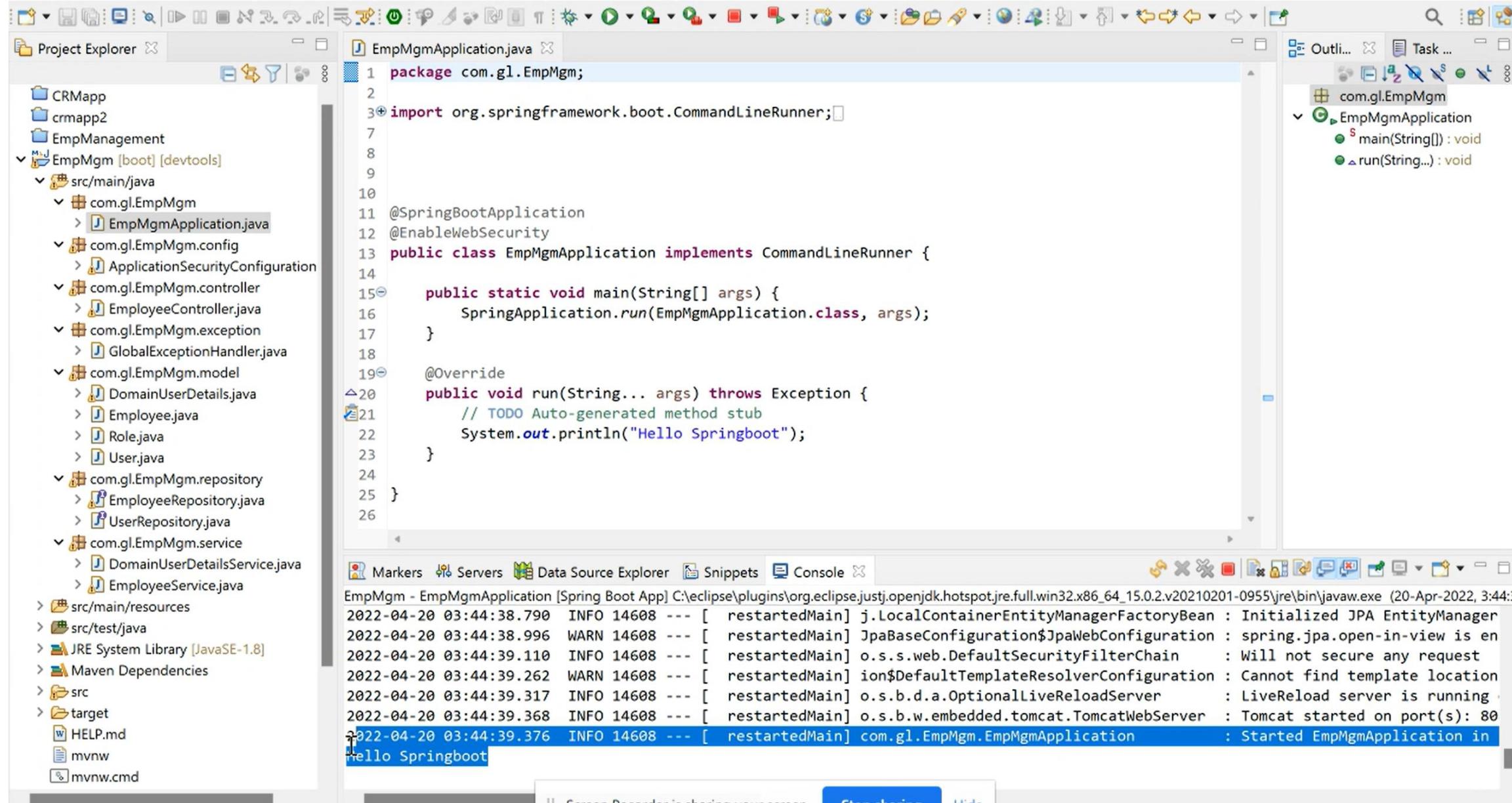


1. Running Spring Boot app

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace2 - EmpMgm/src/main/java/com/gl/EmpMgm/EmpMgmApplication.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar with various icons for file operations, search, and project management.
- Project Explorer:** Shows the project structure under EmpMgm [boot] [devtools].
- Code Editor:** Displays the EmpMgmApplication.java code, which includes annotations like @SpringBootApplication and @EnableWebSecurity, and implements the CommandLineRunner interface.
- Outline View:** Shows the class structure with EmpMgmApplication highlighted.
- Bottom Status Bar:** Shows "Screen Recorder is sharing your screen." with "Stop sharing" and "Hide" buttons.
- Taskbar:** Shows two taskbars at the bottom. The left one has icons for weather (23°C Mostly cloudy), search, messaging, file, browser, and others. The right one shows system status (ENG IN, 03:44 AM, 20-04-2022) and a notification count of 6.



localhost:8082/h2-console/login.jsp?jsessionid=beb21f95b43db68b53113edd795d08cc



English Preferences Tools Help

Login

Saved Settings:

Setting Name:

Driver Class:

JDBC URL:

User Name:

Password:

|| Screen Recorder is sharing your screen.

23°C
Mostly cloudy



ENG IN 03:45 AM 20-04-2022 6

23°C
Mostly cloudy



ENG IN 04:13 AM 20-04-2022 7

2. Adding Users and roles in H2 database

The screenshot shows the H2 Console interface running in a browser window. The URL is `localhost:8082/h2-console/login.do?jsessionid=beb21f95b43db68b53113edd795d08cc`. The left sidebar lists database objects: EMPLOYEES, ROLES, USERS, USERS_ROLES, INFORMATION_SCHEMA, Sequences, and Users. The main area contains the following SQL statements:

```
