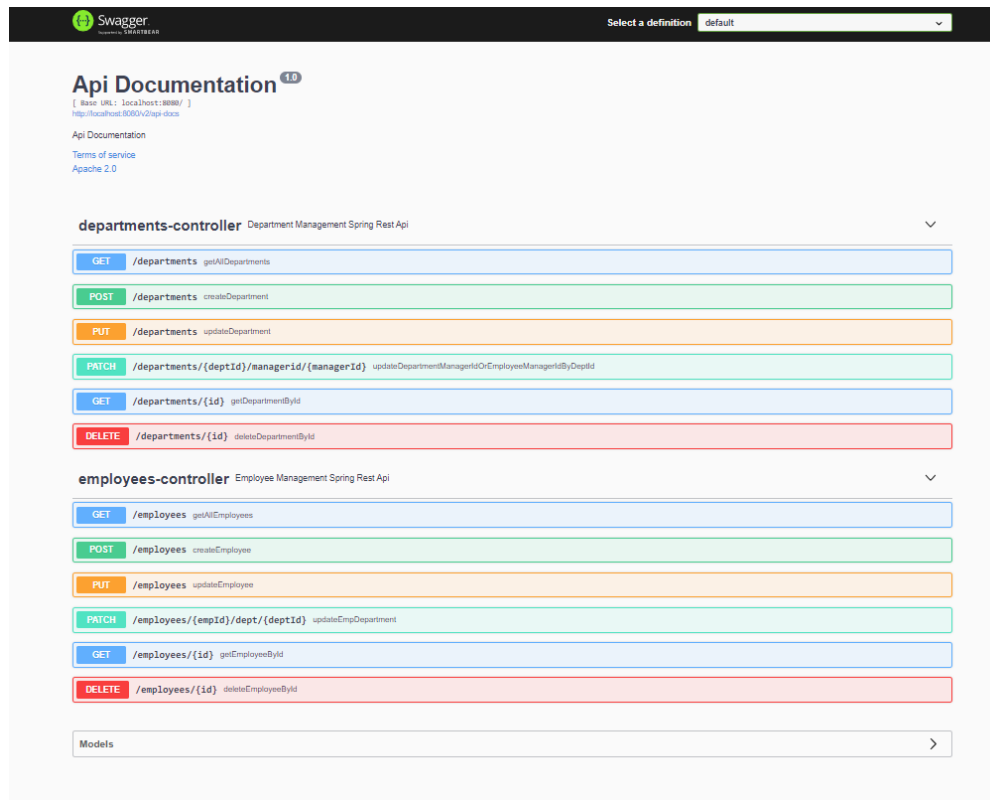


## Assignment

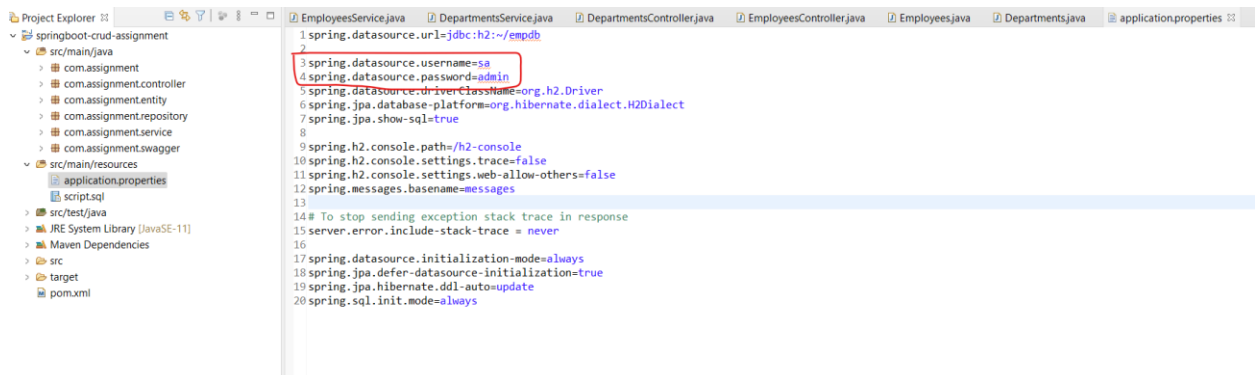
**NOTE: Screenshots attached for reference each step. Also separately Images added in Repository for better view.**

