

The screenshot shows the IntelliJ IDEA interface with the following details:

- Top Bar:** IntelliJ IDEA, File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, Git, Window, Help.
- Title Bar:** EmployeeManagementRestAPI - EmployeeManagementRestApiApplication.java
- Project Tree:** EmployeeManagementRestAPI / src / main / java / com / greatlearning / mainApp / EmployeeManagementRestApiApplication / main. The tree includes config, controller, dao, entity, errorHandler, mainApp, service, and various DAO, Entity, and Service classes.
- Code Editor:** The main editor window displays the `EmployeeManagementRestApiApplication.java` file. It contains Java code for a Spring Boot application, including annotations like `@EntityScan`, `@EnableJpaRepositories`, and `@SpringBootApplication`. A specific section of code is highlighted in blue, which is the part that populates the database with an initial user entry:

```
import ...  
@EntityScan(basePackages = {"com.greatlearning.entity"})  
@EnableJpaRepositories(basePackages = "com.greatlearning.dao")  
@SpringBootApplication(scanBasePackages = {"com.greatlearning.*"})  
public class EmployeeManagementRestApiApplication {  
  
    public static void main(String[] args) {  
  
        ConfigurableApplicationContext applicationContext = SpringApplication.run(EmployeeManagementRestApiApplication.class, args);  
  
        //populate initial user entry in database having role as ADMIN  
        UserService userServiceJpaRepository = applicationContext.getBean(UserServiceJpaRepositoryImpl.class);  
        User user = new User(username: "Akhil", password: "123");  
        List<Role> roleList = new LinkedList<>();  
        roleList.add(new Role(name: "ADMIN"));  
        user.setRoles(roleList);  
        userServiceJpaRepository.addUser(user);  
    }  
}
```

- Run Tab:** EmployeeManagementRestApiApplication
- Console Tab:** Shows log output from the application startup. The log includes messages from Hibernate, Spring, and Tomcat, indicating the application has started successfully on port 8080.
- Bottom Bar:** Git, Find, Run, Debug, TODO, Problems, Profiler, Terminal, Endpoints, Build, Dependencies, Spring.

localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

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

jdbc:h2:mem:~/employeeDB

EMPLOYEES | ROLE | USER | USERS_ROLES | INFORMATION_SCHEMA | Sequences | Users

H2 1.4.200 (2019-10-14)

Run | Run Selected | Auto complete | Clear | SQL statement:

SELECT * FROM USER

SELECT * FROM USER;

ID	PASSWORD	USERNAME
1	\$2a\$10\$XwJgaQRTXiTvcAqa0VY0GuWEoRO1W328FSZw53T2tx5wbYSYNjtm	Akhil

(1 row, 9 ms)

Edit

localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

Auto commit | Max rows: 1000 | Auto complete Off | Auto select On | SQL statement:

jdbc:h2:mem:~/employeeDB

EMPLOYEES | ROLE | USER | USERS_ROLES | INFORMATION_SCHEMA | Sequences | Users

H2 1.4.200 (2019-10-14)

SELECT * FROM ROLE |

SELECT * FROM ROLE;

ID	NAME
1	ADMIN

(1 row, 1 ms)

Edit

A screenshot of the H2 Database Browser interface. The browser has a top navigation bar with various icons and a URL bar showing 'localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203'. Below the navigation is a toolbar with buttons for Auto commit (checked), Max rows (set to 1000), Auto complete (Off), and Auto select (On). A status bar at the bottom shows the connection details: 'jdbc:h2:mem:~/employeeDB' and the version 'H2 1.4.200 (2019-10-14)'. The left sidebar lists database objects: EMPLOYEES, ROLE, USER, USERS_ROLES, INFORMATION_SCHEMA, Sequences, and Users. The main area displays the results of a SQL query: 'SELECT * FROM ROLE;'. The result set is shown in a table with one row: ID 1 and NAME ADMIN. The table has a header row with columns 'ID' and 'NAME'. An 'Edit' button is located below the table.

localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

Auto commit | Max rows: 1000 | Auto complete Off | Auto select On | SQL statement:

jdbc:h2:mem:~/employeeDB

EMPLOYEES | ROLE | USER | USERS_ROLES | INFORMATION_SCHEMA | Sequences | Users

H2 1.4.200 (2019-10-14)

Run | Run Selected | Auto complete | Clear

SELECT * FROM USERS_ROLES;

