

GRADED PROJECT 3 DOCUMENTATION

Question: You are required to create a Employee Management Rest Api based Web application, where you will be developing CRUD(Create, Read, Update and Delete) functionality along with Sorting and some concepts of security. Your Rest Api should be secure. And should have different endpoints for different operations-

1.Your application should be able to add roles in the database dynamically in the db:

Screenshot:

The screenshot shows a REST API documentation interface. At the top, there is a 'Curl' section with the command:

```
curl -X POST "http://localhost:8080/roleManagement/addRole" -H "accept: */*" -H "Content-Type: application/json" -d "{ \"roleId\": 0, \"roleName\": \"USER\"}"
```

Below it is a 'Request URL' field containing:

```
http://localhost:8080/roleManagement/addRole
```

The 'Server response' section is expanded, showing a 200 status code. It includes a 'Response body' with the JSON:{ "roleId": 0, "roleName": "USER"}

A 'Download' button is next to the response body. Below that is a 'Response headers' section with the following content:cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json
date: Mon, 13 Nov 2023 10:57:32 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block

At the bottom of the response section is a 'Responses' table with one row for code 200, labeled 'ok'.

The screenshot shows the 'role-controller' endpoint details. It starts with a 'POST' method and a 'saveRole' operation:

```
POST /roleManagement/addRole saveRole
```

There is a 'Try it out' button to the right.

The 'Parameters' section shows a single parameter 'role' (required, type 'body') with an example value:

```
{ "roleId": 0, "roleName": "USER"}
```

The 'Parameter content type' is set to 'application/json'.

The 'Responses' section contains a 'Curl' command:

```
curl -X POST "http://localhost:8080/roleManagement/addRole" -H "accept: */*" -H "Content-Type: application/json" -d "{ \"roleId\": 0, \"roleName\": \"USER\"}"
```

Below it is a 'Request URL' field containing:

```
http://localhost:8080/roleManagement/addRole
```

The 'Server response' section is collapsed.

2. Your application should be able to add Users in the db which can be used for authentication purposes.

POST /userManagement/addUser saveUser

Parameters

Name	Description
user * required	USER
(body)	Example Value Model

```
{
  "id": 0,
  "password": "$2a$12$thSmbR9HmVbiZP87IGA5G06foHzXoS/OyS.pZ0.ulpovr4x0FEPvG",
  "roles": [
    {
      "roleId": 2,
      "roleName": "USER"
    }
  ],
  "username": "user2"
}
```

Cancel

Parameter content type
application/json

Execute

Curl

```
curl -X POST "http://localhost:8080/userManagement/addUser" -H "accept: */*" -H "Content-Type: application/json" -d "[{"id": 0, "password": "$2a$12$thSmbR9HmVbiZP87IGA5G06foHzXoS/OyS.pZ0.ulpovr4x0FEPvG"}, {"roles": [{"roleId": 2, "roleName": "USER"}], "username": "user2"}]"
```

Request URL
<http://localhost:8080/userManagement/addUser>

Server response

Code	Details
200	Response body <pre>[{"id": 3, "username": "user2", "password": "\$2a\$12\$thSmbR9HmVbiZP87IGA5G06foHzXoS/OyS.pZ0.ulpovr4x0FEPvG", "roles": [{ "roleId": 2, "roleName": "USER" }] }</pre> <p>Download</p> Response headers <pre>cache-control: no-cache, no-store, max-age=0, must-revalidate connection: keep-alive content-type: application/json date: Tue, 10 Nov 2022 17:03:40 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

3. Now Your application should be able to add employees data in the db if and only if the authenticated user is **ADMIN**-

--Logged in as User

POST /employeeManagement/addEmployee saveEmployee

Parameters

Name Description

employee * required employee

(body) Example Value | Model

```
{ "email": "abc", "employeeId": 0, "firstName": "abc", "lastName": "xyz" }
```

Cancel

Parameter content type application/json

Execute Clear

Curl

```
curl -X POST "http://localhost:8080/employeeManagement/addEmployee" -H "accept: */*" -H "Content-Type: application/json" -d "[{"email": "abc", "employeeId": 0, "firstName": "abc", "lastName": "xyz"}]"
```