insert into users values (1, 'kiran', 'welcome');
insert into users values (2, 'vinay', 'welcome');

insert into roles values (1, 'USER');
insert into roles values (2, 'ADMIN');

insert into users_roles values (1, 1);
insert into users_roles values (1, 2);
insert into users_roles values (2, 2);
```

Below the SQL statements, there is a section titled "Important Commands" with the following key mappings:

	Displays this Help Page
	Shows the Command History
	Executes the current SQL statement
	Executes the SQL statement defined by the text selection
	Ctrl+Space Auto complete
	Disconnects from the database

There is also a "Sample SQL Script" section with a table containing various database operations like creating tables, inserting data, and updating records.

At the bottom, a message says "Screen Recorder is sharing your screen." and a "Stop sharing" button is visible. The system tray shows the date and time as 03:44 AM 20-04-2022, and the taskbar includes icons for File Explorer, Task View, Mail, Photos, and other Microsoft applications.

screen-capture (2)

00:01:29

23°C
Mostly cloudy



04:14 AM
20-04-2022

7

localhost:8082/h2-console/login.do?jsessionid=beb21f95b43db68b53113edd795d08cc



Auto commit

Max rows: 1000



Auto complete



Auto select



jdbc:h2:mem:emp_db

EMPLOYEES

ROLES

USERS

USERS_ROLES

INFORMATION_SCHEMA

Sequences

Users

H2 1.4.200 (2019-10-14)

Run

Run Selected

Auto complete

Clear

SQL statement:

```
insert into users_roles values (1, 1);
insert into users_roles values (1, 2);
Insert into users_roles values (2, 2);
```

```
select * from users;
select * from roles;
select * from users_roles;
```

```
select * from users;
```

USER_ID	NAME	PASSWORD
1	kiran	welcome
2	vinay	welcome

(2 rows, 0 ms)

Edit

|| Screen Recorder is sharing your screen.