USER_ID	ROLE_ID
1	1

(1 row, 2 ms)

localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

Auto commit | Max rows: 1000 | Auto complete Off | Auto select On | SQL statement:

jdbc:h2:mem:~/employeeDB

EMPLOYEES

ROLE

USER

USERS_ROLES

INFORMATION_SCHEMA

Sequences

Users

H2 1.4.200 (2019-10-14)

Run Selected | Auto complete | Clear

SELECT * FROM EMPLOYEES

SELECT * FROM EMPLOYEES;

ID	EMAIL	FIRST_NAME	LAST_NAME
(no rows, 1 ms)			

Edit

Postman

File Edit View Window Help

Home Workspaces Reports Explore

Search Postman

New Import

My Workspace

Collections APIs Environments Mock Servers Monitors History

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

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

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

Body (JSON)

```
1 ...
2 ... "name": "USER"
3 ...
4 ...
5 ...
```

Send Cookies Beautify

Body Cookies (1) Headers (10) Test Results

Pretty Raw Preview Visualize Text

Status: 200 OK Time: 422 ms Size: 305 B Save Response

200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action.

Find and Replace Console

Bootcamp Runner Trash



localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

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

jdbc:h2:mem:~/employeeDB

EMPLOYEES | ROLE | USER | USERS_ROLES | INFORMATION_SCHEMA | Sequences | Users

H2 1.4.200 (2019-10-14)

Run | Run Selected | Auto complete | Clear | SQL statement: SELECT * FROM ROLE

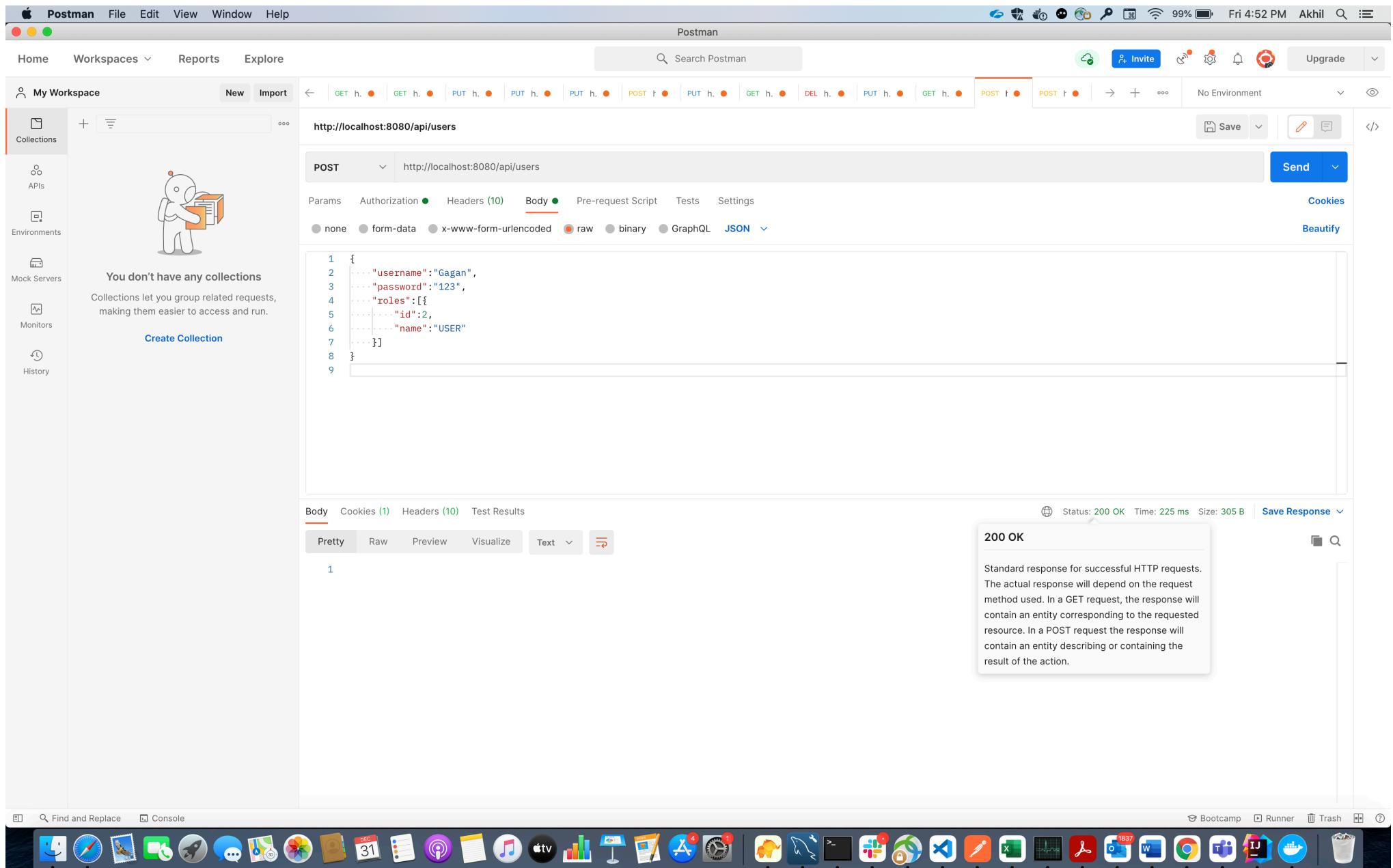
SELECT * FROM ROLE;