Request URL

http://localhost:8080/employeeManagement/addEmployee

Server response

Code Details

404 Error:

Response body

```
{ "timestamp": "2022-11-10T17:06:03.764+00:00", "status": 404, "error": "Not Found", "message": "No message available", "path": "/employeeManagement/addEmployee" }
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json; charset=ISO-8859-1
date: Thu, 10 Nov 2022 17:06:03 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

Responses

Code Description

200 OK

--Logged in as Admin

POST /employeeManagement/addEmployee saveEmployee

Parameters

Name	Description
employee * required (body)	employee

Example Value | Model

```
{ "email": "Hello@gmail.com", "employeeId": 0, "firstName": "Swathi", "lastName": "GL" }
```

Cancel

Parameter content type

application/json

Execute

curl -X POST "http://localhost:8080/employeeManagement/addEmployee" -H "accept: */*" -H "Content-Type: application/json" -d "[{ \"email\": \"Hello@gmail.com\", \"employeeId\": 0, \"firstName\": \"Swathi\", \"lastName\": \"GL\" }]"

Request URL

http://localhost:8080/employeeManagement/addEmployee

Server response

Code	Details
200	Response body

```
{ "employeeId": 5, "firstName": "Swathi", "lastName": "GL", "email": "Hello@gmail.com" }
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json
date: Thu, 10 Nov 2022 17:17:57 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
x-content-type-prefix: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

Responses

Code	Description
200	OK

4. Your application should provide an endpoint to list all the employees stored in the database.

GET /employeeManagement/allEmployees getAllEmployees

Parameters

No parameters

Responses

Code Description

200 OK

Example Value | Model

```
[ { "email": "string", "employeeId": 0, "firstName": "string", "lastName": "string" } ]
```

401 Unauthorized

403 Forbidden

404 Not Found

Execute Response content type */*

Cancel

Request URL
<http://localhost:8080/employeeManagement/allEmployees>

Server response

Code Details

200 Response body

```
[ { "employeeId": 2, "firstName": "Gouri S", "lastName": "Pendit", "email": "abc@gmail.com" }, { "employeeId": 3, "firstName": "Yash", "lastName": "Yadav", "email": "abc@xyz.com" }, { "employeeId": 4, "firstName": "Swathi", "lastName": "Great Learning", "email": "abc@xyz.com" } ]
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json
date: Thu, 10 Nov 2022 17:07:48 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
```

Responses

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

GET /employeeManagement/getEmployeeById/{id} getEmployeeById

Cancel

Parameters

Name	Description
Id * required integer(\$int32) (path)	id <input type="text" value="3"/>

Execute

Responses

Code	Description
200	OK
	Example Value Model { "empty": true, "present": true }
401	Unauthorized
403	Forbidden
404	Not Found

Response content type */*

Curl

```
curl -X GET "http://localhost:8080/employeeManagement/getEmployeeById/3" -H "accept: */*"
```

Request URL

<http://localhost:8080/employeeManagement/getEmployeeById/3>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "employeeId": 3, "firstName": "Yash", "lastName": "Yadav", "email": "abc@xyz.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: Thu, 10 Nov 2022 17:09:19 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

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

--Logged in as User

PUT /employeeManagement/editEmployee editEmployee

Parameters

Name	Description
employee * required (body)	employee

Example Value | Model

```
{
  "email": "aaa@bbb.com",
  "employeeId": 3,
  "firstName": "Yash",
  "lastName": "Great Learning"
}
```

Cancel

Parameter content type
application/json

Execute

Curl

```
curl -X PUT "http://localhost:8080/employeeManagement/editEmployee" -H "accept: */*" -H "Content-Type: application/json" -d "[{"email": "aaa@bbb.com", "employeeId": 3, "firstName": "Yash", "lastName": "Great Learning"}"]"
```