**Swagger URL :** <http://localhost:8080/swagger-ui/index.html#/>



**H2 URL :** <http://localhost:8080/h2-console/login.do?sessionId=a89806ddcdded94583e1033f36b5cc44>

**DB USER NAME : “sa” and DB USER NAME : “admin”**



jdbc:h2:~/empdb

DEPARTMENTS  
EMPLOYEES  
INFORMATION\_SCHEMA  
Sequences  
Users  
H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM DEPARTMENTS

SELECT \* FROM EMPLOYEES

SELECT \* FROM DEPARTMENTS;

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID
2	IT	101
3	HR	103
4	Admin	104
5	CS	102

(4 rows, 1 ms)

Edit

jdbc:h2:~/empdb

DEPARTMENTS  
EMPLOYEES  
INFORMATION\_SCHEMA  
Sequences  
Users  
H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM DEPARTMENTS

SELECT \* FROM EMPLOYEES

SELECT \* FROM EMPLOYEES;

EMPLOYEE_ID	EMAIL	FIRST_NAME	HIRE_DATE	LAST_NAME	MANAGER_ID	PHONE_NUMBER	SALARY	DEPARTMENT_ID
1	z4zainulhassan@gmail.com	Zain	2022-04-07 05:00:00	Hassan	104	0332-4808624	4700	4
2	haris@gmail.com	Haris	2022-04-08 05:00:00	Naeem	102	7773-8908765	8888	5
3	usman@gmail.com	Usman	2022-04-07 05:00:00	Saad	101	0332-1234567	100	2
4	ali@gmail.com	Ali	2022-04-06 05:00:00	Usman	103	2341-4444444	5600	3
5	rana@gmail.com	Rana	2022-04-06 05:00:00	Zohaib	103	567-1111111	3200	3
6	sohail@gmail.com	Sohail	2022-04-06 05:00:00	Latif	104	534-2215562	8900	4

(6 rows, 1 ms)

Edit

\*\*\*\*\*

## Employee API's :

**Get Request Pagination** : <http://localhost:8080/employees?pageNo=0&pageSize=2&sortBy=employeeId>

**Get Request By Id** : <http://localhost:8080/employees/1>

**Post Request** : <http://localhost:8080/employees>

**PUT Request** : <http://localhost:8080/employees>

**DELETE Request** : http://localhost:8080/employees/1

**PATCH Request** : http://localhost:8080/employees/3/dept/1

**JSON REQUEST Example for Employee:**

```
{  
  "firstName": "Zain",  
  "lastName": "Hassan",  
  "email": "z4zainulhassan@gmail.com",  
  "phoneNumber": "0332-4808624",  
  "hireDate": "2022-04-07",  
  "salary": 1000  
}
```

\*\*\*\*\*

**Department API's :**

**Get Request Pagination:**

http://localhost:8080/departments?pageNo=0&pageSize=3&sortBy=departmentId

**Get Request By Id** : http://localhost:8080/department/1

**Post Request** : http://localhost:8080/department

**PUT Request** : http://localhost:8080/department

**DELETE Request** : <http://localhost:8080/department/1>

**PATCH Request** : http://localhost:8080/departments/1/managerid/101

**JSON REQUEST Example for Employee:**

```
{  
  "departmentName": "IT",  
  "managerId": 101  
}
```

\*\*\*\*\*

## STEPS

In this project Employee data CRUD operations will be added separately or Department data CRUD operations will be added separately.

(Note: It's your choice to perform any operations randomly like [to add department data first or either add employee data first]).

The screenshot displays a REST client interface for a POST request to `/employees` with the `createEmployee` action. The request body is a JSON object representing an employee.