Stop sharing

Hide

23°C
Mostly cloudy

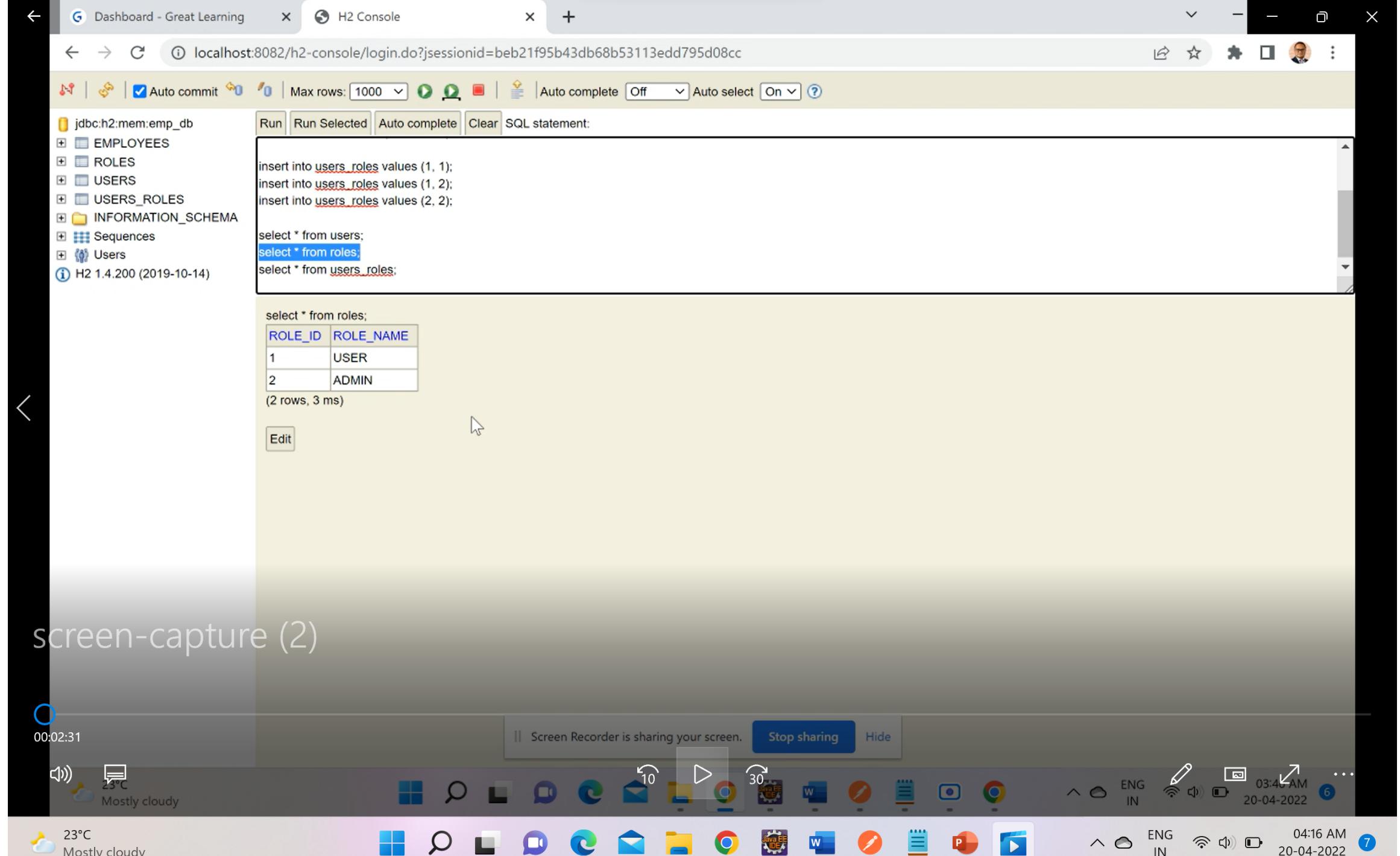


ENG IN 03:46 AM 20-04-2022 6

23°C
Mostly cloudy



ENG IN 04:15 AM 20-04-2022 7



Dashboard - Great Learning

H2 Console

localhost:8082/h2-console/login.do?jsessionid=beb21f95b43db68b53113edd795d08cc

Auto commit | Max rows: 1000 | Auto complete | Off | Auto select | On | ?

jdbc:h2:mem:emp_db | Run | Run Selected | Auto complete | Clear | SQL statement:

EMPLOYEES
ROLES
USERS
USERS_ROLES
INFORMATION_SCHEMA
Sequences
Users
H2 1.4.200 (2019-10-14)

```
insert into users_roles values (1, 1);
insert into users_roles values (1, 2);
Insert into users_roles values (2, 2);

select * from users;
select * from roles;
select * from users_roles;
```

select * from users_roles;

ROLE_ID	USER_ID
1	1
1	2
2	2

(3 rows, 0 ms)

Edit

00:02:39

Screen Recorder is sharing your screen.