Request URL
<http://localhost:8080/employeeManagement/editEmployee>

Server response

Code	Details
404	Error: Response body

```
{
  "timestamp": "2022-11-10T17:10:59.401+00:00",
  "status": 404,
  "error": "Not Found",
  "message": "No message available",
  "path": "/employeeManagement/editEmployee"
}
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json; charset=ISO-8859-1
date: Thu, 10 Nov 2022 17:10:59 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

Responses

--Logged in as Admin

PUT /employeeManagement/editEmployee editEmployee

Parameters

Name	Description
employee * required (body)	employee

Example Value | Model

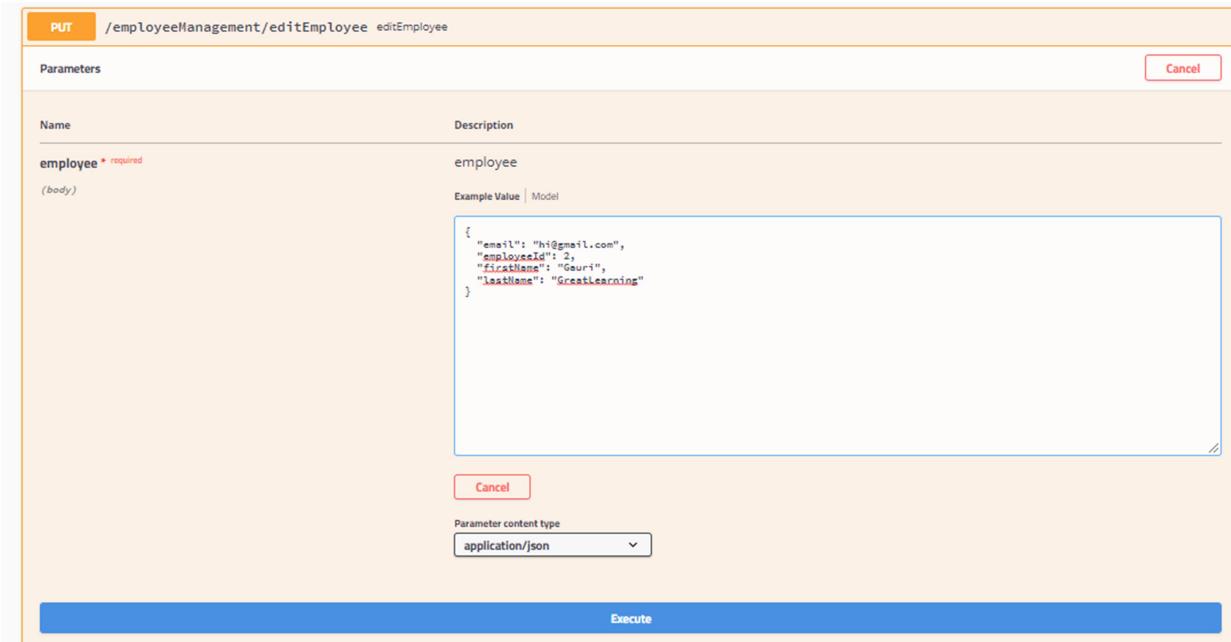
```
{ "email": "hi@gmail.com", "employeeId": 2, "firstName": "Gauri", "lastName": "GreatLearning" }
```

Cancel

Parameter content type

application/json

Execute



Curl

```
curl -X PUT "http://localhost:8080/employeeManagement/editEmployee" -H "accept: */*" -H "Content-Type: application/json" -d "[{"email": "hi@gmail.com", "employeeId": 2, "firstName": "Gauri", "lastName": "GreatLearning"}]"
```

Request URL

http://localhost:8080/employeeManagement/editEmployee

Server response

Code	Details
200	Response body