**Parameters**

Name	Description
employee	employee
object (body)	Edit Value   Model

**Request Body (JSON):**

```
{  "firstName": "Zain",  "lastName": "Hassan",  "email": "z4zainulhassan@gmail.com",  "phoneNumber": "0332-4808624",  "hireDate": "2022-04-07",  "salary": 100}
```

**Responses**

**Request URL:** `http://localhost:8080/employees`

**Server response:**

**Code:** 200

**Response body (JSON):**

```
{  "employeeId": 1,  "firstName": "Zain",  "lastName": "Hassan",  "email": "z4zainulhassan@gmail.com",  "phoneNumber": "0332-4808624",  "salary": 100,  "hireDate": "1649289600000"}
```

**Response headers:**

```
connection: keep-alive
content-type: application/json
date: Thu, 07 Apr 2022 20:23:06 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

**SQL Client Interface:**

Database: jdbc:h2:~/empdb

Tables: DEPARTMENTS, EMPLOYEES, INFORMATION\_SCHEMA, Sequences, Users

Version: H2 1.4.200 (2019-10-14)

**SQL Statement:**

```
SELECT * FROM EMPLOYEES;
```

**Result:**

EMPLOYEE_ID	EMAIL	FIRST_NAME	HIRE_DATE	LAST_NAME	MANAGER_ID	PHONE_NUMBER	SALARY	DEPARTMENT_ID
1	z4zainulhassan@gmail.com	Zain	2022-04-07 05:00:00	Hassan	null	0332-4808624	100	null

(1 row, 0 ms)

**Edit**

POST

/departments createDepartment

Parameters

Cancel

Name	Description
departments <small>* required</small>	departments
object (body)	<div>Edit Value   Model</div> <div><pre>{   "departmentName": "IT",   "managerId": 101 }</pre></div> <div>Cancel</div> <div>Parameter content type application/json</div>

ExecuteClear

Responses

Response content type \*/\*

Curl

```
curl -X POST "http://localhost:8080/departments" -H "accept: */*" -H "Content-Type: application/json" -d "{ \"departmentName\": \"IT\", \"managerId\": 101}"
```

Request URL

```
http://localhost:8080/departments
```

Server response

Code	Details
200	<div>Response body<div><pre>{   "departmentId": 1,   "departmentName": "IT",   "managerId": 101 }</pre></div><div>Download</div></div> <div>Response headers<div><pre>connection: keep-alive content-type: application/json date: Thu07 Apr 2022 20:49:47 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre></div></div>

jdbc:h2:~/empdb

DEPARTMENTS  
EMPLOYEES  
INFORMATION\_SCHEMA  
Sequences  
Users  
H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM DEPARTMENTS

SELECT \* FROM EMPLOYEES

SELECT \* FROM DEPARTMENTS;

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID
2	IT	101
3	HR	103
4	Admin	104
5	CS	102

(4 rows, 1 ms)

Edit

Suppose Employee and Department data is added successfully after all validations as mentioned in (provided REQUIRMENT file)

-> first\_name and last\_name should be not be null and more than 2 chars each

-> validate that salary is greater than 0

-> validate phone\_number contain only digits and dashes

SEE Images blow

PUT

/employees updateEmployee

Parameters

Cancel

Name	Description
employee <small>* required</small>	employee
object	
(body)	<div>Edit Value   Model</div> <div><pre>{  "employeeId": 1,  "firstName": "Zaid",  "lastName": "Bin",  "email": "abc@gmail.com",  "phoneNumber": "0332-4808624",  "hireDate": "2022-04-07",  "salary": 0}</pre></div> <div>Cancel</div> <div>Parameter content type application/json</div>

ExecuteClear

Responses

Response content type \*/

Curl

```
curl -X PUT "http://localhost:8080/employees" -H "accept: */*" -H "Content-Type: application/json" -d '{ "employeeId": 1, "firstName": "Zaid", "lastName": "Bin", "email": "abc@gmail.com", "phoneNumber": "0332-4808624", "hireDate": "2022-04-07", "salary": 0}'
```

Request URL

http://localhost:8080/employees

Server response

Code	Details
500	Error:
<small>Undocumented</small>	<div>Response body<div><pre>{  "timestamp": 1649363679962,  "status": 500,  "error": "Internal Server Error",  "message": "Employee salary must be more than 0.",  "path": "/employees"}</pre></div><div>Download</div></div> <div>Response headers<div>connection: close content-type: application/json date: Thu07 Apr 2022 20:34:39 GMT transfer-encoding: chunked</div></div>

PUT

/employees updateEmployee

## Parameters

Cancel

Name Description

employee \* required

employee

object

Edit Value | Model

(body)