ID	NAME
1	ADMIN
2	USER

(2 rows, 1 ms)

Edit

The screenshot shows the H2 Database Browser interface. The left sidebar lists database objects: EMPLOYEES, ROLE, USER, USERS_ROLES, INFORMATION_SCHEMA, Sequences, and Users. The main area displays the result of the SQL query 'SELECT * FROM ROLE;'. The result is presented in a table with two columns: ID and NAME. There are two rows: one with ID 1 and NAME 'ADMIN', and another with ID 2 and NAME 'USER'. The status bar at the bottom indicates '(2 rows, 1 ms)'.



localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

Auto commit | Max rows: 1000 | Auto complete: Off | Auto select: On | SQL statement:

jdbc:h2:mem:~/employeeDB

EMPLOYEES | ROLE | USER | USERS_ROLES | INFORMATION_SCHEMA | Sequences | Users

H2 1.4.200 (2019-10-14)

SELECT * FROM USER

SELECT * FROM USER;

ID	PASSWORD	USERNAME
1	\$2a\$10\$XwJgaQRTXiTvcAqa0VY0GuWEoRO1W328FSZw53T2tcx5wbYSYNjtm	Akhil
2	\$2a\$10\$o8v54/74rZ9ptA2r2zsRe/vyznJ2cljPtxayaG3fAfGHAltikGO	Gagan

(2 rows, 1 ms)

Edit

Postman

File Edit View Window Help

Home Workspaces Reports Explore

Search Postman

New Import

My Workspace

Collections APIs Environments Mock Servers Monitors History

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

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

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

Body (JSON)

```
1
2   ...
3     "firstName": "WOW",
4     "lastName": "Andrews",
5     "email": "leslie@luv2code.com"
```

Send Cookies Beautify

Body Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   {
3     "id": 1,
4     "firstName": "WOW",
5     "lastName": "Andrews",
6     "email": "leslie@luv2code.com"
```

Status: 200 OK Time: 137 ms Size: 423 B Save Response

Find and Replace Console

Bootcamp Runner Trash

98% Fri 4:54 PM Akhil

The screenshot shows the Postman application window. On the left, there's a sidebar with icons for Collections, APIs, Environments, Mock Servers, Monitors, and History. Below the sidebar, it says 'You don't have any collections' and 'Collections let you group related requests, making them easier to access and run.' There's also a 'Create Collection' button. The main area shows a POST request to 'http://localhost:8080/api/employees/'. The 'Body' tab is selected, showing a JSON payload:

```
1
2   ...
3     "firstName": "WOW",
4     "lastName": "Andrews",
5     "email": "leslie@luv2code.com"
```

Below the request, the response is shown with a status of 200 OK, time 137 ms, and size 423 B. The response body is identical to the request body. At the bottom, there's a toolbar with icons for Find and Replace, Console, Bootcamp, Runner, and Trash. The system tray at the top right shows battery level (98%), network signal, and the date/time (Fri 4:54 PM). The dock at the bottom has icons for various Mac applications like Finder, Safari, Mail, and others.

Postman

File Edit View Window Help

Home Workspaces Reports Explore

Search Postman

New Import

My Workspace

Collections APIs Environments Mock Servers Monitors History

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

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

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

Type Basic Auth

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

Learn more about authorization ↗

Username Akhil

Password 123

Show Password

Body Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```
1 "id": 1,
2   "firstName": "WOW",
3   "lastName": "Andrews",
4   "email": "leslie@luv2code.com"
```