Stop sharing Hide

23°C Mostly cloudy

Java EE IDE

ENG IN

03:46 AM 20-04-2022

23°C Mostly cloudy

ENG IN

04:16 AM 20-04-2022

2. “vinay” has GET access

The screenshot shows the Postman application interface. A request to `localhost:8082/api/employees` is being made via a `GET` method. The `Authorization` tab is selected, showing basic authentication details: `Type: Basic Auth`, `Username: vinay`, and `Password: welcome`. The response status is `200 OK` with a time of `331 ms` and a size of `325 B`. The response body is currently empty, indicated by the number `1` and an empty JSON object `[]`.

Postman UI Elements:

- File Edit View Help
- Home Workspaces API Network Reports Explore
- Search Postman
- Cloud Sync Invite Settings Notifications Upgrade

Request Details:

- Method: GET
- URL: `localhost:8082/api/employees`
- Authorization Type: Basic Auth
- Username: vinay
- Password: welcome
- Show Password checked

Response Status:

- Status: 200 OK
- Time: 331 ms
- Size: 325 B

Response Body:

- Pretty
- Raw
- Preview
- Visualize
- JSON

Bottom Status Bar:

- 00:03:11
- Find and Replace
- Console
- Screen Recorder is sharing your screen. Stop sharing
- Hide
- Cookies
- Capture requests
- Bootcamp
- Runner
- Trash

System Tray:

- 23°C Mostly cloudy
- 25°C Mostly cloudy
- 03:41 AM 20-04-2022
- ENG IN
- 04:17 AM 20-04-2022
- 7

3. “Kiran” has GET access

The screenshot shows the Postman application interface. A GET request is being made to `localhost:8082/api/employees`. The 'Authorization' tab is selected, showing 'Basic Auth' selected with 'Username' set to `kir'an` and 'Password' set to `welcome`. The response status is `200 OK` with a size of `325 B`.

Below the request details, the response body is displayed as a JSON object:

```
[{"id": 1, "name": "John Doe", "age": 30, "city": "New York"}]
```

The task bar at the bottom of the screen shows various application icons and system status.

4. “Kiran” does not have POST access

The screenshot shows the Postman application interface. A request is being made to `localhost:8082/api/employees` using the `POST` method. The `Authorization` tab is selected, showing `Basic Auth` with `Username: kirin` and `Password: welcome`. The response status is `403 Forbidden`, indicating that the user lacks sufficient permissions. The response body is a JSON object:

```
1 "timestamp": "2022-04-19T22:17:55.373+00:00",
2 "status": 403,
3 "error": "Forbidden",
4 "message": "Forbidden",
5 "path": "/api/employees"
```

The Postman interface includes a sidebar with sections like Overview, Home, Workspaces, API Network, Reports, Explore, and a search bar at the top. The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

5. “Vinay” has POST access

The screenshot shows the Postman application interface. At the top, the title bar says "Postman". Below it is a navigation bar with "File", "Edit", "View", "Help", "Home", "Workspaces", "API Network", "Reports", "Explore", and a search bar "Search Postman". To the right of the search bar are icons for "Invite", "Settings", "Bell", and "Upgrade".

The main workspace shows a "POST" request to "localhost:8082/api/employees". The "Body" tab is selected, showing JSON input:

```
1
2   ...
3     "firstname": "aaaa",
4     "lastname": "saha",
5     "email": "us1@gmail.com"
6
```

The response status is "Status: 200 OK" with "Time: 172 ms" and "Size: 395 B". The response body is also displayed in JSON format:

```
1
2   {
3     "empid": 1,
4     "firstname": "aaaa",
5     "lastname": "saha",
6     "email": "us1@gmail.com"
7}
```

At the bottom of the screen, there is a taskbar with various icons for file operations, system status, and date/time information.

5. Listing all employees – Both “Vinay”& “Kiran” has GET access

The screenshot shows the Postman application interface. In the top navigation bar, the title "Postman" is visible along with "File Edit View Help", "Home", "Workspaces", "API Network", "Reports", and "Explore". A search bar "Search Postman" is present, along with icons for cloud, invite, settings, and notifications. A "Upgrade" button is also visible.