```
{
  "employeeId": 1,
  "firstName": "Usman",
  "lastName": "Dr",
  "email": "z4rainulhassan@gmail.com",
  "phoneNumber": "9332-488624",
  "hireDate": "2022-04-07",
  "salary": 100
}
```

Cancel

Parameter content type

application/json

Execute

Clear

## Responses

Response content type

json

## Curl

```
curl -X PUT "http://localhost:8080/employees" -H "accept: */*" -H "Content-Type: application/json" -d '{"employeeId": 1, "firstName": "Usman", "lastName": "Dr", "email": "z4rainulhassan@gmail.com", "phoneNumber": "9332-488624", "hireDate": "2022-04-07", "salary": 100}'
```

## Request URL

http://localhost:8080/employees

## Server response

Code

Details

500

Undocumented

Error:

Response body

```
{
  "timestamp": 1648363596162,
  "status": 500,
  "error": "Internal Server Error",
  "message": "Employee lastname length must be more than 2 characters.",
  "path": "/employees"
}
```

Download

Response headers

```
connection: close
content-type: application/json
date: Thu 07 Apr 2022 20:33:16 GMT
transfer-encoding: chunked
```



PUT

/employees updateEmployee

Parameters

Cancel

Name	Description
employee <small>* required</small>	employee
object (body)	<div>Edit Value   Model</div> <div><pre>{   "employeeId": 1,   "firstName": "Al",   "lastName": "Hassan",   "email": "z4rainulhassan@gmail.com",   "phoneNumber": "0332-4808624",   "hireDate": "2022-04-07",   "salary": 100 }</pre></div> <div>Cancel</div> <div>Parameter content type application/json</div>

ExecuteClear

Responses

Response content type \*/\*

Curl

```
curl -X PUT "http://localhost:8080/employees" -H "accept: */*" -H "Content-Type: application/json" -d "{ \"employeeId\": 1, \"firstName\": \"Al\", \"lastName\": \"Hassan\", \"email\": \"z4rainulhassan@gmail.com\", \"phoneNumber\": \"0332-4808624\", \"hireDate\": \"2022-04-07\", \"salary\": 100}"
```

Request URL

```
http://localhost:8080/employees
```

Server response

Code	Details
500 <small>undocumented</small>	<div>Error:</div> <div>Response body</div> <div><pre>{   "timestamp": 1649363357644,   "status": 500,   "error": "Internal Server Error",   "message": "Employee firstame length must be more than 2 characters.",   "path": "/employees" }</pre></div> <div>Download</div> <div>Response headers</div> <div><pre>connection: close content-type: application/json date: Thu07 Apr 2022 20:29:17 GMT transfer-encoding: chunked</pre></div>

PUT

/employees

updateEmployee

Parameters

Cancel

Name	Description
employee * required	employee
object	
(body)	Edit Value   Model

```
{  "employeeId": 1,  "firstName": "Zaid",  "lastName": "Bin",  "email": "abc@gmail.com",  "phoneNumber": "0332-4808624 ABC * % # @ ! ",  "hireDate": "2022-04-07",  "salary": 1000}
```

Cancel

Parameter content type

application/json

Execute

Clear

Responses

Response content type

Curl

```
curl -X PUT "http://localhost:8080/employees" -H "accept: /*" -H "Content-Type: application/json" -d '{ "employeeId": 1, "firstName": "Zaid", "lastName": "Bin", "email": "abc@gmail.com", "phoneNumber": "0332-4808624 ABC * % # @ ! ", "hireDate": "2022-04-07", "salary": 1000}'
```

Request URL

http://localhost:8080/employees

Server response

Code	Details
500	Error:
Undocumented	Response body

```
{  "timestamp": 1649363774403,  "status": 500,  "error": "Internal Server Error",  "message": "Employee phone number contain only digits and dashes.",  "path": "/employees"}
```

Download

Response headers