Status: 200 OK Time: 137 ms Size: 423 B Save Response

Find and Replace Console

Bootcamp

This screenshot shows the Postman application interface on a Mac OS X desktop. The main window displays a collection named 'My Workspace' containing a single POST request to the endpoint 'http://localhost:8080/api/employees/'. The request is configured to use 'Basic Auth' with the username 'Akhil' and password '123'. The response body is shown in a JSONpretty-printed format:

```
1 "id": 1,
2   "firstName": "WOW",
3   "lastName": "Andrews",
4   "email": "leslie@luv2code.com"
```

The Postman interface includes a sidebar with links for Collections, APIs, Environments, Mock Servers, Monitors, and History. The bottom of the screen shows the Mac OS X Dock with various application icons.

localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

Auto commit | Max rows: 1000 | Auto complete: Off | Auto select: On | SQL statement:

jdbc:h2:mem:~/employeeDB

EMPLOYEES | ROLE | USER | USERS_ROLES | INFORMATION_SCHEMA | Sequences | Users

H2 1.4.200 (2019-10-14)

SELECT * FROM EMPLOYEES |

SELECT * FROM EMPLOYEES;

ID	EMAIL	FIRST_NAME	LAST_NAME
1	leslie@luv2code.com	WOW	Andrews

(1 row, 1 ms)

Edit

localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

Auto commit | Max rows: 1000 | Auto complete: Off | Auto select: On | SQL statement:

jdbc:h2:mem:~/employeeDB

EMPLOYEES | ROLE | USER | USERS_ROLES | INFORMATION_SCHEMA | Sequences | Users

H2 1.4.200 (2019-10-14)

