

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

# Login

<http://localhost:8080/login>  
Use admin for both un and pw, as startup login credentials

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login

200 676 ms

API Network

New Import

GET http://localhost:8080/login

Save

GET

http://localhost:8080/login

Send

Params

Auth

Headers (9)

Body

Pre-req.

Tests

Settings

Cookies

Type

Basic Auth

The authorization header will

Username

admin

Password

admin

Body

200 OK 676 ms 949 B

Save as example

Pretty

Raw

Preview

Visualize

Login

Username:

Password:

Log in

Postbot

Runner

Start Proxy

Cookies

Trash

Windows Taskbar

22:03 11-02-2024

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

## Roles

<http://localhost:8080/api/roles/>  
1. Your application should be able to add roles in the database dynamically in the db.

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login 200 676 ms

▶ POST http://localhost:8080/api/roles/ 201 231 ms

API Network

New Import

GET http:// GET http:// POST http:// POST http:// POS

+

No Environment

Save

Send

Send

Params Auth Headers (10) Body Pre-req. Tests Settings

raw JSON

1 {

2   "role": "ROLE\_ADMIN"

3 }

Body

201 Created 231 ms 356 B Save as example

Pretty Raw Preview Visualize JSON

1 { "id": 1, "name": "ROLE\_ADMIN" }

Postman Console

Postbot Runner Start Proxy Cookies Trash

22:07 11-02-2024

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

User Add

<http://localhost:8080/api/users/>  
2. Your application should be able to add Users in the db which can be used for authentication purposes. (ROLE\_ADMIN)

Postman Console

File Edit View Help

Search messages

All Logs

Clear

GET http://localhost:8080/login 200 676 ms

POST http://localhost:8080/api/roles/ 201 231 ms

POST http://localhost:8080/api/users/ 201 242 ms

API Network

New Import

GET http://localhost:8080/login

POST http://localhost:8080/api/roles/

POST http://localhost:8080/api/users/

POST http://localhost:8080/api/users/

Save

Send

Params Auth Headers (10) Body Pre-req. Tests Settings Cookies Beautify

raw JSON

```
1 {
2   "username": "temp1",
3   "password": "temp1",
4   "roles": [{
5     "id": 1,
6     "name": "ROLE_ADMIN"
7   }]
8 }
```

Body 201 Created 242 ms 476 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "username": "temp1",
4   "password": "{bcrypt}$2a$10$SRM5F9/tftzP7U.tBFx3DP.
5     NkDvGLYz4Jpn2CCJehwc5UHyzEzZG",
6   "roles": [
7     {
8       "id": 1,
9       "name": "ROLE_ADMIN"
10    }
11  ]
12 }
```

Skype - Signed in (Active)

Windows Taskbar

Start Menu Search Task View

File Explorer Edge Chrome Word Excel Outlook

Postman Runner Start Proxy Cookies Trash

ENG IN 22:10 11-02-2024

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

User Add

<http://localhost:8080/api/users/>  
2. Your application should be able to add Users in the db which can be used for authentication purposes. (ROLE\_USER)

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login20018 ms

▶ POST http://localhost:8080/api/roles/201581 ms

▶ POST http://localhost:8080/api/roles/20123 ms

▶ POST http://localhost:8080/api/users/201488 ms

▶ POST http://localhost:8080/api/users/201296 ms

Show timestampsHide network

http://localhost:8080/api/users/

POST

Send

ParamsAuthHeaders (10)BodyPre-reqTestsSettingsCookiesBeautify

rawJSON

```
1 {
2   "username": "temp2",
3   "password": "temp2",
4   "roles": [{
5     "id": 2,
6     "name": "ROLE_USER"
7   }]
8 }
```

Body201 Created296 ms475 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 2,
3   "username": "temp2",
4   "password": "{bcrypt}$2a$10$WpRW6uWHBd4Z/n.gcm90x0xNutiuk3BFbq0ccY/JV061sFPC.
5   "roles": [
6     {
7       "id": 2,
8       "name": "ROLE_USER"
9     }
10  ]
11 }
```