```
[ { "employeeId": 2, "firstName": "Gauri", "lastName": "GreatLearning", "email": "hi@gmail.com" } ]
```

Download

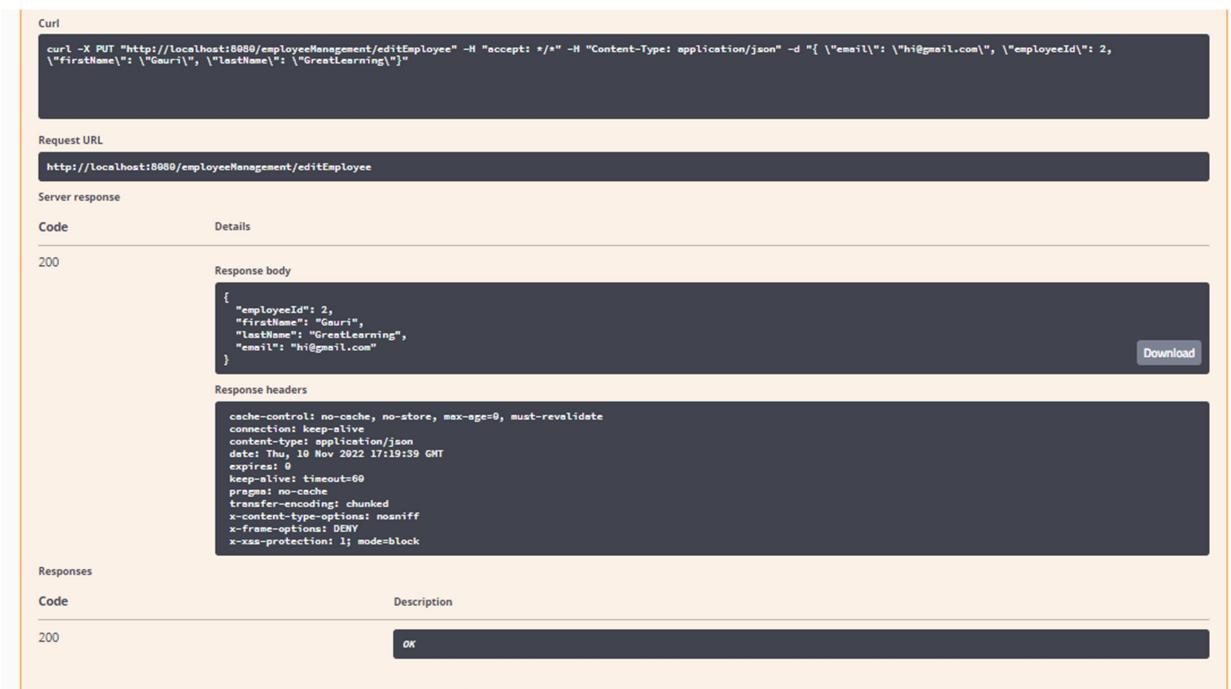
Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json
date: Fri, 10 Nov 2022 17:19:39 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
```

Responses

Code	Description
200	OK

Powered by [Swagger UI](#) | [API](#)



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

--Logged in as User

DELETE /employeeManagement/deleteEmployees/{id} deleteEmployee

Cancel

Parameters

Name	Description
id <small>required</small> integer(\$int32) (path)	id <input type="text" value="2"/>

Execute Clear

Responses Response content type */*

Curl

```
curl -X DELETE "http://localhost:8080/employeeManagement/deleteEmployees/2" -H "accept: */"
```

Request URL

```
http://localhost:8080/employeeManagement/deleteEmployees/2
```

Server response

```
Content-Type: application/json; charset=UTF-8
Date: Thu, 10 Nov 2022 17:12:10 GMT
Expires: 0
Keep-Alive: timeout=60
Pragma: no-cache
Transfer-Encoding: chunked
Vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-Xss-Protection: 1; mode=block

{
    "timestamp": "2022-11-10T17:12:10.199+00:00",
    "status": 404,
    "error": "Not Found",
    "message": "No message available",
    "path": "/employeeManagement/deleteEmployees/2"
}
```

Curl

```
curl -X DELETE "http://localhost:8080/employeeManagement/deleteEmployees/2" -H "accept: */"
```

Request URL

```
http://localhost:8080/employeeManagement/deleteEmployees/2
```

Server response

Code	Details
404 <small>Undocumented</small>	Error: Response body <pre>{ "timestamp": "2022-11-10T17:12:10.199+00:00", "status": 404, "error": "Not Found", "message": "No message available", "path": "/employeeManagement/deleteEmployees/2" }</pre> <p>Download</p> Response headers <pre>cache-control: no-cache, no-store, max-age=0, must-revalidate connection: keep-alive content-type: application/json;charset=ISO-8859-1 date: Thu, 10 Nov 2022 17:12:10 GMT expires: 0 keepalive: timeout=60 pragma: no-cache transfer-encoding: chunked vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 1; mode=block</pre>