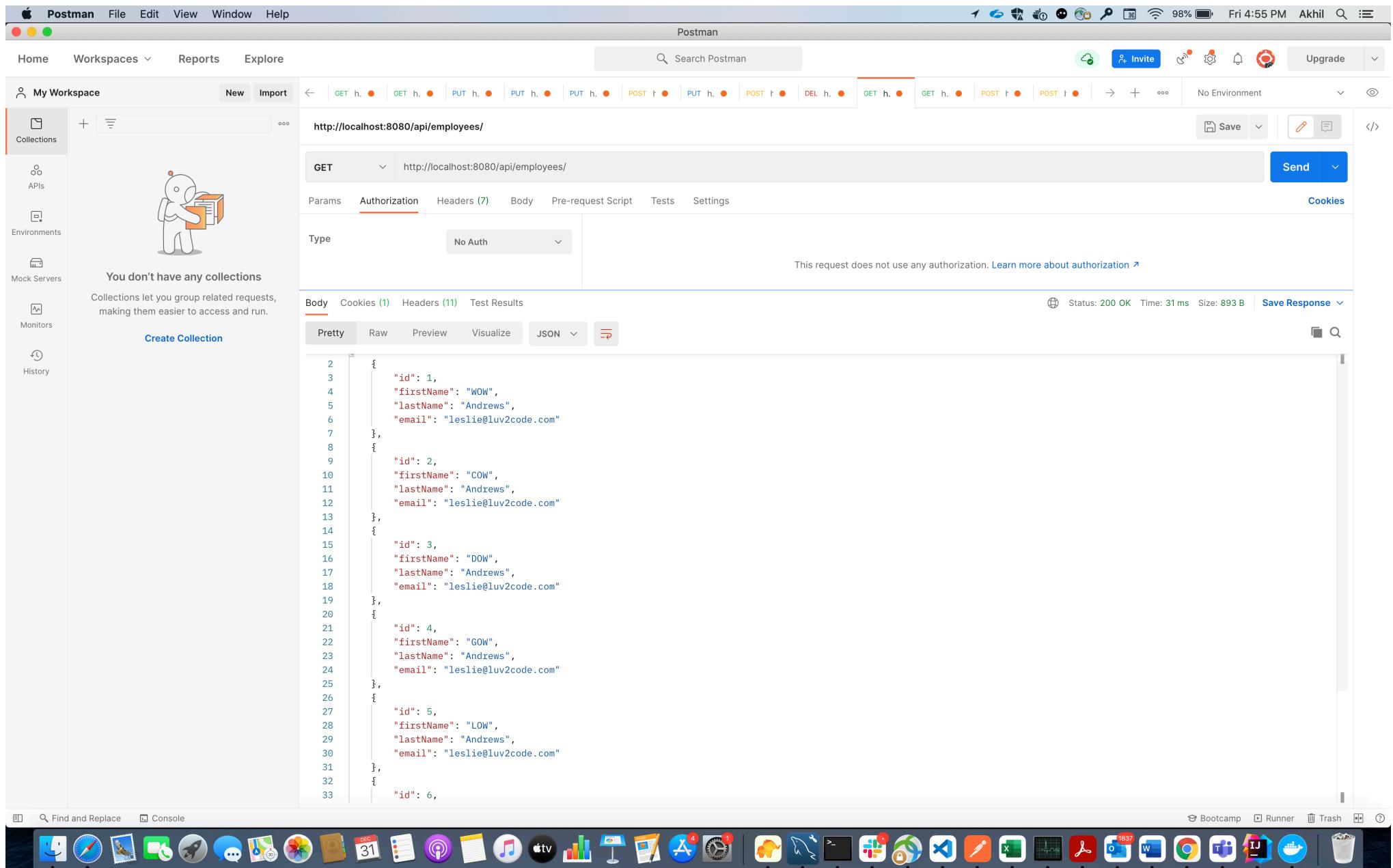
SELECT * FROM EMPLOYEES

SELECT * FROM EMPLOYEES;

ID	EMAIL	FIRST_NAME	LAST_NAME
1	leslie@luv2code.com	WOW	Andrews
2	leslie@luv2code.com	COW	Andrews
3	leslie@luv2code.com	DOW	Andrews
4	leslie@luv2code.com	GOW	Andrews
5	leslie@luv2code.com	LOW	Andrews
6	leslie@luv2code.com	LOW	Andrews
7	leslie@luv2code.com	BOW	Andrews

(7 rows, 1 ms)

Edit



Postman

File Edit View Window Help

My Workspace

New Import

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

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

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

Type No Auth

This request does not use any authorization. [Learn more about authorization](#)

Body Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize JSON

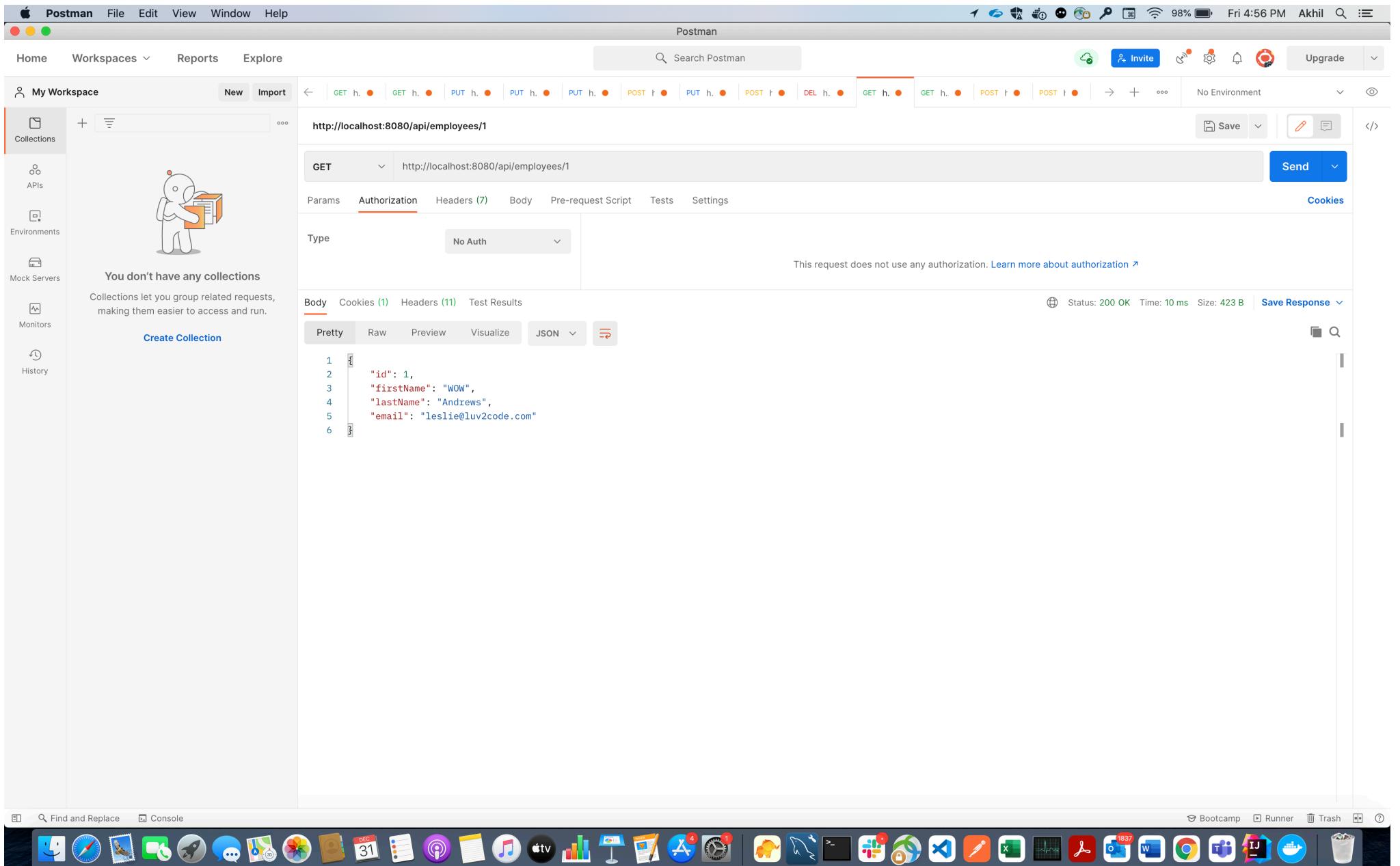
1 2 3 4 5 6

```
1    "id": 1,
2    "firstName": "WOW",
3    "lastName": "Andrews",
4    "email": "leslie@luv2code.com"
```