PostbotRunnerStart ProxyCookiesTrash

Page 4 | 16

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

## Logout

<http://localhost:8080/logout>  
Logout of the super admin user

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login200676 ms

▶ POST http://localhost:8080/api/roles/201231 ms

▶ POST http://localhost:8080/api/users/201242 ms

▶ POST http://localhost:8080/logout20027 ms

API Network

NewImport

GET http://localhost:8080/loginGET http://localhost:8080/api/roles/POST http://localhost:8080/api/users/POST http://localhost:8080/logout

HTTP http://localhost:8080/logout

POSThttp://localhost:8080/logout

Send

ParamsAuthHeaders (8)BodyPre-reqTestsSettingsCookies

Query Params

|  | Key | Value | Description | ... | Bulk Edit |
|--|-----|-------|-------------|-----|-----------|
|  | Key | Value | Description |     |           |

Body

200 OK27 ms477 BSave as example

PrettyRawPreviewVisualizeJSON

1{

2  "message": "You have been logged out successfully."

3}

PostbotRunnerStart ProxyCookiesTrash

22:1211-02-2024

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

## Login

<http://localhost:8080/login>  
Login of the newly created user with ROLE\_ADMIN

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login200676 ms

▶ POST http://localhost:8080/api/roles/201231 ms

▶ POST http://localhost:8080/api/users/201242 ms

▶ POST http://localhost:8080/logout20027 ms

▶ GET http://localhost:8080/login200572 ms

API Network

NewImport

GET http://localhost:8080/login

Save

</>

GET

http://localhost:8080/login

Send

ParamsAuthHeaders (9)BodyPre-req. TestsSettingsCookies

TypeBasic Auth

Username temp1

Password temp1

The authorization header will

Body

200 OK572 ms949 B

Save as example

PrettyRawPreviewVisualize

### Login

Username:

Password:

Log in

PostbotRunnerStart ProxyCookiesTrash

22:1511-02-2024

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

Employee Add

<http://localhost:8080/api/employees/>  
3. Now Your application should be able to add employees data in the db if and only if the authenticated user is ADMIN

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login 200 676 ms

▶ POST http://localhost:8080/api/roles/ 201 231 ms

▶ POST http://localhost:8080/api/users/ 201 242 ms

▶ POST http://localhost:8080/logout 200 27 ms

▶ GET http://localhost:8080/login 200 572 ms

▶ POST http://localhost:8080/api/employees/ 201 63 ms

▶ POST http://localhost:8080/api/employees/ 201 16 ms

▶ POST http://localhost:8080/api/employees/ 201 10 ms

▶ POST http://localhost:8080/api/employees/ 201 14 ms

▶ POST http://localhost:8080/api/employees/ 201 12 ms

API Network

New Import

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

Send

Params Auth Headers (10) Body Pre-req. Tests Settings Cookies Beautify

raw JSON

```
1 {
2   "firstName": "Sohini",
3   "lastName": "Ghosh",
4   "email": "sohini.ghosh@g3b7.com"
5 }
6
```