```
connection: close
content-type: application/json
date: Thu07 Apr 2022 20:36:14 GMT
transfer-encoding: chunked
```

Now, we want to assign an employee a Manager ID call this API (Mapping type is PATCH-MAPPING)  
`http://localhost:8080/employees/3/dept/1` (Note: 3 is an employee id or 1 is a department id)

Hit Request: `manager_id` or `department_id` will be updated in Employee Table successfully and get all specific fields record from Department Table.

PATCH

/employees/{empId}/dept/{deptId} updateEmpDepartment

Parameters

Cancel

Name	Description
<div><div>deptId * required</div><div>integer(\$int32)</div><div>(path)</div></div>	<div>deptId</div> <div><div>1</div></div>
<div><div>empId * required</div><div>integer(\$int32)</div><div>(path)</div></div>	<div>empId</div> <div><div>3</div></div>

Execute

Clear

Responses

Response content type \*/\*

Curl

```
curl -X PATCH "http://localhost:8080/employees/3/dept/1" -H "accept: */*"
```

Request URL

```
http://localhost:8080/employees/3/dept/1
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{  "employeeId": 3,  "firstName": "Usman",  "lastName": "Saad",  "email": "abc@gmail.com",  "phoneNumber": "0332-1234567",  "salary": 100,  "hireDate": 1649289600000}</pre></div><div>Download</div></div>

jdbc:h2:~/empdb | Auto commit | Max rows: 1000 | Auto complete: Off | Auto select: On

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

SELECT \* FROM DEPARTMENTS

SELECT \* FROM EMPLOYEES

SELECT \* FROM EMPLOYEES;

EMPLOYEE_ID	EMAIL	FIRST_NAME	HIRE_DATE	LAST_NAME	MANAGER_ID	PHONE_NUMBER	SALARY	DEPARTMENT_ID
1	z4zainulhassan@gmail.com	Zain	2022-04-07 05:00:00	Hassan	null	0332-4808624	100	null
2	abc@gmail.com	Haris	2022-04-07 05:00:00	Hassan	null	0332-1234567	100	null
3	abc@gmail.com	Usman	2022-04-07 05:00:00	Saad	102	0332-1234567	100	1
4	abc@gmail.com	Sheikh	2022-04-07 05:00:00	Qasim	null	0332-1234567	100	null
5	abc@gmail.com	Ali	2022-04-07 05:00:00	Saeed	null	0332-1234567	100	null
6	abc@gmail.com	Musa	2022-04-07 05:00:00	Ali	null	0332-1234567	100	null

(6 rows, 3 ms)

Edit

Suppose in Department Table department\_name or manager\_id is already existing and user want to (create or update) new Department Name, error message will be shown like

"Department name already exists. Please try another name."

"Department Manager Id already exists. Please try another name."

PUT

/departments

updateDepartment

Parameters

Cancel

Name	Description
departments <small>* required</small>	departments
object (body)	<div>Edit Value   Model</div> <div><pre>{   "departmentId": 1,   "departmentName": "Admin",   "managerId": 102 }</pre></div> <div>Cancel</div> <div>Parameter content type application/json</div>

ExecuteClear

Responses

Response content type \*/\*

Curl

curl -X PUT "http://localhost:8080/departments" -H "accept: \*/\*" -H "Content-Type: application/json" -d "{ \"departmentId\": 1, \"departmentName\": \"Admin\", \"managerId\": 102}"

Request URL

http://localhost:8080/departments

Server response

Code	Details
200	<div>Response body</div> <div><pre>{   "departmentId": 1,   "departmentName": "Admin",   "managerId": 102 }</pre><div>Download</div></div> <div>Response headers</div> <div>connection: keep-alive content-type: application/json date: Thu07 Apr 2022 20:54:14 GMT keep-alive: timeout=60 transfer-encoding: chunked</div>

PUT

/departments

updateDepartment

Parameters

Cancel

Name	Description
departments <span>required</span>	departments
object (body)	<div>Edit Value   Model</div> <div><pre>{   "departmentId": 1,   "departmentName": "Admin",   "managerId": 101 }</pre></div> <div>Cancel</div> <div>Parameter content type application/json</div>

