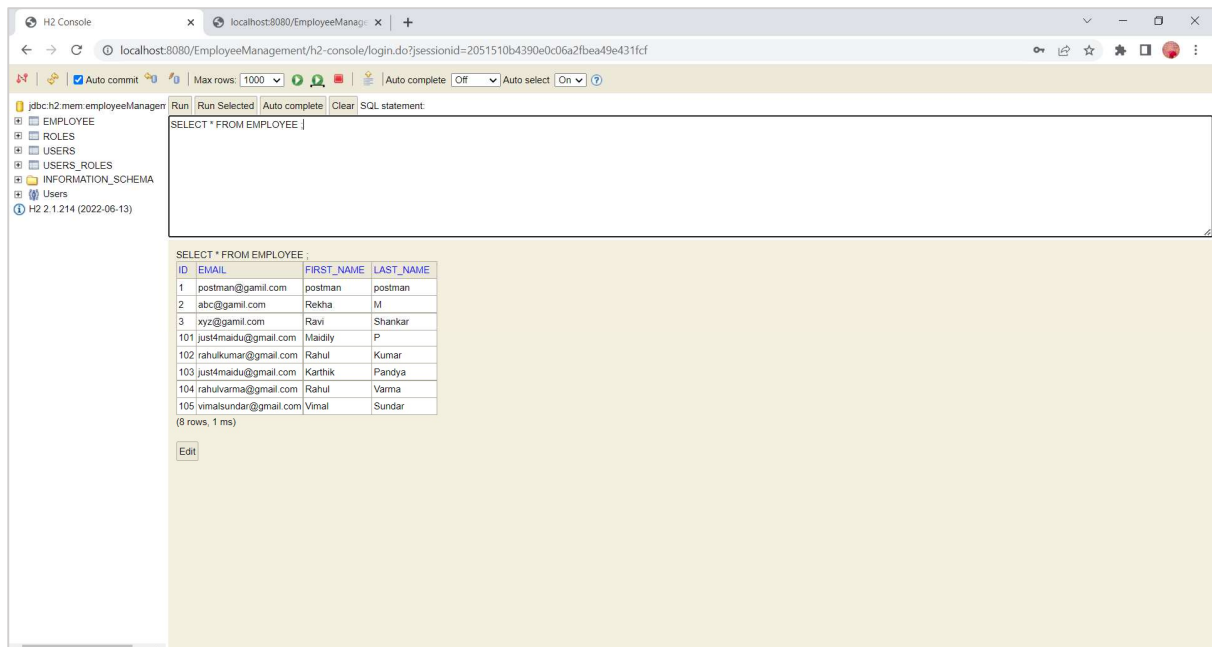




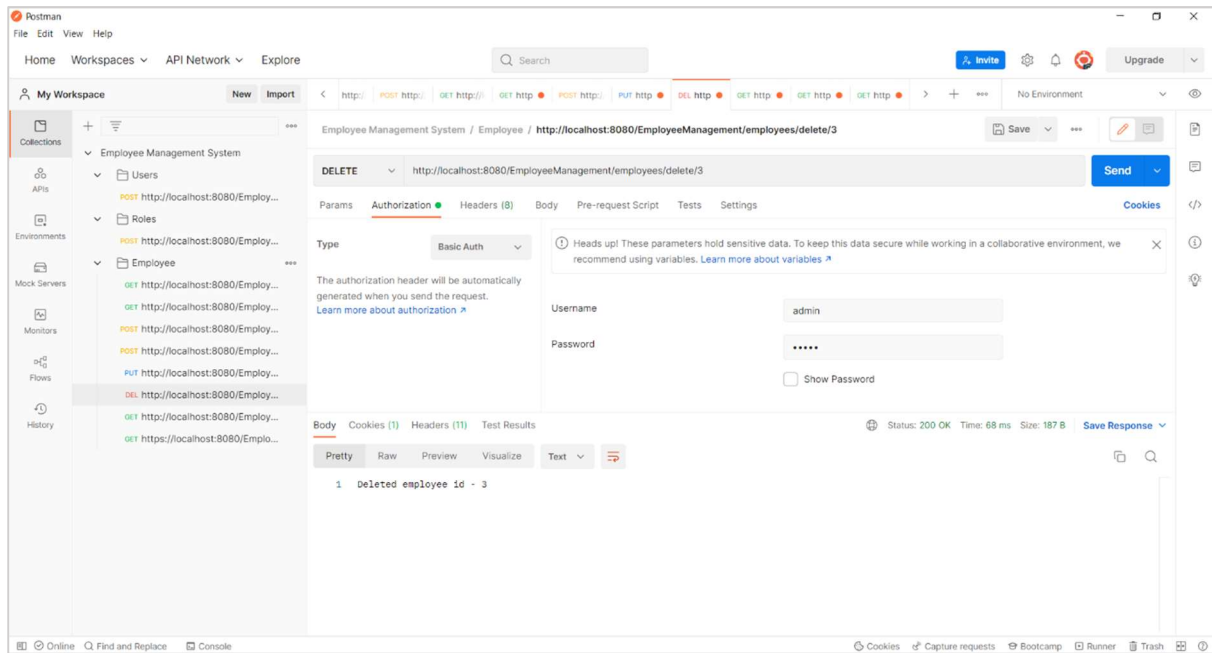
## Update existing Employee record by “id”: [id = 1]