Status: 200 OK Time: 10 ms Size: 423 B Save Response

Find and Replace Console

Bootcamp Runner Trash



Postman

File Edit View Window Help

Home Workspaces Reports Explore

Search Postman

New Import

My Workspace

Collections APIs Environments Mock Servers Monitors History

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

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

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

Body (JSON)

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

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

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

Body (JSON)

1 "id": 1,
2 "firstName": "WOW updated",
3 "lastName": "Andrews",
4 "email": "leslie@luv2code.com"

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 117 ms Size: 431 B Save Response

Pretty Raw Preview Visualize JSON

1 "id": 1,
2 "firstName": "WOW updated",
3 "lastName": "Andrews",
4 "email": "leslie@luv2code.com"

Find and Replace Console

Bootcamp Runner Trash

98% Fri 4:57 PM Akhil

1837

localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

Auto commit | Max rows: 1000 | Auto complete: Off | Auto select: On | SQL statement:

jdbc:h2:mem:~/employeeDB

EMPLOYEES | ROLE | USER | USERS_ROLES | INFORMATION_SCHEMA | Sequences | Users

H2 1.4.200 (2019-10-14)

SELECT * FROM EMPLOYEES

SELECT * FROM EMPLOYEES;

ID	EMAIL	FIRST_NAME	LAST_NAME
1	leslie@luv2code.com	WOW updated	Andrews
2	leslie@luv2code.com	COW	Andrews
3	leslie@luv2code.com	DOW	Andrews
4	leslie@luv2code.com	GOW	Andrews
5	leslie@luv2code.com	LOW	Andrews
6	leslie@luv2code.com	LOW	Andrews
7	leslie@luv2code.com	BOW	Andrews

(7 rows, 1 ms)

Edit

Postman

File Edit View Window Help

Home Workspaces Reports Explore

Search Postman

New Import

My Workspace

Collections APIs Environments Mock Servers Monitors History

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

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

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

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

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 115 ms Size: 376 B Save Response

Pretty Raw Preview Visualize Text

1 Deleted employee having id : 2

Find and Replace Console

Bootcamp Runner Trash

97% Fri 4:57 PM Akhil

The screenshot shows the Postman application running on a Mac OS X desktop. The main window displays a DELETE request to the endpoint `http://localhost:8080/api/employees/2`. The 'Body' tab is selected, showing the response message: `1 Deleted employee having id : 2`. The 'Headers' tab shows 8 items. The status bar at the bottom indicates the system is at 97% battery, the time is 4:57 PM, and the user is Akhil. The Dock at the bottom is visible, showing various Mac OS X application icons.

localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

Auto commit | Max rows: 1000 | Auto complete: Off | Auto select: On | SQL statement:

jdbc:h2:mem:~/employeeDB

EMPLOYEES | ROLE | USER | USERS_ROLES | INFORMATION_SCHEMA | Sequences | Users

H2 1.4.200 (2019-10-14)

SELECT * FROM EMPLOYEES |

SELECT * FROM EMPLOYEES;

