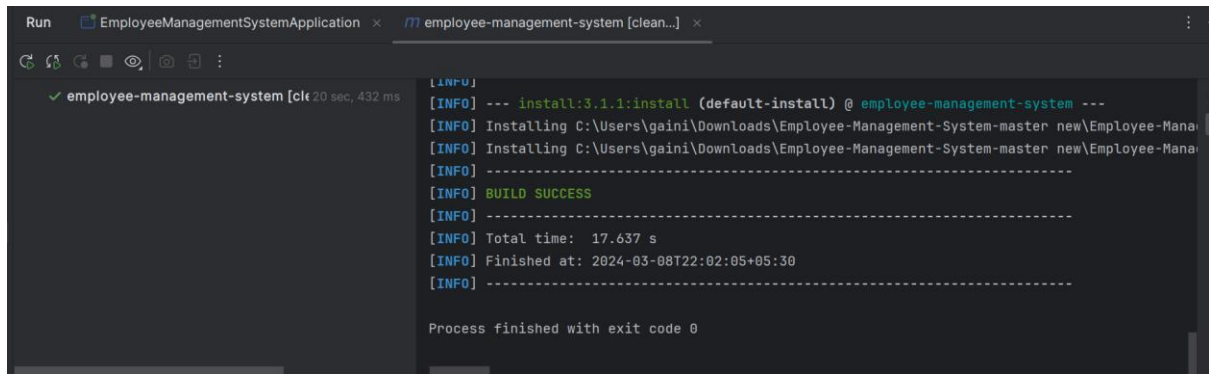


# Employee-Management-System Project

Did the maven clean install:



```
Run EmployeeManagementSystemApplication x employee-management-system [clean...] x

✓ employee-management-system [clt 20 sec, 432 ms]

[INFO] --- install:3.1.1:install (default-install) @ employee-management-system ---
[INFO] Installing C:\Users\gaini\Downloads\Employee-Management-System-master\new\Employee-Mana
[INFO] Installing C:\Users\gaini\Downloads\Employee-Management-System-master\new\Employee-Mana
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 17.637 s
[INFO] Finished at: 2024-03-08T13:02:02:05+05:30
[INFO] -----

Process finished with exit code 0
```

Application started successfully:



```
Console x Problems x Debug Shell
Employee-Management-Directory - EmployeeManagementSystemApplication [Spring Boot App] C:\Users\gaini\Downloads\sts-4.14.0.RELEASE\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin

:: Spring Boot :: (v3.2.3)

2024-03-08T13:24:30.328+05:30 INFO 28512 --- [
2024-03-08T13:24:30.336+05:30 INFO 28512 --- [
2024-03-08T13:24:31.583+05:30 INFO 28512 --- [
2024-03-08T13:24:31.609+05:30 INFO 28512 --- [
2024-03-08T13:24:32.259+05:30 INFO 28512 --- [
2024-03-08T13:24:32.279+05:30 INFO 28512 --- [
2024-03-08T13:24:32.279+05:30 INFO 28512 --- [
2024-03-08T13:24:32.385+05:30 INFO 28512 --- [
2024-03-08T13:24:32.385+05:30 INFO 28512 --- [
2024-03-08T13:24:32.657+05:30 INFO 28512 --- [
2024-03-08T13:24:32.778+05:30 INFO 28512 --- [
2024-03-08T13:24:32.829+05:30 INFO 28512 --- [
2024-03-08T13:24:33.215+05:30 INFO 28512 --- [
2024-03-08T13:24:33.264+05:30 INFO 28512 --- [
2024-03-08T13:24:33.837+05:30 INFO 28512 --- [
2024-03-08T13:24:33.841+05:30 INFO 28512 --- [
2024-03-08T13:24:33.918+05:30 WARN 28512 --- [
2024-03-08T13:24:33.920+05:30 WARN 28512 --- [
2024-03-08T13:24:35.255+05:30 INFO 28512 --- [
2024-03-08T13:24:35.335+05:30 INFO 28512 --- [
2024-03-08T13:24:35.499+05:30 WARN 28512 --- [
2024-03-08T13:24:36.000+05:30 INFO 28512 --- [
2024-03-08T13:24:36.013+05:30 INFO 28512 --- [

main] .m.e.EmployeeManagementSystemApplication : Starting EmployeeManagementSystemApplication us
main] .m.e.EmployeeManagementSystemApplication : No active profile set, falling back to 1 default
main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in I
main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 18
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.16
main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationCont
main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization comp
main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name
main] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.4.4.Fir
main] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache disabled
main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup; ignoring JPA class trs
main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection com.mysql.cj.jc
main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
main] org.hibernate.orm.deprecation : HHH90000025: MySQL8Dialect does not need to be
main] org.hibernate.orm.deprecation : HHH90000026: MySQL8Dialect has been deprecated
main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hibe
main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persi
main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default.
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context
main] .m.e.EmployeeManagementSystemApplication : Started EmployeeManagementSystemApplication in
```

