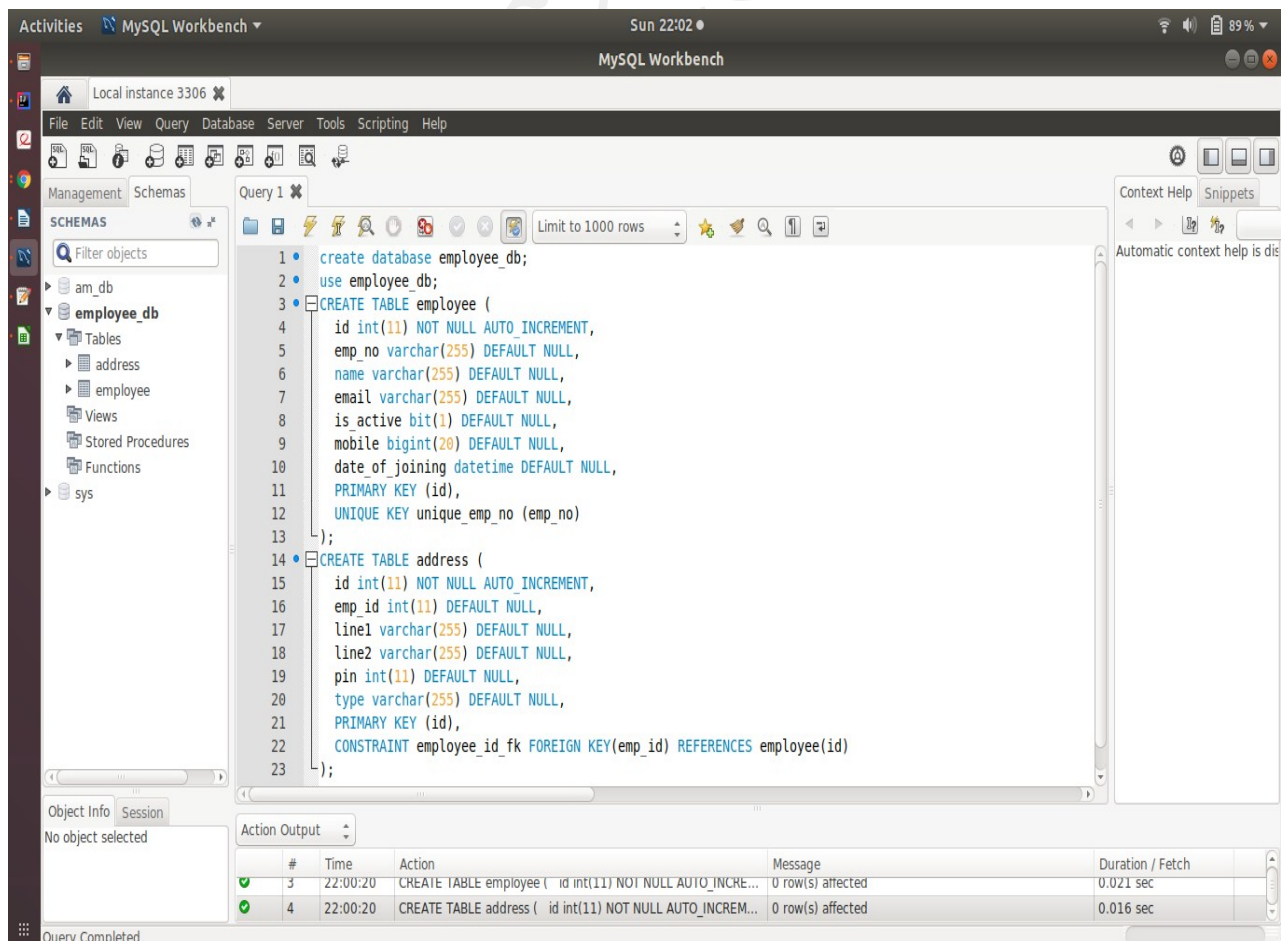


## springboot-jpa-restful-crud

- Execute this database script in your mysql workbench and update the database username,port,password in **application.properties** files inside the maven spring-boot project as per your database configuration.

### database script.sql


```
create database employee_db;
use employee_db;
CREATE TABLE employee (
  id int(11) NOT NULL AUTO_INCREMENT,
  emp_no varchar(255) DEFAULT NULL,
  name varchar(255) DEFAULT NULL,
  email varchar(255) DEFAULT NULL,
  is_active bit(1) DEFAULT NULL,
  mobile bigint(20) DEFAULT NULL,
  date_of_joining datetime DEFAULT NULL,
  PRIMARY KEY (id),
  UNIQUE KEY unique_emp_no (emp_no)
);
CREATE TABLE address (
  id int(11) NOT NULL AUTO_INCREMENT,
  emp_id int(11) DEFAULT NULL,
  line1 varchar(255) DEFAULT NULL,
  line2 varchar(255) DEFAULT NULL,
  pin int(11) DEFAULT NULL,
  type varchar(255) DEFAULT NULL,
  PRIMARY KEY (id),
  CONSTRAINT employee_id_fk FOREIGN KEY(emp_id) REFERENCES employee(id)
);
```



```
Address newlyAddedAddress =  
    savedEmployee.getAddress().get(savedEmployee.getAddress().size()-1);  
AddressDTO newlyAddedAddressDTO =  
    addressAssembler.fromDomain(newlyAddedAddress);  
return newlyAddedAddressDTO;  
}  
}
```

- Run the Application.java
- Open the browser, hit this url to see the rest-api in swagger docs
- <http://localhost:8999/swagger-ui.html>
- [http://localhost:8999/swagger-ui.html#/Employee\\_CRUD\\_Operation](http://localhost:8999/swagger-ui.html#/Employee_CRUD_Operation)

localhost:8999/swagger-ui.html

 **swagger**

default (/v2/api-docs)  Explore

## Api Documentation

Api Documentation

[Apache 2.0](#)

### Employee CRUD Operation : Employee Api

Show/Hide | List Operations | Expand Operations

GET	/api/employeeByName	to fetch all employee by employee name
POST	/api/employees	to create an employee in the system
PUT	/api/employees	to update an employee in the system
DELETE	/api/employees/{employeeNo}	to delete an employee by employeeNo
GET	/api/employees/{employeeNo}	to fetch an employee by employeeNo
PATCH	/api/employees/{employeeNo}	to update the email of an employee
GET	/api/employees/{employeeNo}/addresses	to fetch all address of an employee by employeeNo
POST	/api/employees/{employeeNo}/addresses	to add the address of an employee by employeeNo
GET	/api/employees/{employeeNo}/addresses/{addressId}	to fetch address of an employee by employeeNo and addressId

[ BASE URL: / , API VERSION: 1.0 ]

- To create an employee use this sample input

```
{
  "name": "Alice",
  "email": "alice@gmail.com",
  "mobile": 8790890987,
  "dateOfJoining": "2019-12-17T11:07:25.007Z",
  "active": true,
  "addressDTOS": [
    {
      "line1": "Prestige Villa",
      "line2": "Hebbal",
      "pin": 560024,
      "type": "PRESENT"
    }
  ]
}
```

POST

/api/employees

to create an employee in the system

Implementation Notes

This api is a valid restful service

Response Class (Status 200)

OK

Model | Model Schema

```
{}
```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
employee	<pre>{   "name": "Alice",   "email": "alice@gmail.com",   "mobile": 8790890987,   "dateOfJoining": "2019-12-17T11:07:25.007Z",   "active": true,   "addressDTOS": [     {       "line1": "Prestige Villa",       "line2": "Hebbal",       "pin": 560024,       "type": "PRESENT"     }   ] }</pre>	employee	body	Model   Model Schema <pre>{   "active": true,   "addressDTOS": [     {       "id": 0,       "line1": "string",       "line2": "string",       "pin": 0,       "type": "string"     }   ],   "dateOfJoining": "2019-02-07T11:07:25.007Z" }</pre> Click to set as parameter value

Parameter content type:

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out! [Hide Response](#)

- Click on the Try it out button and see the response as below

Try it out!

[Hide Response](#)

### Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "name": "Alice",
  "email": "alice@gmail.com",
  "mobile": 8790890987,
  "dateOfJoining": "2019-12-17T11:07:25.007Z",
  "active": true,
  "addressDTOS": [
    {
      "line1": "Prestige Villa",
      "line2": "Hebbal",
      "pin": 560024,
      "type": "PRESENT"
    }
  ]
}' http://localhost:8999/api/employees'
```

### Request URL

http://localhost:8999/api/employees

### Request Headers

```
{
  "Accept": "*/*"
}
```

### Response Body

```
{
  "id": 1,
  "employeeNo": "woc-1",
  "name": "Alice",
  "email": "alice@gmail.com",
  "mobile": 8790890987,
  "dateOfJoining": "2019-12-17T11:07:25.007+0000",
  "addressDTOS": [
    {
      "id": 1,
      "line1": "Prestige Villa",
      "line2": "Hebbal",
      "pin": 560024,
      "type": "PRESENT"
    }
  ],
  "active": true
}
```

### Response Code

200

### Response Headers

```
{
  "content-type": "application/json;charset=UTF-8",
  "date": "Sun, 09 Feb 2020 16:07:21 GMT",
  "transfer-encoding": "chunked"
}
```

- Open the mysql workbench, perform the select query and see the employee has been created along with the given input address

The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' panel on the left displays the 'employee\_db' database with tables 'address' and 'employee'. The 'Query 1' editor contains the SQL statement: `select * from employee;`. The 'Result Grid' shows one row of data for an employee with ID 1, name Alice, and email alice@gmail.com. The 'Action Output' panel at the bottom shows the query execution details, including the time (22:05:00) and the message '1 row(s) returned'.

#	id	emp_no	name	email	is_active	mobile	date_of_joining
1	1	woc-1	Alice	alice@gmail.com	1	8790890987	2019-12-17 16:37:25

#	Time	Action	Message	Duration / Fetch
5	22:04:35	select * from employee LIMIT 0, 1000	1 row(s) returned	0.00062 sec / 0.000...
6	22:05:00	select * from address LIMIT 0, 1000	1 row(s) returned	0.00072 sec / 0.000...

The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' panel on the left displays the 'employee\_db' database with tables 'address' and 'employee'. The 'Query 1' editor contains the SQL statement: `select * from address;`. The 'Result Grid' shows one row of data for an address with ID 1, emp\_id 1, line1 'Prestige Villa', line2 'Hebbal', pin '560024', and type 'PRESENT'. The 'Action Output' panel at the bottom shows the query execution details, including the time (22:05:00) and the message '1 row(s) returned'.

#	id	emp_id	line1	line2	pin	type
1	1	1	Prestige Villa	Hebbal	560024	PRESENT

#	Time	Action	Message	Duration / Fetch
5	22:04:35	select * from employee LIMIT 0, 1000	1 row(s) returned	0.00062 sec / 0.000...
6	22:05:00	select * from address LIMIT 0, 1000	1 row(s) returned	0.00072 sec / 0.000...

➤ you can search the employee by employeeNo using the below api

**GET** `/api/employees/{employeeNo}` to fetch an employee by employeeNo

**Implementation Notes**  
This api is a valid restful service

**Response Class (Status 200)**  
OK

**Model** | **Model Schema**

```
{}
```

**Response Content Type** \*/\*

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>employeeNo</b>	<input type="text" value="woc-1"/>	<b>employeeNo</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

➤ Click on the Try it out button and see the response as below

Try it out!

[Hide Response](#)

**Curl**

```
curl -X GET --header 'Accept: application/json' 'http://localhost:8999/api/employees/woc-1'
```

**Request URL**

```
http://localhost:8999/api/employees/woc-1
```

**Request Headers**

```
{
  "Accept": "*/*"
}
```

**Response Body**

```
{
  "id": 1,
  "employeeNo": "woc-1",
  "name": "Alice",
  "email": "alice@gmail.com",
  "mobile": 8790890987,
  "dateOfJoining": "2019-12-17T11:07:25.000+0000",
  "addressDTOS": [
    {
      "id": 1,
      "line1": "Prestige Villa",
      "line2": "Hebbal",
      "pin": 560024,
      "type": "PRESENT"
    }
  ],
  "active": true
}
```

**Response Code**

```
200
```

**Response Headers**

```
{
  "content-type": "application/json;charset=UTF-8",
  "date": "Sun, 09 Feb 2020 16:37:38 GMT",
  "transfer-encoding": "chunked"
}
```

➤ To update the email use this below api

**PATCH** `/api/employees/{employeeNo}` to update the email of an employee

**Implementation Notes**  
This api is a valid restful service

**Response Class (Status 200)**  
OK

**Model** | **Model Schema**

```
{}
```

Response Content Type

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
employeeNo	woc-1	employeeNo	path	string
email	<div>alice.jemima@gmail.com</div>	email	body	string

Parameter content type:

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
204	No Content		
401	Unauthorized		
403	Forbidden		

Try it out!

➤ Click on the Try it out button and see the response as below

Try it out!

Hide Response

**Curl**

```
curl -X PATCH --header 'Content-Type: application/json' --header 'Accept: application/json' -d 'alice.jemima@gmail.com'
```

**Request URL**

```
http://localhost:8999/api/employees/woc-1
```

**Request Headers**

```
{
  "Accept": "*/*"
}
```

**Response Body**

```
{
  "id": 1,
  "employeeNo": "woc-1",
  "name": "Alice",
  "email": "alice.jemima@gmail.com",
  "mobile": 8790890987,
  "dateOfJoining": "2019-12-17T11:07:25.000+0000",
  "addressDTOS": [
    {
      "id": 1,
      "line1": "Prestige Villa",
      "line2": "Hebbal",
      "pin": 560024,
      "type": "PRESENT"
    }
  ],
  "active": true
}
```

**Response Code**

```
200
```

**Response Headers**

```
{
  "content-type": "application/json; charset=UTF-8",
  "date": "Sun, 09 Feb 2020 16:40:47 GMT",
  "transfer-encoding": "chunked"
}
```

➤ To add a permanent address for employee use this below api

**POST** `/api/employees/{employeeNo}/addresses` to add the address of an employee by employeeNo

**Implementation Notes**  
This api is a valid restful service, with HATEOAS

**Response Class (Status 200)**  
OK

**Model** | **Model Schema**

```
{}
```

**Response Content Type** \*/\*

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>employeeNo</b>	<input type="text" value="woc-1"/>	<b>employeeNo</b>	path	string
<b>address</b>	<pre>{   "line1": "Silver Villa",   "line2": "MSR Nagar",   "pin": 560054,   "type": "PERMANENT" }</pre>	<b>address</b>	body	<b>Model</b>   <b>Model Schema</b>

**Model** | **Model Schema**

```
{
  "id": 0,
  "line1": "string",
  "line2": "string",
  "pin": 0,
  "type": "string"
}
```

Parameter content type: application/json

Click to set as parameter value

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

➤ Click on the Try it out button and see the response as below

Try it out!

[Hide Response](#)

**Curl**

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: */*' -d '{
  "line1": "Silver Villa",
  "line2": "MSR Nagar",
  "pin": 560054,
  "type": "PERMANENT"
}' 'http://localhost:8999/api/employees/woc-1/addresses'
```

**Request URL**

```
http://localhost:8999/api/employees/woc-1/addresses
```

**Request Headers**

```
{
  "Accept": "*/*"
}
```

**Response Body**

```
no content
```

**Response Code**

```
201
```

**Response Headers**

```
{
  "content-length": "0",
  "date": "Sun, 09 Feb 2020 16:43:37 GMT",
  "location": "http://localhost:8999/api/employees/woc-1/addresses/2",
  "content-type": null
}
```



➤ To search an employee by name, use this below api

**Employee CRUD Operation : Employee Api**

Show/Hide | List Operations | Expand Operations

GET /api/employeeByName to fetch all employee by employee name

**Implementation Notes**  
This api is not a valid restful service

**Response Class (Status 200)**  
OK

Model | Model Schema

```
{}
```

Response Content Type \*/\* ▼

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
name	Alice	name	query	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out! [Hide Response](#)

➤ Click on the Try it out button and see the response as below

Try it out! [Hide Response](#)

**Curl**

```
curl -X GET --header 'Accept: application/json' 'http://localhost:8999/api/employeeByName?name=Alice'
```

**Request URL**

```
http://localhost:8999/api/employeeByName?name=Alice
```

**Request Headers**

```
{
  "Accept": "*/*"
}
```

**Response Body**

```
[
  {
    "id": 1,
    "employeeNo": "woc-1",
    "name": "Alice",
    "email": "alice.jemima@gmail.com",
    "mobile": 8790890987,
    "dateOfJoining": "2019-12-17T11:07:25.000+0000",
    "addressDTOS": [
      {
        "id": 1,
        "line1": "Prestige Villa",
        "line2": "Hebbal",
        "pin": 560024,
        "type": "PRESENT"
      },
      {
        "id": 2,
        "line1": "Silver Villa",
        "line2": "MSR Nagar",
        "pin": 560054,
        "type": "PRESENT"
      }
    ]
  }
]
```

**Response Code**

```
200
```

**Response Headers**

```
{
  "content-type": "application/json;charset=UTF-8",
  "date": "Sun, 09 Feb 2020 16:45:09 GMT",
  "transfer-encoding": "chunked"
}
```

➤ To delete and employee, use this below api

**DELETE** `/api/employees/{employeeNo}` to delete an employee by employeeNo

**Implementation Notes**  
This api is a valid restful service

**Response Class (Status 200)**  
OK

Model | Model Schema

```
{}
```

Response Content Type

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
employeeNo	woc-1	employeeNo	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
204	No Content		
401	Unauthorized		
403	Forbidden		

Try it out!

➤ Click on the Try it out button and see the response as below

Try it out!

Hide Response

**Curl**

```
curl -X DELETE --header 'Accept: text/plain' 'http://localhost:8999/api/employees/woc-1'
```

**Request URL**

```
http://localhost:8999/api/employees/woc-1
```

**Request Headers**

```
{
  "Accept": "*/*"
}
```

**Response Body**

```
employeeNo=woc-1, has been deleted successfully
```

**Response Code**

```
200
```

**Response Headers**

```
{
  "content-length": "47",
  "content-type": "text/plain; charset=UTF-8",
  "date": "Sun, 09 Feb 2020 16:51:39 GMT"
}
```

Thank You!