Body 201 Created 12 ms 408 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 5,
3   "firstName": "Sohini",
4   "lastName": "Ghosh",
5   "email": "sohini.ghosh@g3b7.com"
6 }
```

Postbot Runner Start Proxy Cookies Trash

Page 7 | 16

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

List Employees

<http://localhost:8080/api/employees/>  
4. Your application should provide an endpoint to list all the employees stored in the database.

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login200676 ms

▶ POST http://localhost:8080/api/roles/201231 ms

▶ POST http://localhost:8080/api/users/201242 ms

▶ POST http://localhost:8080/logout20027 ms

▶ GET http://localhost:8080/login200572 ms

▶ POST http://localhost:8080/api/employees/20163 ms

▶ POST http://localhost:8080/api/employees/20116 ms

▶ POST http://localhost:8080/api/employees/20110 ms

▶ POST http://localhost:8080/api/employees/20114 ms

▶ POST http://localhost:8080/api/employees/20112 ms

▶ GET http://localhost:8080/api/employees/20075 ms

API Network

NewImport

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

Save

GET

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

Send

ParamsAuthHeaders (8)BodyPre-req. TestsSettingsCookies

Query Params

| Key | Value | Description | Bulk Edit |
|-----|-------|-------------|-----------|
| Key | Value | Description |           |

Body

200 OK75 ms757 B

Save as example

PrettyRawPreviewVisualize

[{"id":2,"firstName":"Arnav","lastName":"Chatterjee","email":"arnav.chatterjee@g3b7.com"}, {"id":1,"firstName":"Arnav","lastName":"Banerjee","email":"arnav.banerjee@g3b7.com"}, {"id":4,"firstName":"Debashish","lastName":"Bose","email":"debashish.bose@g3b7.com"}, {"id":3,"firstName":"Rituparna","lastName":"Sengupta","email":"rituparna.sengupta@g3b7.com"}, {"id":5,"firstName":"Sohini","lastName":"Ghosh","email":"sohini.ghosh@g3b7.com"}]

PostbotRunnerStart ProxyCookiesTrash

22:2611-02-2024



# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

## Fetch Employee

<http://localhost:8080/api/employees/3>

5. Your application should provide endpoint to fetch an employee record specifically based on the id of that employee

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login200676 ms

▶ POST http://localhost:8080/api/roles/201231 ms

▶ POST http://localhost:8080/api/users/201242 ms

▶ POST http://localhost:8080/logout20027 ms

▶ GET http://localhost:8080/login200572 ms

▶ POST http://localhost:8080/api/employees/20163 ms

▶ POST http://localhost:8080/api/employees/20116 ms

▶ POST http://localhost:8080/api/employees/20110 ms

▶ POST http://localhost:8080/api/employees/20114 ms

▶ POST http://localhost:8080/api/employees/20112 ms

▶ GET http://localhost:8080/api/employees/20075 ms

▶ GET http://localhost:8080/api/employees/320046 ms

API Network

NewImport

GET http://localhost:8080/api/employees/3

Save

GET

http://localhost:8080/api/employees/3

Send

Params

Auth

Headers (8)

Body

Pre-req.

Tests

Settings

Cookies

Query Params

| Key | Value | Description | Bulk Edit |
|-----|-------|-------------|-----------|
| Key | Value | Description |           |

Body

200 OK46 ms415 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": 3,
3   "firstName": "Rituparna",
4   "lastName": "Sengupta",
5   "email": "rituparna_sengupta@g3b7.com"
6 }
```

No action required

Postman Runner

Start Proxy

Cookies

Trash

Windows Taskbar

Windows Start

Search

Taskbar Icons

System Tray

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

## Update Employee

<http://localhost:8080/api/employees/>

Your application should provide an endpoint to update an existing employee with the given updated json object

The image displays two screenshots of a database console (H2 Console) and a REST client, illustrating the process of updating an employee.

**Before Update:** The top screenshot shows the H2 Console with the SQL statement `SELECT * FROM EMPMG.TEMPLOYEE;` executed. The result set shows three employees:

| ID | EMAIL                          | FIRST_NAME | LAST_NAME  |
|----|--------------------------------|------------|------------|
| 1  | tempoylee.templeyer@g3b7.com   | tempoylee  | templeyer  |
| 2  | tempoylee.templeyer@g3b7.com   | tempoylee2 | templeyer2 |
| 3  | tanushya.agarwal@tempoylee.com | Tanushya   | Agarwal    |

A purple arrow points to the third row, indicating the employee to be updated.

**After Update:** The bottom screenshot shows the H2 Console with the same SQL statement executed. The result set now shows the updated employee:

| ID | EMAIL                                | FIRST_NAME       | LAST_NAME       |
|----|--------------------------------------|------------------|-----------------|
| 1  | tempoylee.templeyer@g3b7.com         | tempoylee        | templeyer       |
| 2  | tempoylee.templeyer@g3b7.com         | tempoylee2       | templeyer2      |
| 3  | tanushya.agarwal@updatedemployee.com | Tanushya-updated | Agarwal-updated |

A green arrow points to the third row, indicating the updated employee.