The main workspace displays a collection named "localhost:8082/api/employees". A specific request is selected: "GET localhost:8082/api/employees". The "Body" tab is active, showing a JSON payload:

```
1
2   ...
3     "firstname": "dddd",
4     "lastname": "saha",
5     "email": "us4@gmail.com"
6
```

The "Params", "Authorization", and "Headers" tabs are also visible but not selected. Below the body, the response status is shown as "Status: 200 OK Time: 26 ms Size: 616 B". The "Save Response" button is available.

The "Body" section contains the raw JSON response from the API, which includes two employee records:

```
1
2   [
3     {
4       "empid": 3,
5       "firstname": "cccc",
6       "lastname": "saha",
7       "email": "us2@gmail.com"
8     },
9     {
10       "empid": 4,
11       "firstname": "dddd",
12       "lastname": "saha",
13     }
14 ]
```

At the bottom of the screen, the taskbar shows various pinned icons and system status indicators, including the date and time (20-04-2022, 03:42 AM), battery level (6), signal strength, and weather information (23°C, Mostly cloudy).

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman Invite Upgrade

No Environment

localhost:8082/api/employees

GET localhost:8082/api/employees

Params Authorization Headers (10) Body Tests Settings Cookies Beautify

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

```
1 {  
2   "firstname": "dddd",  
3   "lastname": "saha",  
4   "email": "us4@gmail.com"  
5 }  
6
```

Status: 200 OK Time: 26 ms Size: 616 B Save Response

Body Cookies (1) Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
16   "firstname": "aaaa",  
17   "lastname": "saha",  
18   "email": "us1@gmail.com"  
19 },  
20 {  
21   "empid": 2,  
22   "firstname": "bbbb",  
23   "lastname": "saha",  
24   "email": "us2@gmail.com"  
25 }  
26
```

00:05:10 Find and Replace Console Stop sharing Hide Cookies Capture requests Bootcamp Runner Trash

Screen Recorder is sharing your screen.

23°C Mostly cloudy ENG IN 03:45 AM 20-04-2022

Java EE IDE W P 04:24 AM 20-04-2022

6. Kiran through GET sees the list of employees

The screenshot shows the Postman application interface. In the top navigation bar, the URL `localhost:8082/api/employees` is entered under the 'GET' method. The 'Authorization' tab is selected, showing 'Basic Auth' configured with 'Username' `kiran` and 'Password' `welcome`. The response status is `200 OK` with a size of `616 B`. The JSON response body is displayed in the 'Pretty' format:

```
1  {
2   "empid": 3,
3   "firstname": "cccc",
4   "lastname": "saha",
5   "email": "us2@gmail.com"
6 },
7  {
8   "empid": 4,
9   "firstname": "dddd",
10  "lastname": "saha",
11 }
```

The bottom status bar indicates the screen is being recorded.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman Invite Upgrade

No Environment

PUT localhost:8082/api/employees

localhost:8082/api/employees

PUT localhost:8082/api/employees Send

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

Type Basic Auth Username kiran

The authorization header will be automatically generated when you send the request.

Learn more about authorization ↗

Password welcome Show Password

Body Cookies (1) Headers (10) Test Results Status: 200 OK Time: 8 ms Size: 616 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "empid": 3,  
3   "firstname": "cccc",  
4   "lastname": "saha",  
5   "email": "us2@gmail.com"  
6 },  
7 {  
8   "empid": 4,  
9   "firstname": "dddd",  
10  "lastname": "saha",  
11 }
```

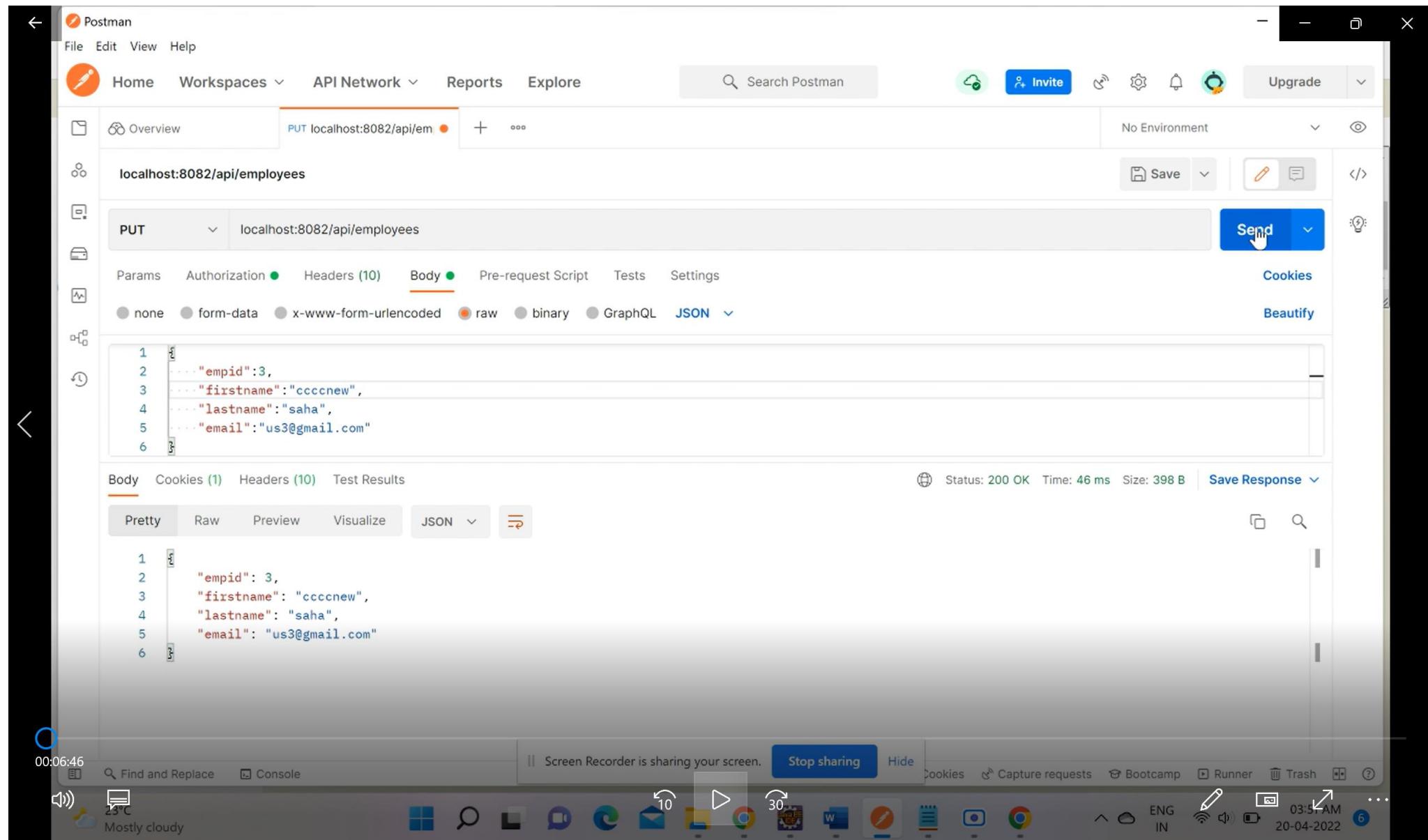
00:06:03 Find and Replace Console Stop sharing Hide Cookies Capture requests Bootcamp Runner Trash

Screen Recorder is sharing your screen.

23°C Mostly cloudy ENG IN 03:50 AM 20-04-2022

23°C Mostly cloudy ENG IN 04:26 AM 20-04-2022

7. Updating an employee through PUT – Both “Vinay” & “Kiran” has access



8. Listing a specific employee – Both “Vinay” & “Kiran” GET access

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help' options. Below the navigation bar is a header with 'Home', 'Workspaces', 'API Network', 'Reports', and 'Explore' buttons. A search bar says 'Search Postman'. To the right of the search bar are icons for 'Invite', 'Settings', and 'Upgrade'. The main workspace shows an 'Overview' card with a green 'localhost:8082/api/employees/2' status indicator. Below it, a request card for 'localhost:8082/api/employees/2' is displayed with a 'GET' method. The 'Body' tab is selected, showing JSON input:

```
1 ... "empid":3,
2 ... "firstname":"ccccnew",
3 ... "lastname":"saha",
4 ... "email":"us3@gmail.com"
5
6
```

Below the input, the response section shows a status of 'Status: 200 OK' and a response body:

```
1
2   "empid": 2,
3   "firstname": "bbbb",
4   "lastname": "saha",
5   "email": "us2@gmail.com"
6
```

The bottom of the screen shows a system tray with various icons and a weather forecast for 'Mostly cloudy' at 23°C.