ID	EMAIL	FIRST_NAME	LAST_NAME
1	leslie@luv2code.com	WOW updated	Andrews
3	leslie@luv2code.com	DOW	Andrews
4	leslie@luv2code.com	GOW	Andrews
5	leslie@luv2code.com	LOW	Andrews
6	leslie@luv2code.com	LOW	Andrews
7	leslie@luv2code.com	BOW	Andrews

(6 rows, 1 ms)

Edit

Postman

File Edit View Window Help

Home Workspaces Reports Explore

Search Postman

New Import

My Workspace

Collections APIs Environments Mock Servers Monitors History

+

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

Send Cookies

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

Body Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize JSON

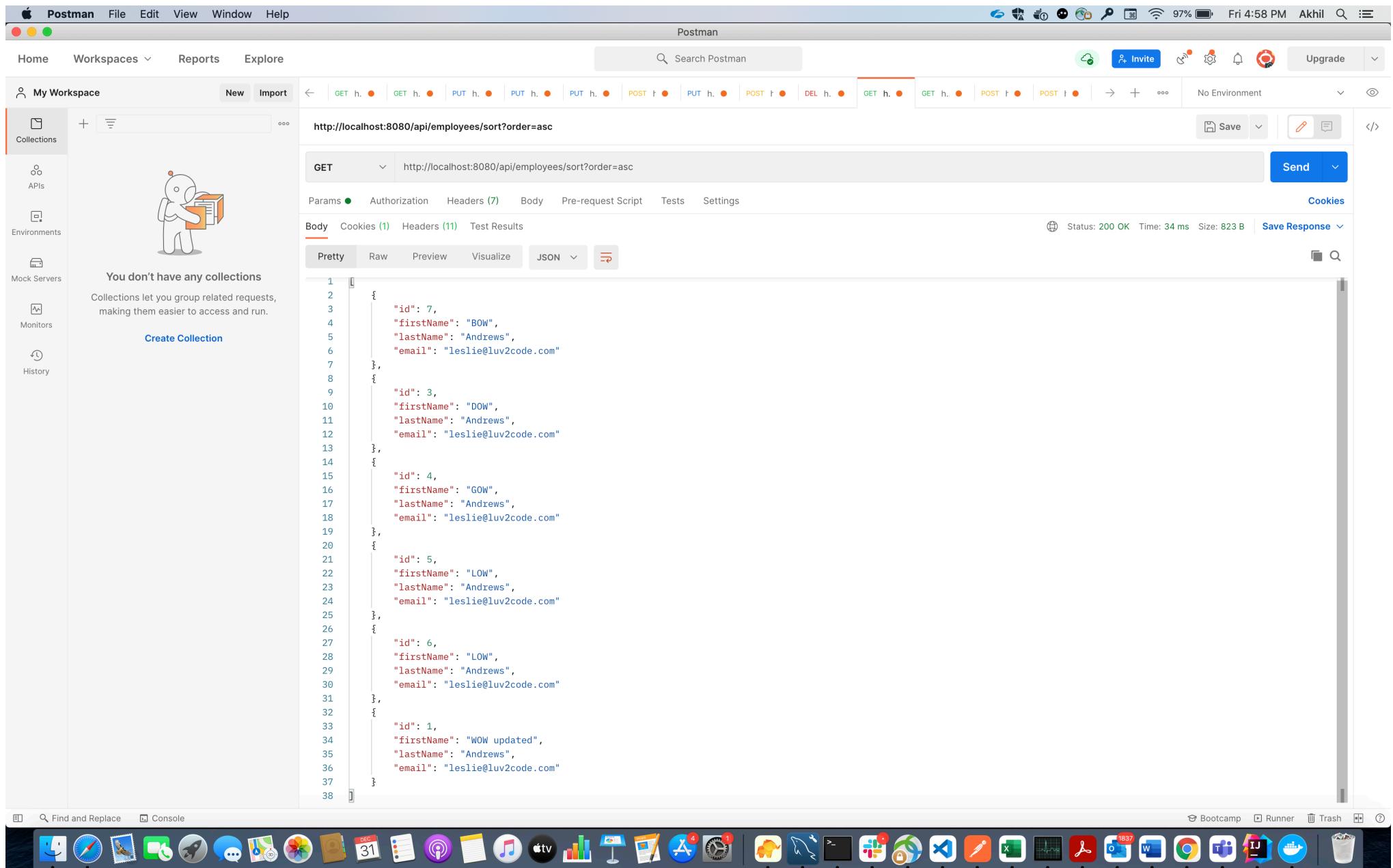
1 {
2 "id": 7,
3 "firstName": "BOW",
4 "lastName": "Andrews",
5 "email": "leslie@luv2code.com"
6 },
7 {
8 "id": 3,
9 "firstName": "DOW",
10 "lastName": "Andrews",
11 "email": "leslie@luv2code.com"
12 },
13 {
14 "id": 4,
15 "firstName": "GOW",
16 "lastName": "Andrews",
17 "email": "leslie@luv2code.com"
18 },
19 {
20 "id": 5,
21 "firstName": "LOW",
22 "lastName": "Andrews",
23 "email": "leslie@luv2code.com"
24 },
25 {
26 "id": 6,
27 "firstName": "LOW",
28 "lastName": "Andrews",
29 "email": "leslie@luv2code.com"
30 },
31 {
32 }