**REST Client:** The right screenshot shows a REST client interface with the endpoint `http://localhost:8080/api/employees/` and the method `PUT`. The request body is a JSON object:

```
{
  "id": 3,
  "firstName": "Tanushya-updated",
  "lastName": "Agarwal-updated",
  "email": "tanushya.agarwal@updatedemployee.com"
}
```

The response status is `200 OK` with a response time of `12 ms` and a response size of `438 B`.

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

## Delete Employee

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

7. Your application to have endpoint to delete an existing employee record based on the id of the employee

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login 200 535 ms

▶ POST http://localhost:8080/api/roles/ 201 203 ms

▶ POST http://localhost:8080/api/employees/ 201 27 ms

⚠ "Binary file load error: invalid or missing file source"

▶ DELETE http://localhost:8080/api/employees/1 200 54 ms

1 warning

API Network

New Import

POST http: DEL http: GET http: POST http: PUT

No Environment

Save

Send

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

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Query Params

| Key | Value | Description | Bulk Edit |
|-----|-------|-------------|-----------|
| Key | Value | Description |           |

Body

200 OK 54 ms 342 B Save as example

Pretty Raw Preview Visualize Text

1 Deleted user id - 1

Postbot Runner Start Proxy Cookies Trash

23:01 11-02-2024

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

First name Fetch

<http://localhost:8080/api/employees/by-first-name/arn>  
8. Your application should fetch employee by his/her first name and if found more than one record then list them all

Postman Console

File Edit View Help

Search messages All Logs Clear

▶ GET http://localhost:8080/login 200 676 ms

▶ POST http://localhost:8080/api/roles/ 201 231 ms

▶ POST http://localhost:8080/api/users/ 201 242 ms

▶ POST http://localhost:8080/logout 200 27 ms

▶ GET http://localhost:8080/login 200 572 ms

▶ POST http://localhost:8080/api/employees/ 201 63 ms

▶ POST http://localhost:8080/api/employees/ 201 16 ms

▶ POST http://localhost:8080/api/employees/ 201 10 ms

▶ POST http://localhost:8080/api/employees/ 201 14 ms

▶ POST http://localhost:8080/api/employees/ 201 12 ms

▶ GET http://localhost:8080/api/employees/ 200 75 ms

▶ GET http://localhost:8080/api/employees/3 200 46 ms

▶ PUT http://localhost:8080/api/employees/3 200 82 ms

▶ PUT http://localhost:8080/api/employees/3 200 21 ms

▶ POST http://localhost:8080/api/employees/ 201 21 ms

▶ PUT http://localhost:8080/api/employees/3 200 19 ms

⚠ "Binary file load error: invalid or missing file source"

▶ DELETE http://localhost:8080/api/employees/6 204 92 ms

▶ GET http://localhost:8080/api/employees/by-first-name/arn 200 194 ms

1 warning

API Network

New Import

http://localhost:8080/api/employees/by-first-name/arn

GET http://localhost:8080/api/employees/by-first-name/arn

Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Query Params

| Key | Value | Description | Bulk Edit |
|-----|-------|-------------|-----------|
| Key | Value | Description |           |

Body

200 OK 194 ms 585 B Save as example

Pretty Raw Preview Visualize

[{"id":1,"firstName":"Arnav","lastName":"Banerjee","email":"arnav.banerjee@g3b7.com"}, {"id":2,"firstName":"Arnab","lastName":"Chatterjee","email":"arnab.chatterjee@g3b7.com"}, {"id":3,"firstName":"Tarnya","lastName":"Bhatti","email":"tarnya.bhatti@employee.com"}]

Page 12 | 16

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

## Employee Sort List

<http://localhost:8080/api/employees/?sort=asc>

9a. Your application should be able to list all employee records sorted on their first name in ascending order.