## MySQL database and table creation:

The screenshot shows a MySQL database interface with the following SQL commands in the editor:

```
39 • use employee_directory;
40 • show tables;
41 • select * from employee;
```

The interface displays the 'Tables\_in\_employee\_directory' table, which contains one table named 'employee'. The 'Output' section shows the results of the executed commands:

#	Time	Action	Message
72	22:08:18	use employee_directory	0 row(s) affected
73	22:08:25	show tables	1 row(s) returned

## Adding the details with POST command:

The screenshot shows a REST client interface with the following details:

URL: `http://localhost:8080/employee/addEmployee`

Method: **POST**

Body (JSON):

```
1 {
2   "firstName": "shiva",
3   "lastName": "developer",
4   "email": "shivadeveloper@microsoft.com",
5   "phone": 7894561230,
6   "doj": "2011-01-01"
7 }
```

Response (JSON):

```
1 {
2   "id": 2,
3   "firstName": "shiva",
4   "lastName": "developer",
5   "email": "shivadeveloper@microsoft.com",
6   "phone": 7894561230,
7   "doj": "2011-01-01"
8 }
```

HTTP <http://localhost:8080/employee/allEmployee> Save </>

GET <http://localhost:8080/employee/allEmployee> Send

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "firstName": "shiva",
3   "lastName": "developer",
4   "email": "shivadeveloper@microsoft.com",
5   "phone": 7894561230,
6   "doj": "2011-01-01"
7 }
```

Body Cookies Headers (5) Test Results 200 OK 783 ms 294 B Save as example

Pretty Raw Preview Visualize **JSON** Copy Search

```
1 [
2   {
3     "id": 1,
4     "firstName": "shiva",
5     "lastName": "developer",
6     "email": "shivadeveloper@microsoft.com",
7     "phone": 7894561230,
8     "doj": "2011-01-01"
9   }
10 ]
```

## Fetching all the details using GET command:

GET <http://localhost:8080/employee/allEmployee>

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

```
1 {
```

Body Cookies Headers (5) Test Results 200 OK 2.10 s 544 B

Pretty Raw Preview Visualize **JSON** Copy

```
1 [
2   {
3     "id": 3,
4     "firstName": "prakash",
5     "lastName": "developer",
6     "email": "prakashdeveloper@tesla.com",
7     "phone": 7396456123,
8     "doj": "2013-03-03"
9   },
10  {
11    "id": 5,
12    "firstName": "venki",
13    "lastName": "developer",
14    "email": "venkideveloper@tesla.com",
15    "phone": 7387456123,
```

GET http://localhost:8080/employee/allEmployee

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

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

1 {

Body Cookies Headers (5) Test Results 200 OK 2.10 s 544 B

Pretty Raw Preview Visualize JSON

```
12     "firstName": "venki",
13     "lastName": "developer",
14     "email": "venkideveloper@tesla.com",
15     "phone": 7387456123,
16     "doj": "2014-04-04"
17   },
18   {
19     "id": 7,
20     "firstName": "siraj",
21     "lastName": "developer",
22     "email": "sirajdeveloper@tesla.com",
23     "phone": 7387456123,
24     "doj": "2016-06-06"
25   }
26 ]
```

Postbot Runner Start Proxy

Before using Update command the table looks like:

GET http://localhost:8080/employee/allEmployee Send

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

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

```
1 {
2   ... "firstName": "prakash",
3   ... "lastName": "developer",
4   ... "email": "prakashdeveloper@microsoft.com",
5   ... "phone": 7396456123,
6   ... "doj": "2013-03-03"
7 }
8
9
10
```

Body Cookies Headers (5) Test Results 200 OK 1101 ms 298 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 3,
4     "firstName": "prakash",
5     "lastName": "developer",
6     "email": "prakashdeveloper@microsoft.com",
7     "phone": 7396456123,
8     "doj": "2013-03-03"
9   }
10 ]
```

Postbot Runner Start Proxy Cookies Trash

## After using Update command:

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/employee/update/3`. The request body is a JSON object with the following fields:

```
3 {
4   "lastName": "developer",
5   "email": "prakashdeveloper@tesla.com",
6   "phone": 7396456123,
7   "doj": "2013-03-03"
8 }
```

The response is a 200 OK status with a response time of 1674 ms and a body size of 292 B. The response body is a JSON object:

```
1 {
2   "id": 3,
3   "firstName": "prakash",
4   "lastName": "developer",
5   "email": "prakashdeveloper@tesla.com",
6   "phone": 7396456123,
7   "doj": "2013-03-03"
8 }
```

## Before using deleting command the table looks like:

The screenshot shows a SQL query `select * from employee;` executed in a database client. The result is displayed in a table with the following columns: `id`, `doj`, `email`, `first_name`, `last_name`, and `phone`.

id	doj	email	first_name	last_name	phone
3	2013-03-03	prakashdeveloper@tesla.com	prakash	developer	7396456123
5	2014-04-04	venkideveloper@tesla.com	venki	developer	7387456123
6	2015-05-05	saideveloper@tesla.com	sai	developer	7387456123
7	2016-06-06	sirajdeveloper@tesla.com	siraj	developer	7387456123
NULL	NULL	NULL	NULL	NULL	NULL

The interface also shows a tab for the query, labeled "employee 30" and "employee 48".

