

localhost:8080/EmployeeManagement/h2-console/login.do?jsessionId=5428b18678728f940e050ddc1850928b

Gmail YouTube Maps font-family - CSS: C... Get Started At Cod...

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

jdbc:h2:mem:employeeDB

- AUTHORITIES
- EMPLOYEE
- USERS
- INFORMATION_SCHEMA
- Users
- H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM AUTHORITIES

SELECT * FROM AUTHORITIES;

| USERNAME | AUTHORITY |
|----------|------------------|
| shubham | ROLE_Normal_User |
| srikanth | ROLE_Admin |

(2 rows, 1 ms)

Edit

1. Added roles in the database

```
New Eclipse WorkSpace - EmployeeManagement-Assignment4/src/main/java/com/greatlearning/employeemanagement/config/SecurityConfigurationWithJDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

SecurityConfigurationWithJDBC.java X
12 import org.springframework.security.core.userdetails.User;
13 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
14 import org.springframework.security.crypto.password.PasswordEncoder;
15
16 @EnableWebSecurity
17 public class SecurityConfigurationWithJDBC extends WebSecurityConfigurerAdapter {
18
19     @Autowired
20     DataSource dataSource;
21
22     @Override
23     protected void configure(AuthenticationManagerBuilder auth) throws Exception {
24
25         auth.jdbcAuthentication().dataSource(dataSource).withDefaultSchema()
26             .withUser(
27                 User.withUsername("srikanth").password(getPasswordEncoder().encode("srikanth")).roles("Admin"))
28             .withUser(User.withUsername("shubham").password(getPasswordEncoder().encode("shubham"))
29                 .roles("Normal_User"));
30     }
31 }
```

Added Users and roles in the database for authentication

Gmail YouTube Maps font-family - CSS: C... Get Started At Cod...

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

jdbc:h2:mem:employeeDB

- AUTHORITIES
- EMPLOYEE
- USERS
- INFORMATION_SCHEMA
- Users
- H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM USERS

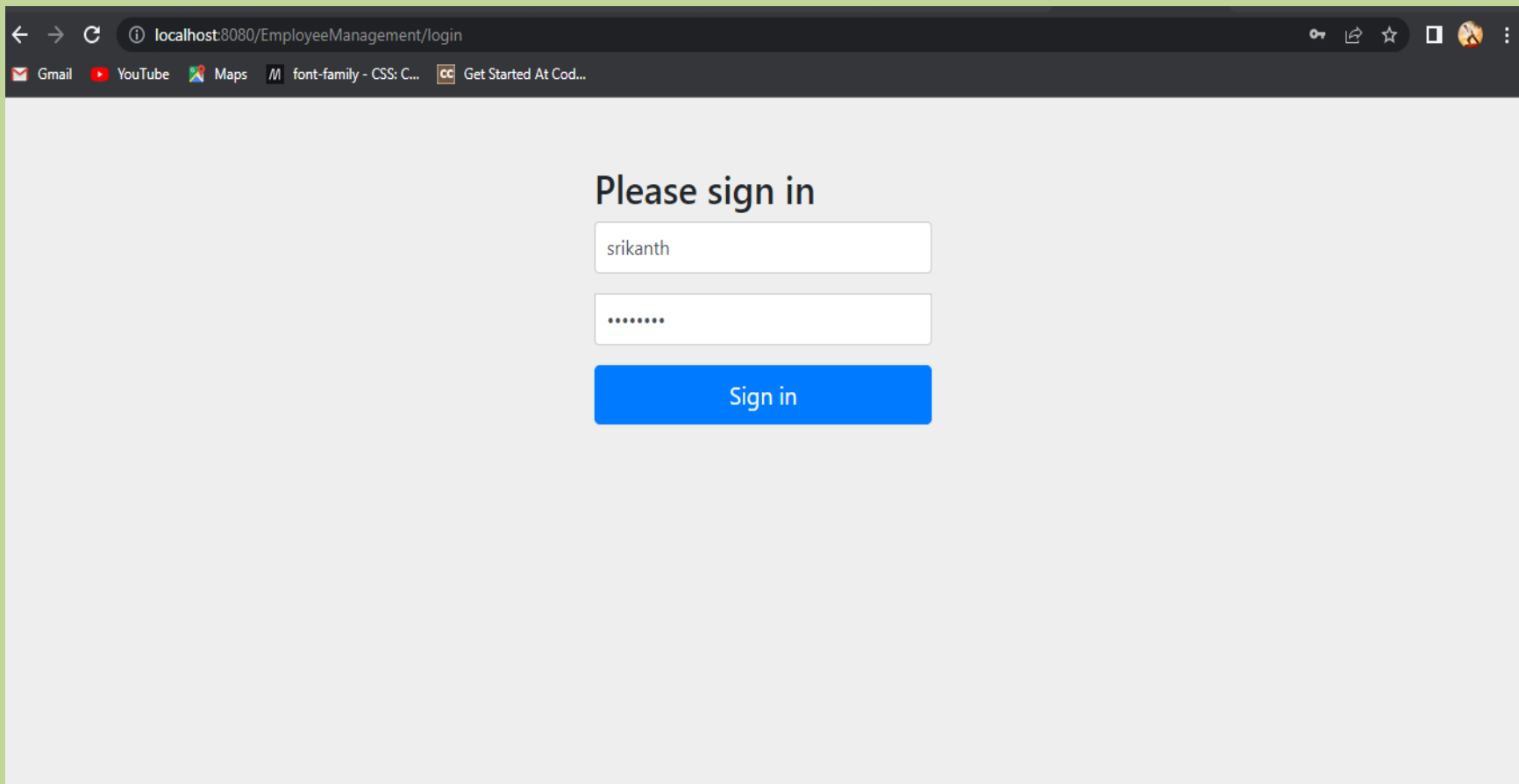
SELECT * FROM USERS;