ExecuteClear

Responses

Response content type \*/\*

Curl

```
curl -X PUT "http://localhost:8080/departments" -H "accept: */*" -H "Content-Type: application/json" -d '{"departmentId": 1, "departmentName": "Admin", "managerId": 101}'
```

Request URL

```
http://localhost:8080/departments
```

Server response

Code	Details
500	Error:
undocumented	<div>Response body</div> <div><pre>{   "timestamp": 1649364775959,   "status": 500,   "error": "Internal Server Error",   "message": "Department Manager Id already exists.Please try another name.",   "path": "/departments" }</pre></div> <div>Download</div> <div>Response headers</div> <div><pre>connection: close content-type: application/json date: Thu09 Apr 2022 20:52:55 GMT transfer-encoding: chunked</pre></div>

PUT /departments updateDepartment

Parameters

Cancel

Name	Description
departments * required	departments
object (body)	departments Edit Value   Model

```
{
  "departmentId": 1,
  "departmentName": "IT",
  "managerId": 101
}
```

Cancel

Parameter content type  
application/json

Execute Clear

Responses

Response content type \*/\*

Curl

```
curl -X PUT "http://localhost:8080/departments" -H "accept: */*" -H "Content-Type: application/json" -d "{ \"departmentId\": 1, \"departmentName\": \"IT\", \"managerId\": 101}"
```

Request URL

```
http://localhost:8080/departments
```

Server response

Code	Details
500 undocumented	<div>Error:</div> <div>Response body</div> <div><pre>{   "timestamp": 1649364699448,   "status": 500,   "error": "Internal Server Error",   "message": "Department name already exists. Please try another name.",   "path": "/departments" }</pre></div> <div>Download</div>

Response headers

```
connection: close
content-type: application/json
date: Thu07 Apr 2022 20:51:39 GMT
transfer-encoding: chunked
```

If user wants to update only manager\_id field in Department Table separate API will be call API (Mapping type is PATCH-MAPPING) `http://localhost:8080/departments/1/managerid/109`

(Note: 1 is department id or 109 is a new manager id).

Hit Request: manager\_id will be updated in Department Table or Employee Table and value set in both tables against specific department id.

PATCH

/departments/{deptId}/managerId/{managerId} updateDepartmentManagerIdOrEmployeeManagerIdByDeptId

Parameters

Cancel

Name	Description
<b>deptId</b> * required integer(\$int32) (path)	deptId
<input type="text" value="1"/>	
<b>managerId</b> * required integer(\$int32) (path)	managerId
<input type="text" value="109"/>	

Execute

Clear

Responses

Response content type \*/\*

Curl

curl -X PATCH "http://localhost:8080/departments/1/managerId/109" -H "accept: \*/\*"

Request URL

http://localhost:8080/departments/1/managerId/109

Server response

Code	Details
200	<div><div>Response body</div><div>{   "departmentId": 1,   "departmentName": "Admin",   "managerId": 109 }</div><div>Download</div></div> <div><div>Response headers</div><div>connection: keep-alive content-type: application/json date: Thu87 Apr 2022 21:07:27 GMT keep-alive: timeout=60 transfer-encoding: chunked </div></div>

Suppose we want to delete any department record but it's value is assigned to Employee Table as well will be updated and all values update into 0 and null

(For Example: manager\_id = 0 and department\_id = null).



jdbc:h2:~/empdb  
DEPARTMENTS  
EMPLOYEES  
INFORMATION\_SCHEMA  
Sequences  
Users  
H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM DEPARTMENTS

SELECT \* FROM EMPLOYEES

SELECT \* FROM EMPLOYEES;

EMPLOYEE_ID	EMAIL	FIRST_NAME	HIRE_DATE	LAST_NAME	MANAGER_ID	PHONE_NUMBER	SALARY	DEPARTMENT_ID
1	z4zainulhassan@gmail.com	Zain	2022-04-07 05:00:00	Hassan	null	0332-4808624	100	null
2	abc@gmail.com	Haris	2022-04-07 05:00:00	Hassan	null	0332-1234567	100	null
3	abc@gmail.com	Usman	2022-04-07 05:00:00	Saad	0	0332-1234567	100	null
4	abc@gmail.com	Sheikh	2022-04-07 05:00:00	Qasim	null	0332-1234567	100	null
5	abc@gmail.com	Ali	2022-04-07 05:00:00	Saeed	null	0332-1234567	100	null
6	abc@gmail.com	Musa	2022-04-07 05:00:00	Ali	null	0332-1234567	100	null

(6 rows, 2 ms)

Edit

Pagination added and also set default values can be seen in Swagger Employee.