Find and Replace Console

Bootcamp Runner Trash

97% Fri 4:58 PM Akhil

1837



Postman

File Edit View Window Help

Home Workspaces Reports Explore

Search Postman

New Import

My Workspace

Collections APIs Environments Mock Servers Monitors History

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 Cookies

Status: 200 OK Time: 8 ms Size: 823 B Save Response

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

```
1 {
2   "id": 1,
3   "firstName": "WOW updated",
4   "lastName": "Andrews",
5   "email": "leslie@luv2code.com"
6 },
7 {
8   "id": 5,
9   "firstName": "LOW",
10  "lastName": "Andrews",
11  "email": "leslie@luv2code.com"
12 },
13 {
14   "id": 6,
15   "firstName": "LOW",
16   "lastName": "Andrews",
17   "email": "leslie@luv2code.com"
18 },
19 {
20   "id": 4,
21   "firstName": "GOW",
22   "lastName": "Andrews",
23   "email": "leslie@luv2code.com"
24 },
25 {
26   "id": 3,
27   "firstName": "DOW",
28   "lastName": "Andrews",
29   "email": "leslie@luv2code.com"
30 },
31 {
32   "id": 7,
33   "firstName": "BOW",
34   "lastName": "Andrews",
35   "email": "leslie@luv2code.com"
36 }
37 }
```

Find and Replace Console

Bootcamp Runner Trash



localhost:8080/h2/login.do?sessionId=0d09852f0835a0187eda24cbbd64c203

Auto commit | Max rows: 1000 | Run Selected | Auto complete | Clear SQL statement:

SELECT * FROM EMPLOYEES

EMPLOYEES ROLE USER USERS_ROLES INFORMATION_SCHEMA Sequences Users

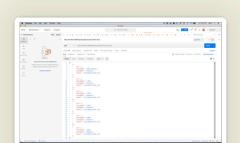
H2 1.4.200 (2019-10-14)

SELECT * FROM EMPLOYEES;

ID	EMAIL	FIRST_NAME	LAST_NAME
1	leslie@luv2code.com	WOW updated	Andrews
3	leslie@luv2code.com	DOW	Andrews
4	leslie@luv2code.com	GOW	Andrews
5	leslie@luv2code.com	LOW	Andrews
6	leslie@luv2code.com	LOW	Andrews
7	leslie@luv2code.com	BOW	Andrews

(6 rows, 1 ms)

Edit



Postman

File Edit View Window Help

Home Workspaces Reports Explore

Search Postman

New Import

My Workspace

Collections APIs Environments Mock Servers Monitors History

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

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

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

Type No Auth

This request does not use any authorization. [Learn more about authorization](#)

Body Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 11 ms Size: 503 B Save Response

Find and Replace Console

Bootcamp Runner Trash

Find and Replace Console

1 [{ "id": 5, "firstName": "LOW", "lastName": "Andrews", "email": "leslie@luv2code.com" }, { "id": 6, "firstName": "LOW", "lastName": "Andrews", "email": "leslie@luv2code.com" }]

The screenshot shows the Postman application window. In the center, there's a request card for a GET request to `http://localhost:8080/api/employees/search/LOW`. The 'Authorization' tab is selected. Below the card, the response body is displayed in a pretty-printed JSON format:

```
1 [ { "id": 5, "firstName": "LOW", "lastName": "Andrews", "email": "leslie@luv2code.com" }, { "id": 6, "firstName": "LOW", "lastName": "Andrews", "email": "leslie@luv2code.com" } ]
```

The status bar at the bottom indicates a 200 OK status with a time of 11 ms and a size of 503 B. The Postman interface includes a sidebar with sections for Collections, APIs, Environments, Mock Servers, Monitors, and History. The bottom of the screen shows the Mac OS X Dock with various application icons.