9. Searching an employee by firstname – Both “Vinay”& “Kiran” has access

The screenshot shows the Postman application interface. A GET request is made to `localhost:8082/api/employees/search/dddd`. The Authorization tab is selected, using Basic Auth with Username `kiran` and Password `welcome`. The response body is displayed in JSON format:

```
1 [ { 2   "empid": 4, 3   "firstname": "dddd", 4   "lastname": "saha", 5   "email": "us4@gmail.com" 6 } ] 7 8 ]
```

The status bar at the bottom indicates `Status: 200 OK Time: 47 ms Size: 397 B`.

9. Deleting an employee by employee id – Both “Vinay” & “Kiran” has access

The screenshot shows the Postman application interface. A DELETE request is being made to `localhost:8082/api/employees/3`. The request is set up with Basic Auth credentials: Username `kiran` and Password `welcome`. The response status is `200 OK`.

Request Details:

- Method: `DELETE`
- URL: `localhost:8082/api/employees/3`
- Authorization Type: `Basic Auth`
- Username: `kiran`
- Password: `welcome`

Response Status:

- Status: `200 OK`
- Time: `20 ms`
- Size: `282 B`

Postman Status Bar:

- 00:08:20
- Screen Recorder is sharing your screen.
- Stop sharing
- Hide
- Cookies
- Capture requests
- Bootcamp
- Runner
- Trash
- ...
- ENG IN
- 20-04-2022
- 03:52 AM
- 22°C
- Mostly cloudy

Taskbar Icons:

- Windows Start
- Search
- File Explorer
- Mail
- OneDrive
- Google Chrome
- Java IDE
- Word
- Excel
- PowerPoint
- Teams
- Play Store
- Cloud Storage
- ENG IN
- Wi-Fi
- Battery
- 04:55 AM
- 20-04-2022

11. Sorted list in ascending order – Both “Vinay”& “Kiran” has access

The screenshot shows the Postman application interface. A GET request is being made to `localhost:8082/api/employees/sort?order=asc`. The Authorization tab is selected, showing 'Basic Auth' with 'Username' set to 'kiran' and 'Password' set to 'welcome'. The response status is 200 OK, with a size of 543 B. The JSON response body is displayed as:

```
1  [
2    {
3      "empid": 1,
4      "firstname": "aaaa",
5      "lastname": "saha",
6      "email": "aaa@gmail.com"
7    },
8    {
9      "empid": 2,
10     "firstname": "bbbb",
11     "lastname": "saha",
12   }
```

A watermark 'screen-capture (2)' is overlaid on the left side of the interface.

12. Sorting all employees by descending order – Both “Vinay” & “Kiran” has access

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help' options. Below the navigation bar is a toolbar with icons for 'Home', 'Workspaces', 'API Network', 'Reports', 'Explore', a search bar 'Search Postman', and various settings and sharing buttons.

The main workspace displays a request configuration for a GET request to 'localhost:8082/api/employees/sort?order=desc'. The 'Authorization' tab is selected, showing 'Basic Auth' type with 'Username' 'kiran' and 'Password' 'welcome'. A note indicates that the authorization header will be automatically generated when the request is sent. There are also tabs for 'Params', 'Headers (10)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'.

Below the request configuration, the response section is visible. It shows a status of '200 OK', a response time of '14 ms', and a response size of '543 B'. The response body is displayed in 'Pretty' JSON format:

```
1  {
2     "empid": 4,
3     "firstname": "dddd",
4     "lastname": "saha",
5     "email": "aaaa@gmail.com"
6
7     {
8         "empid": 2,
9         "firstname": "bbbb",
10        "lastname": "saha",
11    }

```

At the bottom of the screen, there's a taskbar with various icons for system functions like volume, brightness, and connectivity. The date and time are also shown at the bottom right.