### Requests.

Employee requests (Get Pagination , Get By Id , DELETE)

Attach below;

GET

/employees

getAllEmployees

Parameters

Cancel

Name	Description
pageNo	
integer(int32)	pageNo
(query)	
	0
pageSize	
integer(int32)	pageSize
(query)	
	2
sortBy	
string	sortBy
(query)	
	employeeid

ExecuteClear

Responses

Response content type

\*/

Curl

curl -X GET "http://localhost:8080/employees?pageNo=0&pageSize=2&sortBy=employeeid" -H "accept: \*/"

Request URL

http://localhost:8080/employees?pageNo=0&pageSize=2&sortBy=employeeid

Server response

Code

Details

200

Response body

Content: {

{

"employeeid": 1,

"firstName": "Zain",

"lastName": "Hasan",

"email": "zainulhasan@gmail.com",

"phoneArea": "8332-488624",

"salary": 100,

"hireDate": 1649289600000

}

,

{

"employeeid": 2,

"firstName": "Harris",

"lastName": "Hasan",

"email": "h2@gmail.com",

"phoneArea": "8332-1234567",

"salary": 100,

"hireDate": 1649289600000

}

}

"pageable": {

"work": {

"empty": false,

"sorter": true,

"sorted": false

}

},

"offset": 0,

"pageSize": 2,

}

Download

Response headers

GET

/employees/{id} getEmployeeById

Parameters

Cancel

Name	Description
id * required	
integer(\$int32)id	
(path)	

1

Execute

Clear

Responses

Response content type 

\*/

Curl

curl -X GET "http://localhost:8080/employees/1" -H "accept: \*/"

Request URL

http://localhost:8080/employees/1

Server response

Code	Details
200	<div><div>Response body</div><div><div>{   "employeeId": 1,   "firstName": "Zain",   "lastName": "Hassan",   "email": "z4zainulhassan@gmail.com",   "phoneNumber": "0332-4808624",   "salary": 100,   "hireDate": 1649289600000 }</div><div>Download</div></div></div> <div><div>Response headers</div><div>connection: keep-alive content-type: application/json date: Thu07 Apr 2022 20:37:43 GMT keep-alive: timeout=60 transfer-encoding: chunked</div></div>

POST

/employees createEmployee

PUT

/employees updateEmployee

PATCH

/employees/{empId}/dept/{deptId} updateEmpDepartment

GET

/employees/{id} getEmployeeById

DELETE

/employees/{id} deleteEmployeeById

Parameters

Cancel

Name	Description
id <small>* required</small>	Integer(\$int32)id (path)

7

Execute

Clear

Responses

Response content type \*/\*

Curl

curl -X DELETE "http://localhost:8080/employees/7" -H "accept: \*/"

Request URL

http://localhost:8080/employees/7

Server response

Code	Details
200	<div>Response headers</div> <div> connection: keep-alive  content-length: 0  date: Thu09 Apr 2022 20:48:08 GMT  keep-alive: timeout=60 </div>

Pagination added and also set default values can be seen in Swagger Department.

### Requests.

Department requests (Get Pagination , Get By Id , DELETE)

Attach below;

## departments-controller Department Management Spring Rest Api



GET

/departments

getAllDepartments

POST

/departments

createDepartment

PUT

/departments

updateDepartment

PATCH

/departments/{deptId}/managerid/{managerId}

updateDepartmentManagerIdOrEmployeeManagerIdByDepId

GET

/departments/{id}

getDepartmentById

Parameters

Cancel

Name	Description
id <small>* required</small>	
integer(\$int32)id	
(path)	

Execute

Clear

Responses

Response content type \*/\*

Curl

```
curl -X GET "http://localhost:8080/departments/2" -H "accept: */*"
```

Request URL

```
http://localhost:8080/departments/2
```

Server response

Code	Details
200	<div>Response body<pre>{   "departmentId": 2,   "departmentName": "IT",   "managerId": 101 }</pre><div>Download</div></div> <div>Response headers<pre>connection: keep-alive content-type: application/json date: Thu 07 Apr 2022 21:02:41 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre></div>

POST

/departments createDepartment

PUT

/departments updateDepartment

PATCH

/departments/{deptId}/managerId/{managerId} updateDepartmentManagerIdOrEmployeeManagerIdByDeptId

GET

/departments/{id} getDepartmentById

DELETE

/departments/{id} deleteDepartmentById

Parameters

Cancel

Name	Description
id <small>required</small>	
integer(\$int32)id	
(path)	
	1

Execute

Clear

Responses

Response content type \*/\*

Curl

curl -X DELETE "http://localhost:8080/departments/1" -H "accept: \*/\*"

Request URL

http://localhost:8080/departments/1

Server response

Code	Details
200	<div><div>Response headers</div><div>connection: keep-alive content-length: 0 date: Thu07 Apr 2022 21:11:12 GMT keep-alive: timeout=60</div></div>

departments-controller

Department Management Spring Rest Api

▼

GET

/departments

getAllDepartments

Parameters

Cancel

Name	Description
pageNo	
integer(\$int32)pageNo	
(query)	
	0
pageSize	
integer(\$int32)pageSize	
(query)	
	3
sortBy	
string	
(query)	
	departmentId

ExecuteClear

Responses

Response content type \*/\*

Curl

curl -X GET "http://localhost:8080/departments?pageNo=0&pageSize=3&sortBy=departmentId" -H "accept: \*/\*"

Request URL

http://localhost:8080/departments?pageNo=0&pageSize=3&sortBy=departmentId

Server response

CodeDetails

200

Response body

```
{
  "content": [
    {
      "departmentId": 1,
      "departmentName": "Admin",
      "managerId": 102
    },
    {
      "departmentId": 2,
      "departmentName": "IT",
      "managerId": 101
    },
    {
      "departmentId": 3,
      "departmentName": "HR",
      "managerId": 103
    }
  ],
  "pageable": {
    "sort": {
      "empty": false,
      "sorted": true,
      "unsorted": false
    },
    "offset": 0,
    "pageSize": 3,
    "pageNumber": 0,
    "paged": true,
    "unpaged": false
  }
}
```

Download

**All other possible exceptions or checks are handled properly.**

Test all these Api's on postman as well.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign InCreate Account

Scratch PadNewImport

GET Get All Employee Data

No Environment

Collections+New Collection

GET Get All Employee Data

GET Get Single Employee Data

POST Post Employee Data

PUT Put Employee Data

DEL Delete Single Employee Data

GET Get Single Department Data

GET Get Department Data

POST Post Department Data

PATCH PATCH Employee Data From De...

PUT PUT Department Data

PATCH Patch Department or Employee...

DEL Delete Single Department Data

New Collection / Get All Employee Data

Save

Send

GET

http://localhost:8080/employees?pageNo=0&pageSize=2&sortBy=employeeId

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> pageNo	0		
<input checked="" type="checkbox"/> pageSize	2		
<input checked="" type="checkbox"/> sortBy	employeeId		
Key	Value	Description	

BodyCookiesHeaders (5)Test Results

Status: 200 OK Time: 10 ms Size: 789 B Save Response

PrettyRawPreviewVisualizeJSON

```
1
2  "content": [
3    {
4      "employeeId": 1,
5      "firstName": "Zain",
6      "lastName": "Hassan",
7      "email": "z4zaimulhassan@gmail.com",
8      "phoneNumber": "8332-4888624",
9      "salary": 4788,
10     "hireDate": 1649289600000
11   },
12   {
13     "employeeId": 2,
```

Find and ReplaceConsoleRunnerTrash