| USERNAME | PASSWORD | ENABLED |
|----------|---|---------|
| srikanth | \$2a\$10\$8JayvwjzhOmz5wixEa81kOJziA5qzR0sl0U7vqVS0hdYTMI1xMIWC | TRUE |
| shubham | \$2a\$10\$4lwGbz0KAWlnR2ELAUow8.S/gbJ5SQhoc2ys9LpZrlt1tp7r2.fMC | TRUE |

(2 rows, 0 ms)

Edit

2. Added Users in the database



3. Logging in as Admin and adding new employee

localhost8080/EmployeeManagement/swagger-ui.html#/employee-create-controller/addEmployeeUsingPOST

Gmail

YouTube

Maps

M font-family - CSS: C...

CC Get Started At Cod...

POST

/createService/addEmployee addEmployee

Parameters

Cancel

| Name | Description |
|-----------------------------------|---|
| email string (query) | <input type="text" value="arjith@gmail.com"/> |
| firstName string (query) | <input type="text" value="Arjith"/> |
| id integer(\$int32) (query) | <input type="text" value="5"/> |
| lastName string (query) | <input type="text" value="Mahendra"/> |

Execute

localhost:8080/EmployeeManagement/swagger-ui.html#/employee-create-controller/addEmployeeUsingPOST

Gmail

YouTube

Maps

font-family - CSS: C...

Get Started At Cod...

Curl

```
curl -X POST "http://localhost:8080/EmployeeManagement/createService/addEmployee?email=arjith%40gmail.com&firstName=Arjith&id=5&lastName=Mahendra" -H "accept: */*"
```

Request URL

```
http://localhost:8080/EmployeeManagement/createService/addEmployee?email=arjith%40gmail.com&firstName=Arjith&id=5&lastName=Mahendra
```

Server response

| Code | Details |
|------|---|
| 200 | <div><div>Response body</div><div><pre>New Employee added successfully</pre><div>Download</div></div></div> <div><div>Response headers</div><div><pre>cache-control: no-cache, no-store, max-age=0, must-revalidate connection: keep-alive content-length: 31 content-type: text/plain;charset=UTF-8 date: Tue, 19 Jul 2022 10:07:48 GMT expires: 0 keep-alive: timeout=60 pragma: no-cache x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 1; mode=block</pre></div></div> |

Responses

localhost:8080/EmployeeManagement/swagger-ui.html#/employee-read-controller/findAllEmployeesUsingGET

Request URL

http://localhost:8080/EmployeeManagement/readService/findAllEmployees

Server response

| Code | Details |
|------|--|
| 200 | <p>Response body</p> <pre>[{ "id": 1, "firstName": "Sai", "lastName": "Srikanth", "email": "ddssrikanth@gmail.com" }, { "id": 2, "firstName": "Shubham", "lastName": "Sharma", "email": "shubham@gmail.com" }, { "id": 3, "firstName": "Hari", "lastName": "Nadh", "email": "hari@gmail.com" }, { "id": 4, "firstName": "Hari", "lastName": "Krishna", "email": "krishna@gmail.com" }, { </pre> <p>Download</p> <p>Response headers</p> |

4. Listing all the employees

localhost:8080/EmployeeManagement/swagger-ui.html#/employee-read-controller/findEmployeeByIdUsingGET

Request URL

http://localhost:8080/EmployeeManagement/readService/findEmployeeById?id=1

Server response

| Code | Details |
|------|--|
| 200 | <p>Response body</p> <pre>{ "id": 1, "firstName": "Sai", "lastName": "Srikanth", "email": "ddssrikanth@gmail.com" }</pre> <p>Download</p> <p>Response headers</p> <pre>cache-control: no-cache, no-store, max-age=0, must-revalidate connection: keep-alive content-type: application/json date: Tue, 19 Jul 2022 10:12:25 GMT expires: 0 keep-alive: timeout=60 pragma: no-cache transfer-encoding: chunked x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 1; mode=block</pre> |

Responses

| Code | Description |
|------|-------------|
| 200 | OK |

5. Fetching an employee based on Id

localhost:8080/EmployeeManagement/swagger-ui.html#/employee-update-controller/updateExistingEmployeeUsingPOST

POST /updateService/updateExistingEmployee updateExistingEmployee

Parameters Cancel

| Name | Description |
|-----------------------------------|--|
| email string (query) | <input type="text" value="varma@gmail.com"/> |
| firstName string (query) | <input type="text" value="Ramanuja"/> |
| id integer(\$int32) (query) | <input type="text" value="2"/> |
| lastName string (query) | <input type="text" value="Varma"/> |

Execute

6. Updating an employee

localhost8080/EmployeeManagement/swagger-ui.html#/employee-update-controller/updateExistingEmployeeUsingPOST

Gmail YouTube Maps font-family - CSS: C... Get Started At Cod...

Curl

```
curl -X POST "http://localhost:8080/EmployeeManagement/updateService/updateExistingEmployee?email=varma%40gmail.com&firstName=Ramanuja&id=2&lastName=Varma" -H "accept: */*"
```

Request URL

```
http://localhost:8080/EmployeeManagement/updateService/updateExistingEmployee?email=varma%40gmail.com&firstName=Ramanuja&id=2&lastName=Varma
```

Server response

| Code | Details |
|------|--|
| 200 | <div><div>Response body</div><div><pre>{ "id": 2, "firstName": "Ramanuja", "lastName": "Varma", "email": "varma@gmail.com"}</pre><div>Download</div></div></div> <div><div>Response headers</div><div><pre>cache-control: no-cache, no-store, max-age=0, must-revalidate connection: keep-alive content-type: application/json date: Tue, 19 Jul 2022 10:15:53 GMT expires: 0 keep-alive: timeout=60 pragma: no-cache transfer-encoding: chunked x-content-type-options: nosniff x-frame-options: DENY</pre></div></div> |

localhost:8080/EmployeeManagement/swagger-ui.html#/employee-delete-controller/deleteByIdUsingDELETE

DELETE /deleteService/deleteById deleteById

Parameters Cancel

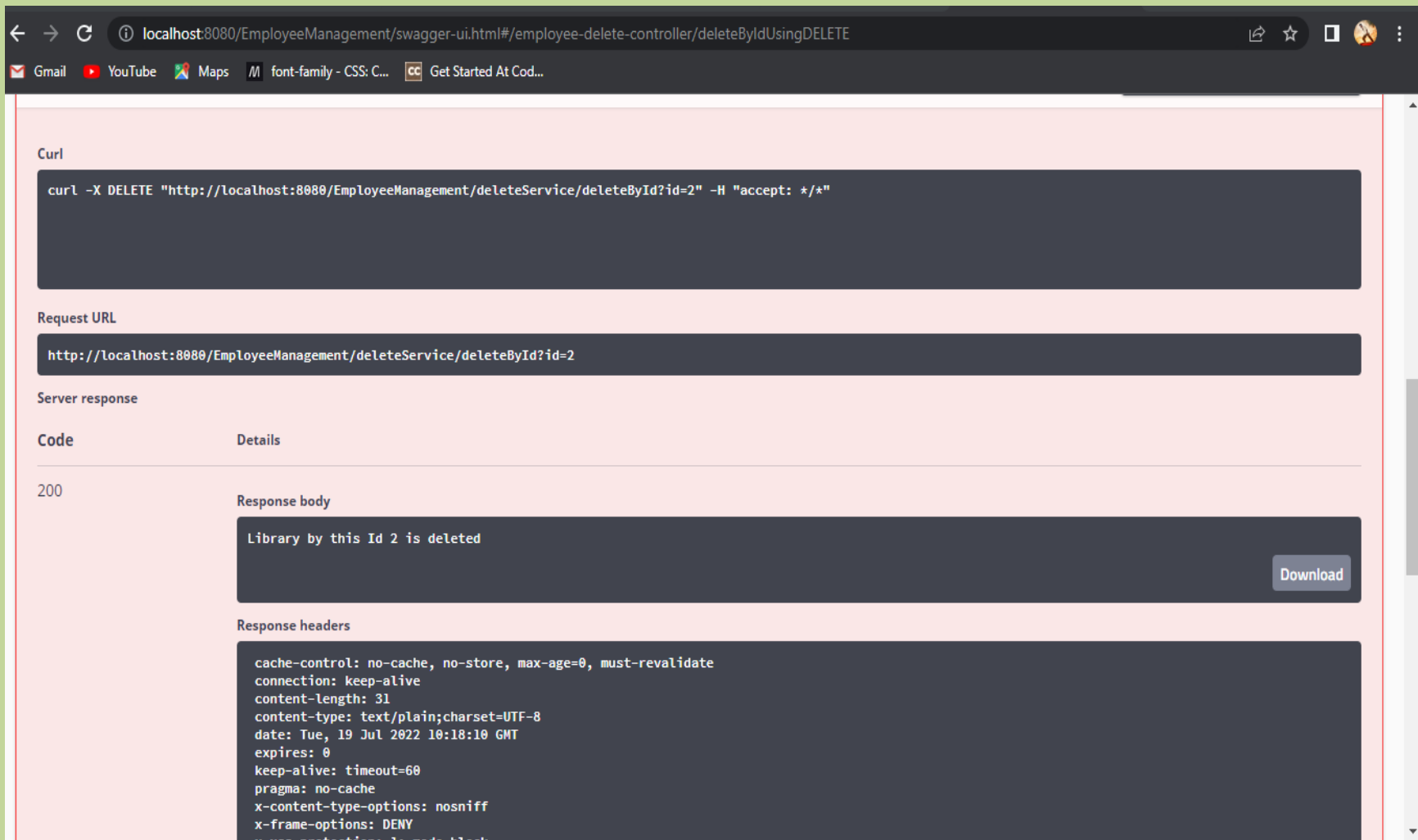
| Name | Description |
|-----------------------------------|--------------------------------------|
| id integer(\$int32) (query) | id <input type="text" value="2"/> |

Execute

Responses Response content type */*

| Code | Description |
|------|--|
| 200 | <p>OK</p> <p>Example Value Model</p> <p>"string"</p> |

7. Deleting an employee based on Id



localhost:8080/EmployeeManagement/swagger-ui.html#/employee-read-controller/findEmployeeByFirstNameUsingGET

GET /readService/findEmployeeByFirstName findEmployeeByFirstName

Parameters Cancel

| Name | Description |
|--------------------------------|--|
| firstName string (query) | firstName <input type="text" value="Hari"/> |

Execute

Responses Response content type */*

| Code | Description |
|------|--|
| 200 | <p>OK</p> <p>Example Value Model</p> <pre>[{ "email": "etring" }]</pre> |

8. Fetching employee by First Name

localhost8080/EmployeeManagement/swagger-ui.html#/employee-read-controller/findEmployeeByFirstNameUsingGET

Gmail

YouTube

Maps

M font-family - CSS: C...

CC Get Started At Cod...

Curl

curl -X GET "http://localhost:8080/EmployeeManagement/readService/findEmployeeByFirstName?firstName=Hari" -H "accept: */*"

Request URL

http://localhost:8080/EmployeeManagement/readService/findEmployeeByFirstName?firstName=Hari

Server response

| Code | Details |
|------|---|
| 200 | <div><div>Response body</div><div><pre>[{ "id": 3, "firstName": "Hari", "lastName": "Nadh", "email": "hari@gmail.com" }, { "id": 4, "firstName": "Hari", "lastName": "Krishna", "email": "krishna@gmail.com" }]</pre></div><div>Download</div></div> <div><div>Response headers</div><div>cache-control: no-cache, no-store, max-age=0, must-revalidate</div></div> |

localhost:8080/EmployeeManagement/swagger-ui.html#/employee-read-controller/employeesSortedByFirstNameUsingGET

GET /readService/employeesSortedByFirstName employeesSortedByFirstName

Parameters Cancel

| Name | Description |
|--------------------------------|------------------|
| direction string (query) | direction ASC |

Execute

Responses Response content type */*

| Code | Description |
|------|-------------|
| 200 | OK |

Example Value | Model

```
[  
  {  
    "email": "string",  
  },  
]
```

9. Fetching by First Name in Ascending order

localhost:8080/EmployeeManagement/swagger-ui.html#/employee-read-controller/employeesSortedByFirstNameUsingGET

Request URL

http://localhost:8080/EmployeeManagement/readService/employeesSortedByFirstName?direction=ASC

Server response

| Code | Details |
|------|--|
| 200 | <p>Response body</p> <pre>[{ "id": 5, "firstName": "Arjith", "lastName": "Mahendra", "email": "arjith@gmail.com" }, { "id": 3, "firstName": "Hari", "lastName": "Nadh", "email": "hari@gmail.com" }, { "id": 4, "firstName": "Hari", "lastName": "Krishna", "email": "krishna@gmail.com" }, { "id": 1, "firstName": "Sai", "lastName": "Srikanth", "email": "ddssrikanth@gmail.com" }]</pre> <p>Download</p> <p>Response headers</p> |

Ascending Order

localhost:8080/EmployeeManagement/swagger-ui.html#/employee-read-controller/employeesSortedByFirstNameUsingGET

Request URL

http://localhost:8080/EmployeeManagement/readService/employeesSortedByFirstName?direction=DESC

Server response

| Code | Details |
|------|--|
| 200 | <p>Response body</p> <pre>[{ "id": 1, "firstName": "Sai", "lastName": "Srikanth", "email": "ddssrikanth@gmail.com" }, { "id": 3, "firstName": "Hari", "lastName": "Nadh", "email": "hari@gmail.com" }, { "id": 4, "firstName": "Hari", "lastName": "Krishna", "email": "krishna@gmail.com" }, { "id": 5, "firstName": "Arjith", "lastName": "Mahendra", "email": "arjith@gmail.com" }]</pre> <p>Download</p> <p>Response headers</p> |

Descending Order