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login 200 676 ms

▶ POST http://localhost:8080/api/roles/ 201 231 ms

▶ POST http://localhost:8080/api/users/ 201 242 ms

▶ POST http://localhost:8080/logout 200 27 ms

▶ GET http://localhost:8080/login 200 572 ms

▶ POST http://localhost:8080/api/employees/ 201 63 ms

▶ POST http://localhost:8080/api/employees/ 201 16 ms

▶ POST http://localhost:8080/api/employees/ 201 10 ms

▶ POST http://localhost:8080/api/employees/ 201 14 ms

▶ POST http://localhost:8080/api/employees/ 201 12 ms

▶ GET http://localhost:8080/api/employees/ 200 75 ms

▶ GET http://localhost:8080/api/employees/3 200 46 ms

▶ PUT http://localhost:8080/api/employees/3 200 82 ms

▶ PUT http://localhost:8080/api/employees/3 200 21 ms

▶ POST http://localhost:8080/api/employees/ 201 21 ms

▶ PUT http://localhost:8080/api/employees/3 200 19 ms

⚠ "Binary file load error: invalid or missing file source"

▶ DELETE http://localhost:8080/api/employees/6 204 92 ms

▶ GET http://localhost:8080/api/employees/by-first-name/arn 200 194 ms

▶ GET http://localhost:8080/api/employees/?sort=asc 200 22 ms

1 warning

API Network

New Import

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

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

Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Query Params

| <input checked="" type="checkbox"/> | Key  | Value | Description | Bulk Edit |
|-------------------------------------|------|-------|-------------|-----------|
| <input checked="" type="checkbox"/> | sort | asc   |             |           |
|                                     | Key  | Value | Description |           |

Body

200 OK 22 ms 751 B Save as example

Pretty Raw Preview Visualize

[{"id":2,"firstName":"Arnab","lastName":"Chatterjee","email":"arnab.chatterjee@g3b7.com"}, {"id":1,"firstName":"Arnav","lastName":"Banerjee","email":"arnav.banerjee@g3b7.com"}, {"id":4,"firstName":"Debashish","lastName":"Bose","email":"debashish.bose@g3b7.com"}, {"id":5,"firstName":"Sohini","lastName":"Ghosh","email":"sohini.ghosh@g3b7.com"}, {"id":3,"firstName":"Tanya","lastName":"Bhatti","email":"tanya.bhatti@employee.com"}]



# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

## Employee Sort List

<http://localhost:8080/api/employees/?sort=desc>

9a. Your application should be able to list all employee records sorted on their first name in descending order.

Postman Console

File Edit View Help

Search messages

All Logs

Clear

▶ GET http://localhost:8080/login 200 676 ms

▶ POST http://localhost:8080/api/roles/ 201 231 ms

▶ POST http://localhost:8080/api/users/ 201 242 ms

▶ POST http://localhost:8080/logout 200 27 ms

▶ GET http://localhost:8080/login 200 572 ms

▶ POST http://localhost:8080/api/employees/ 201 63 ms

▶ POST http://localhost:8080/api/employees/ 201 16 ms

▶ POST http://localhost:8080/api/employees/ 201 10 ms

▶ POST http://localhost:8080/api/employees/ 201 14 ms

▶ POST http://localhost:8080/api/employees/ 201 12 ms

▶ GET http://localhost:8080/api/employees/ 200 75 ms

▶ GET http://localhost:8080/api/employees/3 200 46 ms

▶ PUT http://localhost:8080/api/employees/3 200 82 ms

▶ PUT http://localhost:8080/api/employees/3 200 21 ms

▶ POST http://localhost:8080/api/employees/ 201 21 ms

▶ PUT http://localhost:8080/api/employees/3 200 19 ms

⚠ "Binary file load error: invalid or missing file source"

▶ DELETE http://localhost:8080/api/employees/6 204 92 ms

▶ GET http://localhost:8080/api/employees/by-first-name/arn 200 194 ms

▶ GET http://localhost:8080/api/employees/?sort=asc 200 22 ms

▶ GET http://localhost:8080/api/employees/?sort=desc 200 30 ms

1 warning

API Network

New Import

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

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

Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Query Params

| <input checked="" type="checkbox"/> | Key  | Value | Description | Bulk Edit |
|-------------------------------------|------|-------|-------------|-----------|
| <input checked="" type="checkbox"/> | sort | desc  |             |           |
|                                     | Key  | Value | Description |           |