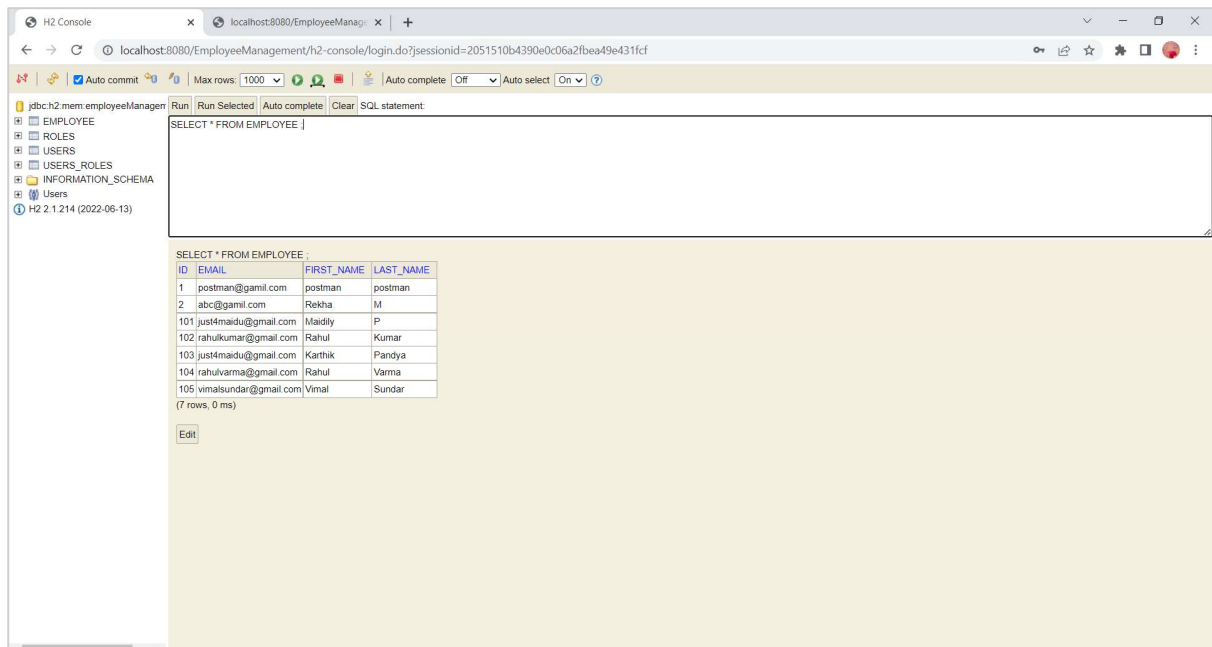
## Update action reflecting on H2-console: [id = 1] updated



## Delete existing Employee record from DB: [id = 3]



## Delete action reflecting on H2-console: [id = 3] removed



## Fetch/ Search Employee records using his/her First name: [firstName = Rahul]

The screenshot shows a Postman interface with a GET request to `http://localhost:8080/EmployeeManagement/employees/search/Rahul`. The request is authenticated with Basic Auth (Username: admin, Password: \*\*\*\*). The response status is 200 OK, and the body is a JSON array of two employee records.

```
1 {
2   "id": 192,
3   "firstName": "Rahul",
4   "lastName": "Kumar",
5   "email": "rahulkumar@gmail.com"
6 }
7
8 {
9   "id": 194,
10  "firstName": "Rahul",
11  "lastName": "Vazma",
12  "email": "rahulvazma@gmail.com"
13 }
14 }
```

## Fetch all Employee records sorted by First name – Ascending order:

The screenshot shows a GET request to `http://localhost:8080/EmployeeManagement/employees/sort?direction=ASC`. The request is authenticated with Basic Auth (Username: admin, Password: \*\*\*\*). The response status is 200 OK, and the body is a JSON array of three employee records sorted by first name in ascending order.

```
1 {
2   "id": 193,
3   "firstName": "Karthik",
4   "lastName": "Pandya",
5   "email": "justamaidu@gmail.com"
6 }
7
8 {
9   "id": 191,
10  "firstName": "Maidily",
11  "lastName": "P",
12  "email": "justamaidu@gmail.com"
13 }
14
15 {
16   "id": 1,
17   "firstName": "Postman",
18   "lastName": "postman",
19   "email": "postman@gmail.com"
20 }
```

## Fetch all Employee records sorted by First name – Descending order:

The screenshot shows the Postman interface with a GET request configured to fetch employee records sorted by first name in descending order. The request URL is `http://localhost:8080/EmployeeManagement/employees/sort?direction=DESC`. The response body is displayed in JSON format, showing three employee records sorted by first name in descending order.

**Request Details:**

- Method: GET
- URL: `http://localhost:8080/EmployeeManagement/employees/sort?direction=DESC`
- Query Params: 

KEY	VALUE	DESCRIPTION
direction	DESC	
Key	Value	Description

**Response Body (JSON):**

```
1 {
2   {
3     "id": 195,
4     "firstName": "Vimal",
5     "lastName": "Sunder",
6     "email": "vimalsunder@gmail.com"
7   },
8   {
9     "id": 2,
10    "firstName": "Rekha",
11    "lastName": "N",
12    "email": "abc@gmail.com"
13  },
14  {
15    "id": 192,
16    "firstName": "Rahul",
17    "lastName": "Kumar",
18  }
19 }
```