Responses

Code	Description
------	-------------

--Logged in as Admin

DELETE /employeeManagement/deleteEmployees/{id} deleteEmployee

Cancel

Parameters

Name	Description
id * required integer(Sint32) (path)	id 5

Execute

Responses

Code Description Response content type */*

Responses

Response content type */*

Curl

```
curl -X DELETE "http://localhost:8080/employeeManagement/deleteEmployees/5" -H "accept: */*"
```

Request URL

```
http://localhost:8080/employeeManagement/deleteEmployees/5
```

Server response

Code Details

200 Response body

```
Deleted employee id - 5
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-length: 23
content-type: text/plain; charset=UTF-8
date: Fri, 10 Nov 2022 17:20:31 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
```

Responses

8. Your application should provide an endpoint to fetch an employee by his/her first name and if found more than one record then list them all-

GET /employeeManagement/searchByFirstName/{firstName} searchByFirstName

Parameters

Name	Description
firstName * required string (path)	firstName <input type="text" value="ya"/>

Responses

Code	Description
200	OK

Example Value | **Model**

```
[  
  {  
    "email": "string",  
    "employeeId": 0,  
    "firstName": "string",  
    "lastName": "string"  
  }  
]
```

Responses

Curl

```
curl -X GET "http://localhost:8080/employeeManagement/searchByFirstName/ya" -H "accept: */*"
```

Request URL

```
http://localhost:8080/employeeManagement/searchByFirstName/ya
```

Server response

Code	Details
200	Response body

```
[  
  {  
    "employeeId": 3,  
    "firstName": "Yash",  
    "lastName": "Yadav",  
    "email": "abc@xyz.com"  
  }  
]
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate  
connection: keep-alive  
content-type: application/json  
date: Mon, 10 Nov 2022 17:15:48 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
```

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

--Ascending (asc)

GET /employeeManagement/sortEmployees/order={order} sortEmployees

Cancel

Parameters

Name	Description
order <small>required</small> string (path)	order asc

Execute

Responses

Code Description

200	OK
-----	----

Curl

```
curl -X GET "http://localhost:8080/employeeManagement/sortEmployees/order=asc" -H "accept: */*"
```

Request URL

```
http://localhost:8080/employeeManagement/sortEmployees/order=asc
```

Server response

Code	Details
200	Response body

```
[{"employeeId": 2, "firstName": "Gauri S", "lastName": "Pandit", "email": "abc@gmail.com"}, {"employeeId": 4, "firstName": "Smith", "lastName": "Great Learning", "email": "abc@xyz.com"}, {"employeeId": 3, "firstName": "Yash", "lastName": "Yodv", "email": "abc@xyz.com"}]
```

Download

Response headers

--Descending (desc)

GET /employeeManagement/sortEmployees/order={order} sortEmployees

Parameters

Name	Description
order * required string (path)	order desc

Execute Clear

Responses Response content type */*

Curl

```
curl -X GET "http://localhost:8080/employeeManagement/sortEmployees/order=desc" -H "accept: */"
```

Request URL

```
http://localhost:8080/employeeManagement/sortEmployees/order=desc
```

Server response

Code	Details
200	Response body

```
[  
  {  
    "employeeId": 3,  
    "firstName": "Yash",  
    "lastName": "Yadav",  
    "email": "abc@xyz.com"  
  },  
  {  
    "employeeId": 4,  
    "firstName": "Sarithi",  
    "lastName": "Great Learning",  
    "email": "abc@xyz.com"  
  },  
  {  
    "employeeId": 2,  
    "firstName": "Gauri S",  
    "lastName": "Pendit",  
    "email": "abc@gmail.com"  
  }]
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate  
connection: keep-alive  
content-type: application/json
```