Body

200 OK 30 ms 751 B Save as example

Pretty Raw Preview Visualize

[{"id":3,"firstName":"Tarnya","lastName":"Bhatti","email":"tarnya.bhatti@employee.com"}, {"id":5,"firstName":"Sohini","lastName":"Ghosh","email":"sohini.ghosh@g3b7.com"}, {"id":4,"firstName":"Debashish","lastName":"Bose","email":"debashish.bose@g3b7.com"}, {"id":1,"firstName":"Arnav","lastName":"Banerjee","email":"arnav.banerjee@g3b7.com"}, {"id":2,"firstName":"Arnab","lastName":"Chatterjee","email":"arnab.chatterjee@g3b7.com"}]

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

ROLE\_USER

<http://localhost:8080/api/employees/>  
Establishing that a login with a ROLE\_USER is "Forbidden".

Postman Console

File Edit View Help

Search messages All Logs Clear

▶ GET http://localhost:8080/login20018 ms

▶ POST http://localhost:8080/api/roles/201581 ms

▶ POST http://localhost:8080/api/roles/20123 ms

▶ POST http://localhost:8080/api/users/201488 ms

▶ POST http://localhost:8080/api/users/201296 ms

▶ POST http://localhost:8080/api/employees/20194 ms

▶ POST http://localhost:8080/api/employees/20122 ms

▶ POST http://localhost:8080/api/employees/20121 ms

▶ PUT http://localhost:8080/api/employees/20070 ms

▶ POST http://localhost:8080/logout20088 ms

▶ GET http://localhost:8080/login200716 ms

▶ POST http://localhost:8080/api/employees/403210 ms

GEThttp://localhost:8080/loginSend

ParamsAuthHeaders (9)BodyPre-req. Tests SettingsCookies

TypeBasic Auth

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

Usernametemp2

Passwordtemp2

temp2 User with ROLE\_USER

Show timestampsHide network

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

POSThttp://localhost:8080/api/employees/Send

ParamsAuthHeaders (10)BodyPre-req. Tests SettingsCookies

rawJSON

```
1 {
2   "firstName": "Try-with-ROLE_USER",
3   "lastName": "Try-with-ROLE_USER",
4   "email": "try.with.role.user@g3b7.com"
5 }
6
```

Body

403 Forbidden210 ms446 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "timestamp": "2024-02-12T16:22:46.298+00:00",
3   "status": 403,
4   "error": "Forbidden",
5   "message": "",
6   "path": "/api/employees/"
7 }
```

Page 15 | 16

# GLG3B7Aug23\_EmployeeManagement\_Screenshots.docx

## H2 Database

H2 Console

H2 showing all the relevant databases

The screenshot shows the H2 Console web interface in a browser. The address bar displays the URL: `localhost:8080/h2-console/login.do?jsessionid=b8e18a5204cebe3d7af3ea0eedb...`. The interface includes a sidebar on the left with a tree view of the database structure, a central SQL editor, and a results pane at the bottom.

**Database Structure (Left Sidebar):**

- jdbc:h2:mem:empmgmt
  - EMPMGT
    - APP\_USER
      - ID
      - PASSWORD
      - USERNAME
      - Indexes
    - APP\_USER\_ROLE
      - USER\_ID
      - ROLE\_ID
      - Indexes
    - EMPLOYEE
      - ID
      - EMAIL
      - FIRST\_NAME
      - LAST\_NAME
      - Indexes
    - ROLE
      - ID
      - NAME
      - Indexes
  - INFORMATION\_SCHEMA
  - Sequences
  - Users

**SQL Editor (Center):**

SQL statement: `SELECT * FROM EMPMGTEMPLOYEE`

**Query Results (Bottom):**

SELECT \* FROM EMPMGTEMPLOYEE;

| ID | EMAIL                      | FIRST_NAME | LAST_NAME  |
|----|----------------------------|------------|------------|
| 1  | arnav.banerjee@g3b7.com    | Arnav      | Banerjee   |
| 2  | arnab.chatterjee@g3b7.com  | Arnab      | Chatterjee |
| 3  | tarnya.bhatti@employee.com | Tarnya     | Bhatti     |
| 4  | debashish.bose@g3b7.com    | Debashish  | Bose       |
| 5  | sohini.ghosh@g3b7.com      | Sohini     | Ghosh      |

(5 rows, 6 ms)