Before using deleting command the table looks like:

DELETE

http://localhost:8080/employee/employees/6

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

2

"firstName": "siraj",

3

"lastName": "developer",

4

"email": "sirajdeveloper@tesla.com",

5

"phone": 7387456123,

6

"doj": "2016-06-06"

}

BodyCookiesHeaders (5)Test Results

200 OK

83 ms

21

Pretty

Raw

Preview

Visualize

Text

1

Employee with ID 6 has been deleted successfully.

9 • `select * from employee;`

Result Grid					
Filter Rows:					
Edit: Export/Import: Wrap C					
id	doj	email	first_name	last_name	phone
3	2013-03-03	prakashdeveloper@tesla.com	prakash	developer	7396456123
5	2014-04-04	venkideveloper@tesla.com	venki	developer	7387456123
7	2016-06-06	sirajdeveloper@tesla.com	siraj	developer	7387456123
NULL	NULL	NULL	NULL	NULL	NULL

## Registering user details with POST command:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8080/register
- Body:** A JSON object with the following structure:

```
1 {
2   "username": "goka",
3   "password": "987143"
4 }
5
```
- Response:** A 201 Created status with a response body:

```
1 {
2   "id": 3,
3   "username": "goka",
4   "password": "987143"
5 }
```
- Metadata:** 201 Created, 1943 ms, 215 B.

## Validating the user access then only able to view all employee data:

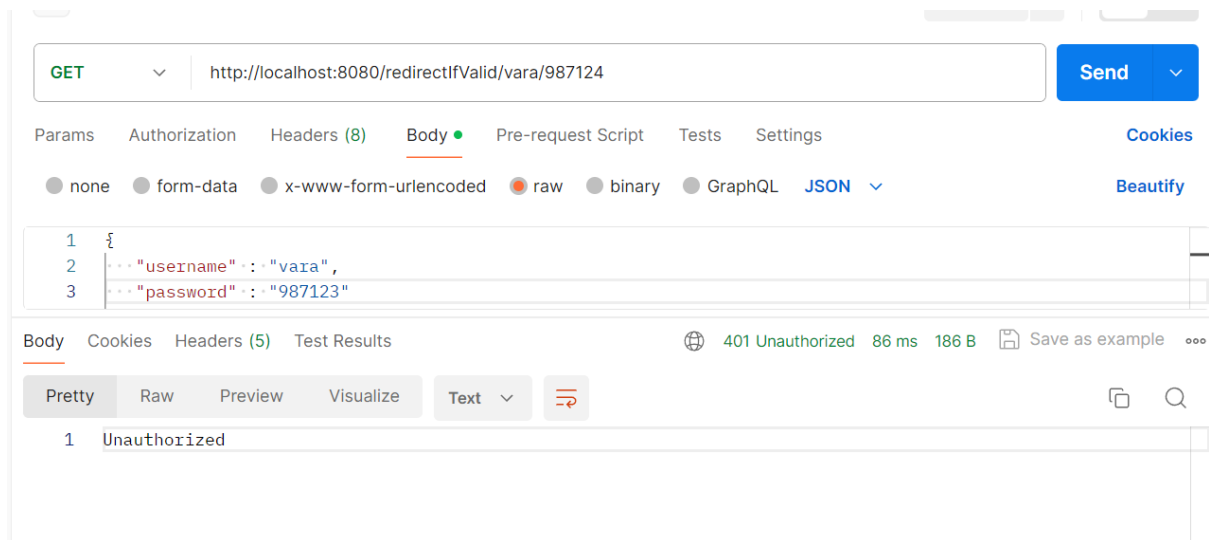
The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8080/redirectIfValid/goka/987143
- Body:** A JSON object with the following structure:

```
1 {
2   "username": "goka",
3   "password": "987143"
4 }
```
- Response:** A 200 OK status with a response body containing employee data:

```
5   "lastName": "developer",
6   "email": "prakashdeveloper@tesla.com",
7   "phone": 7396456123,
8   "doj": "2013-03-03"
9 },
10  {
11   "id": 5,
12   "firstName": "venki",
13   "lastName": "developer",
14   "email": "venki@tesla.com",
15   "phone": 7396456123,
16   "doj": "2013-03-03"
17 }
```
- Metadata:** 200 OK, 456 ms, 670 B.

If incorrect user details given then unauthorized message will appear:



Database view in table after adding user details to it:

