```
//Ques1: Create an Employee Entity which contains following fields
//Name
//Id
//Age
//Location

package com.JPApart1.Exercise1.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Employee {
  private String name;

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private int id;
  private int age;
  private String location;

  public Employee() {    }
public String getName() {
  return name;
}

public void setName(String name) {
  this.name = name;
}

public int getId() {
  return id;
}

public void setId(int id) {
  this.id = id;
}

public int getAge() {
  return age;
}

public void setAge(int age) {
  this.age = age;
}

public String getLocation() {
  return location;
  }

  public void setLocation(String location) {
    this.location = location;
  }
}
```

//Ques 2: Set up EmployeeRepository with Spring Data JPA

```java
package com.JPApart1.Exercise1.repository;



import com.JPApart1.Exercise1.entity.Employee;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;



@Repository

public interface EmployeeRepository extends CrudRepository<Employee,Integer> {



}
```

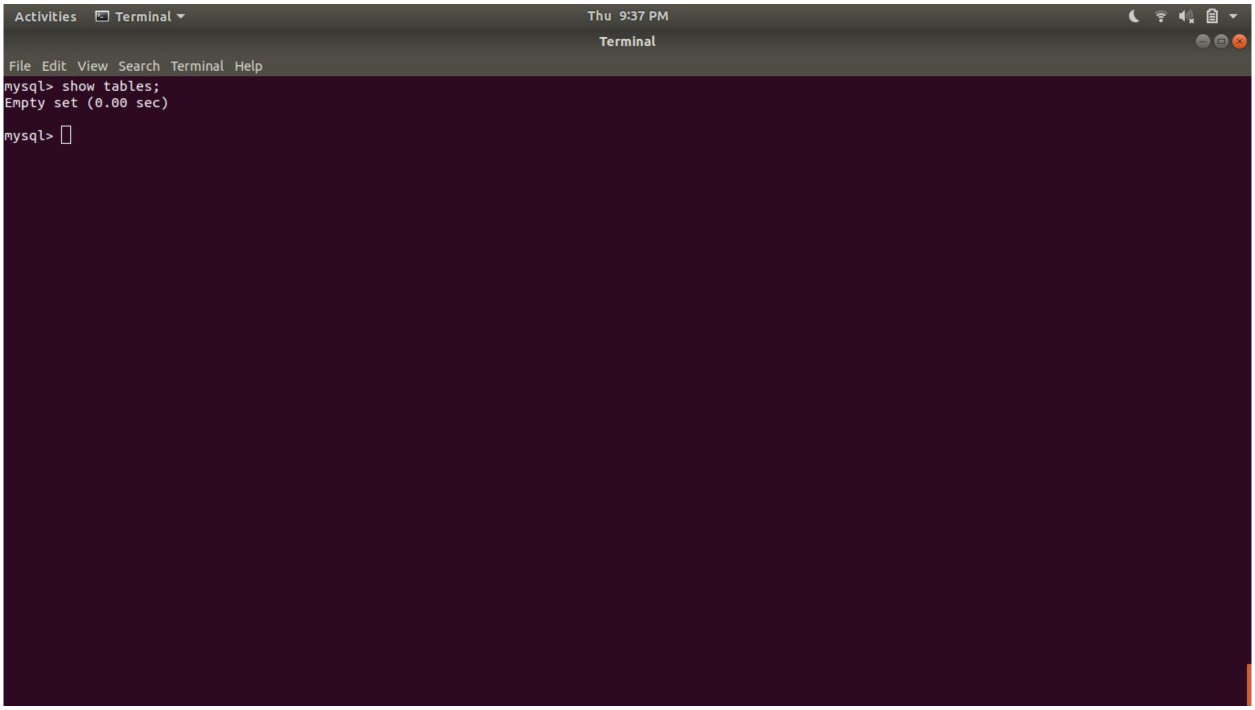
//ques3: Perform Create Operation on Entity using Spring Data JPA

EmployeeService.java

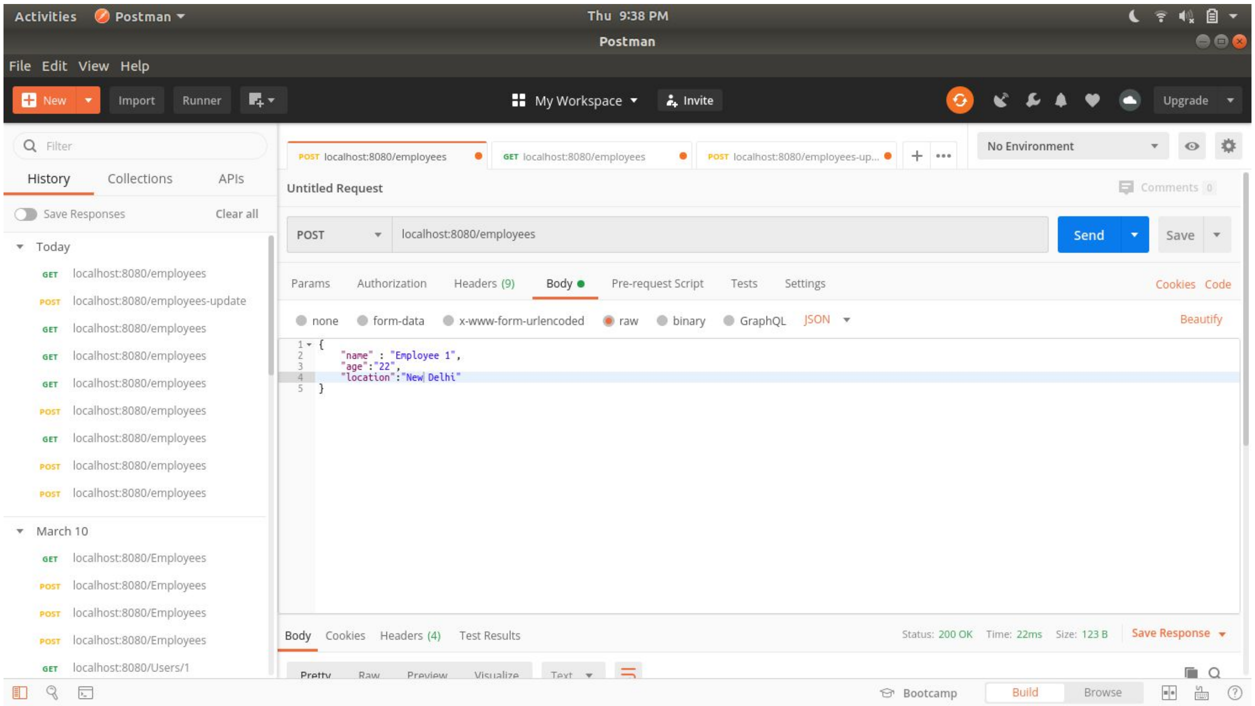
```
@Autowired
EmployeeRepository employeeRepository;

public void addEmployee(Employee employee){
  employeeRepository.save(employee);
}
```

EmployeeController.java

```
@RestController
public class EmployeeController {

  @Autowired
  EmployeeService employeeService;

  @PostMapping(path = "/employees")
  public void addEmployee(@RequestBody Employee employee){
    employeeService.addEmployee(employee);
  }
```



```
mysql> show tables;
Empty set (0.00 sec)

mysql> 
```

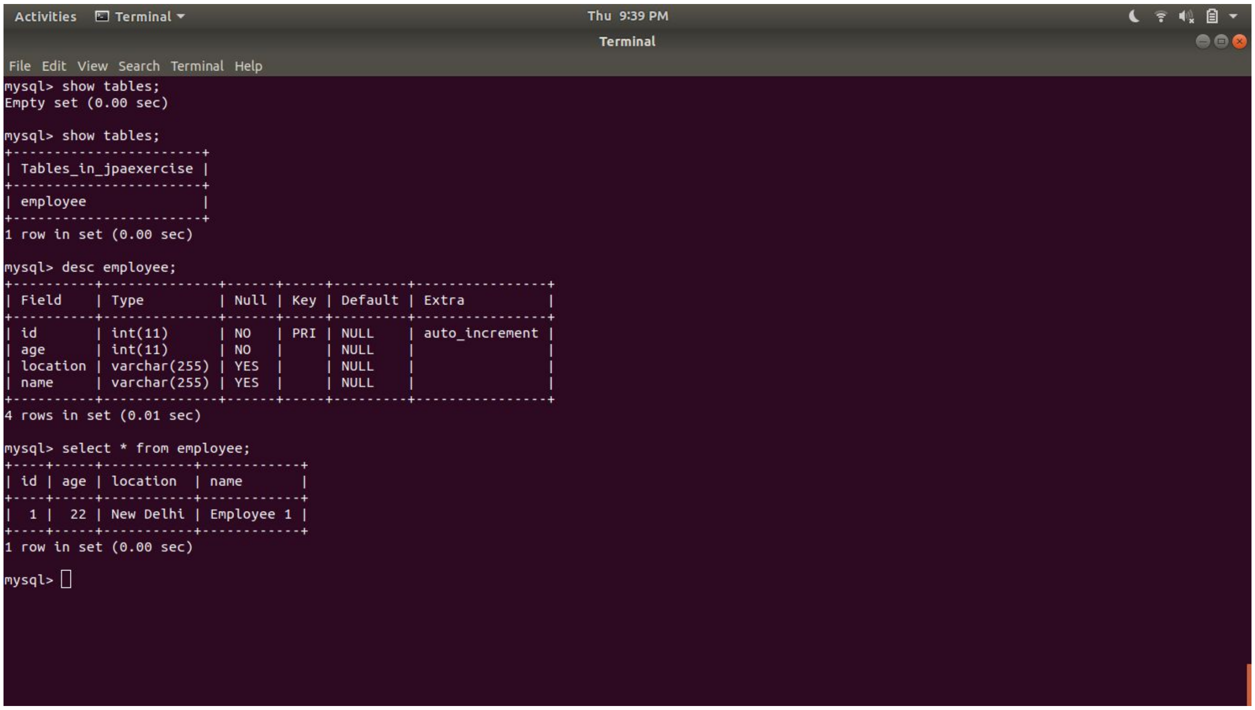

Activities
Postman
Thu 9:38 PM
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
History
Collections
APIs
Save Responses
Clear all
Today
GET localhost:8080/employees
POST localhost:8080/employees-update
GET localhost:8080/employees
GET localhost:8080/employees
GET localhost:8080/employees
POST localhost:8080/employees
GET localhost:8080/employees
POST localhost:8080/employees
POST localhost:8080/employees
March 10
GET localhost:8080/Employees
POST localhost:8080/Employees
POST localhost:8080/Employees
POST localhost:8080/Employees
GET localhost:8080/Users/1
No Environment
Untitled Request
Comments 0
Send
Save
Params
Authorization
Headers (9)
Pre-request Script
Tests
Settings
Cookies
Code
Beautify
Bootcamp
Build
Browse

POST localhost:8080/employees

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "name" : "Employee 1",
    "age":"22",
    "location":"New Delhi"
}
```

Body Cookies Headers (4) Test Results — Status: 200 OK Time: 22ms Size: 123 B — Save Response



```
mysql> show tables;
Empty set (0.00 sec)

mysql> show tables;
+----------------------+
| Tables_in_jpaexercise |
+----------------------+
| employee             |
+----------------------+
1 row in set (0.00 sec)

mysql> desc employee;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| age      | int(11)      | NO   |     | NULL    |                |
| location | varchar(255) | YES  |     | NULL    |                |
| name     | varchar(255) | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)

mysql> select * from employee;
+----+-----+-----------+------------+
| id | age | location  | name       |
+----+-----+-----------+------------+
|  1 |  22 | New Delhi | Employee 1 |
+----+-----+-----------+------------+
1 row in set (0.00 sec)

mysql> 
```

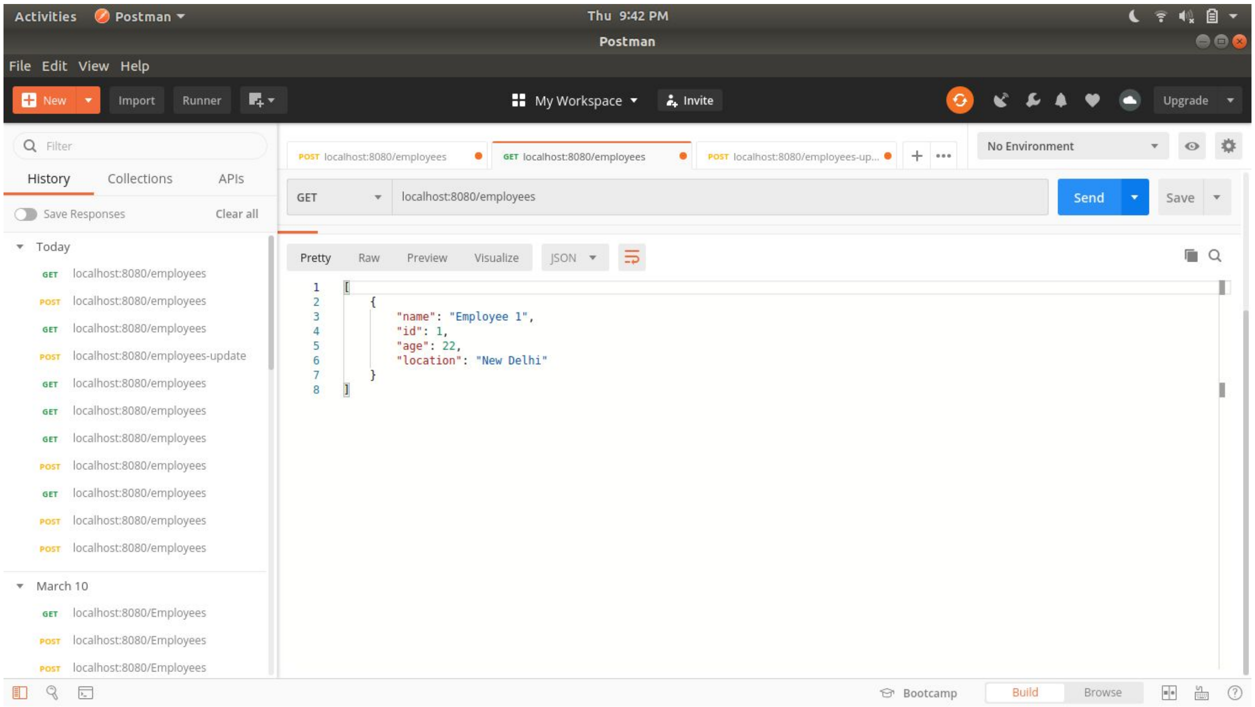
//Ques 4: Perform Update Operation on Entity using Spring Data JPA

EmployeeController.java

```java
@PostMapping(path = "/employees-update")
public void updateEmployee(@RequestBody Employee employee){
  employeeService.updateEmployee(employee);
}
```

EmployeeService.java

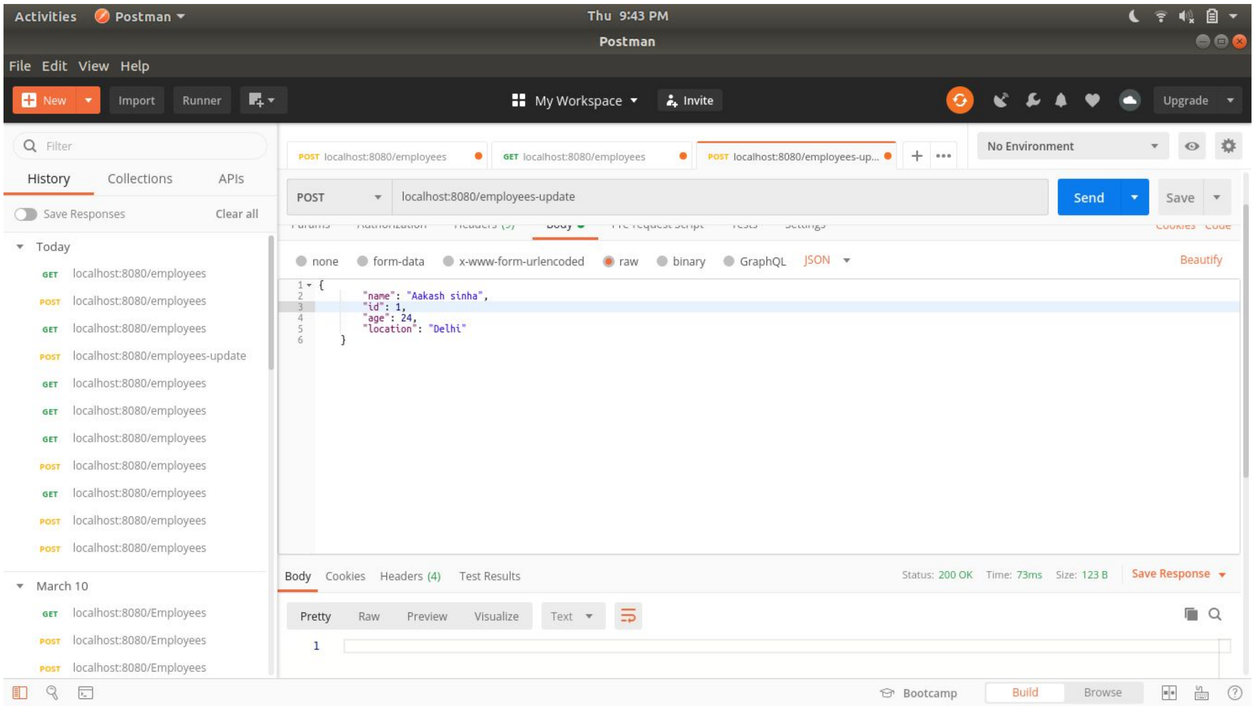
```java
public void updateEmployee(Employee employee){
  employeeRepository.save(employee);
}
```



GET localhost:8080/employees

```
[
    {
        "name": "Employee 1",
        "id": 1,
        "age": 22,
        "location": "New Delhi"
    }
]
```

Activities
Postman
Thu 9:42 PM
Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
POST localhost:8080/employees
GET localhost:8080/employees
POST localhost:8080/employees-up...
No Environment
Send
Save
Filter
History
Collections
APIs
Save Responses
Clear all
Today
GET localhost:8080/employees
POST localhost:8080/employees
GET localhost:8080/employees
POST localhost:8080/employees-update
GET localhost:8080/employees
GET localhost:8080/employees
GET localhost:8080/employees
POST localhost:8080/employees
GET localhost:8080/employees
POST localhost:8080/employees
POST localhost:8080/employees
March 10
GET localhost:8080/Employees
POST localhost:8080/Employees
POST localhost:8080/Employees
Pretty
Raw
Preview
Visualize
JSON
Bootcamp
Build
Browse



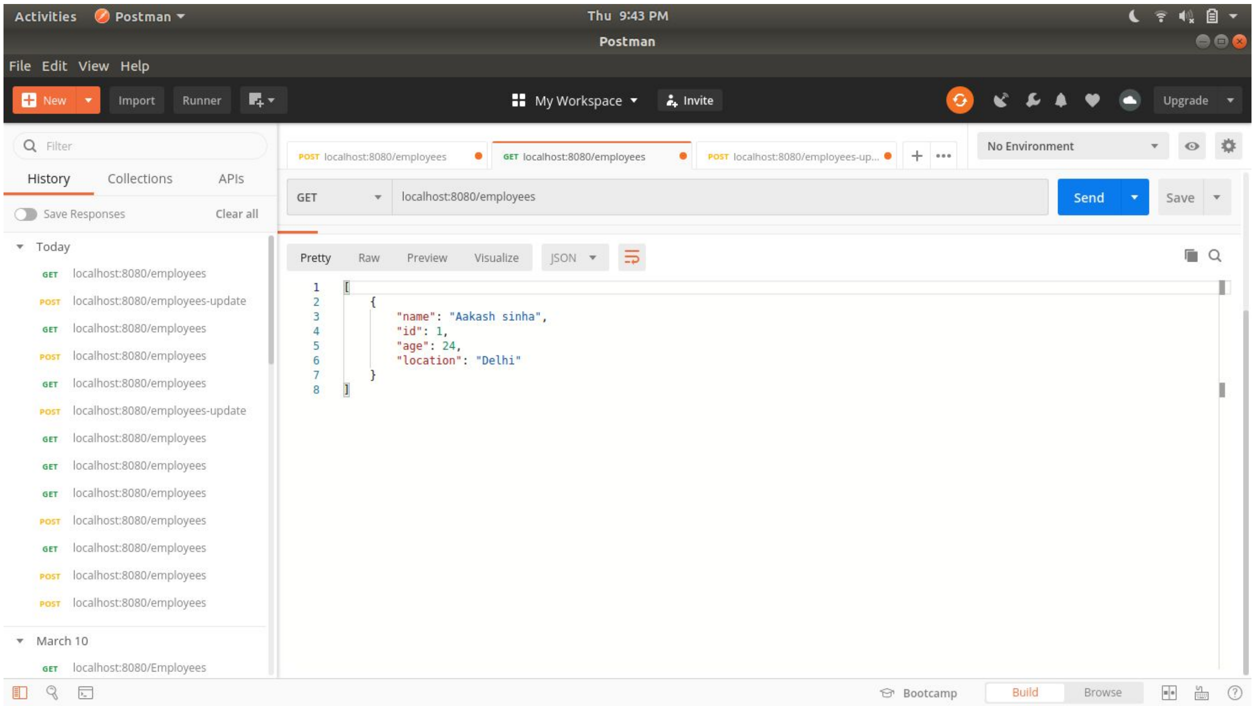
POST localhost:8080/employees-update

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

```
{
    "name": "Aakash sinha",
    "id": 1,
    "age": 24,
    "location": "Delhi"
  }
```

Status: 200 OK Time: 73ms Size: 123 B

Headers (4)



GET localhost:8080/employees

```
[
    {
        "name": "Aakash sinha",
        "id": 1,
        "age": 24,
        "location": "Delhi"
    }
]
```

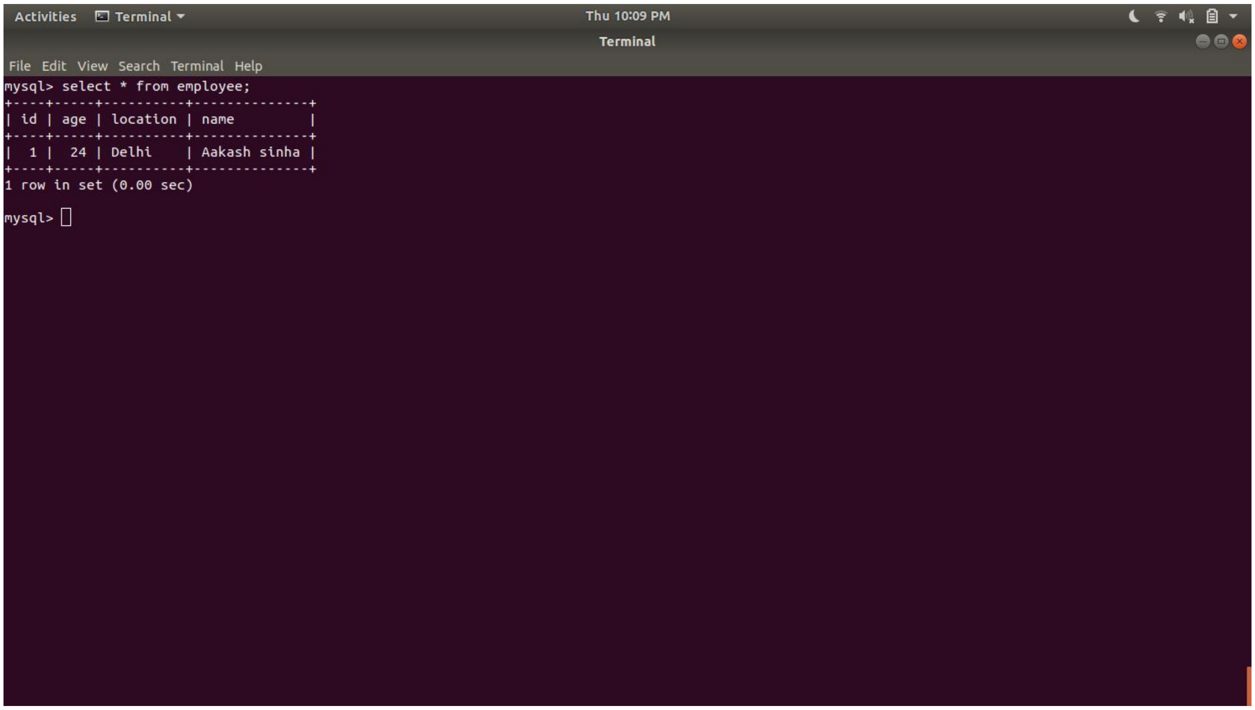
//Ques 5: Perform Delete Operation on Entity using Spring Data JPA

EmployeeController.java

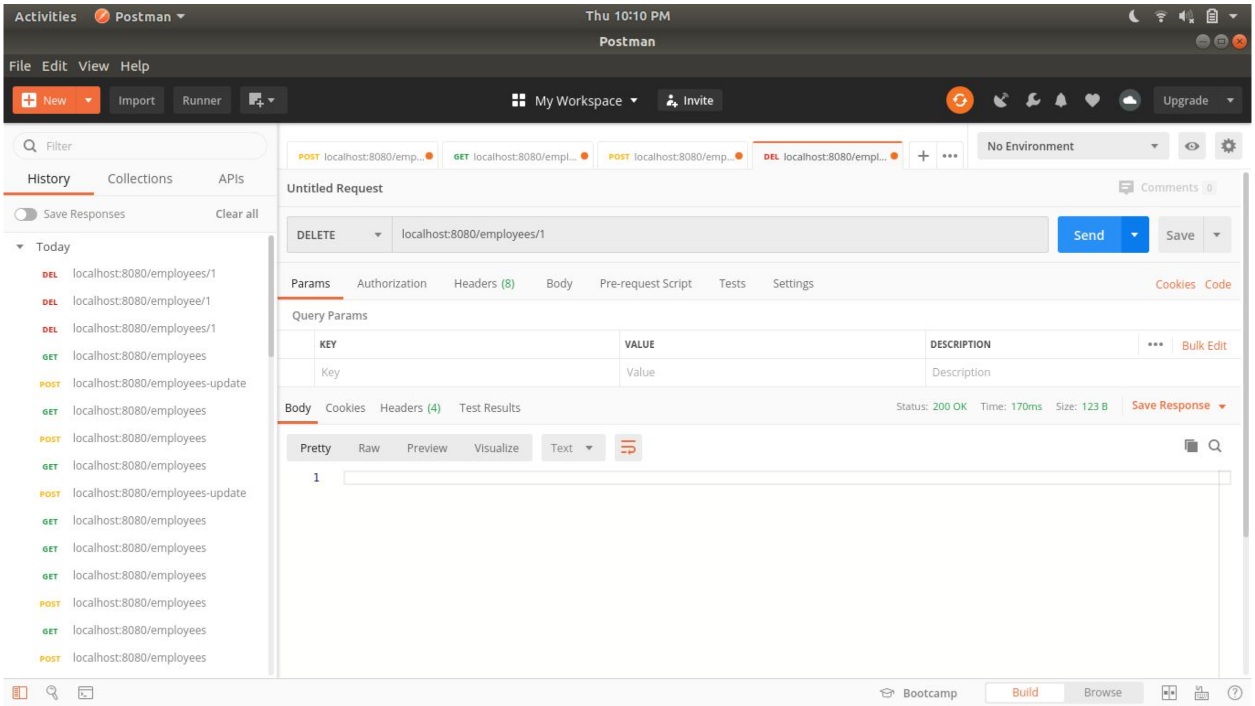
```
@DeleteMapping(path = "/employee/{id}")
public void deleteEmployee(@PathVariable Integer id){
  employeeService.deleteEmployee(id);
}
```

EmployeeService.java

```
public void deleteEmployee(Integer id){
  employeeRepository.deleteById(id);
}
```



```
mysql> select * from employee;
+----+-----+----------+--------------+
| id | age | location | name         |
+----+-----+----------+--------------+
|  1 |  24 | Delhi    | Aakash sinha |
+----+-----+----------+--------------+
1 row in set (0.00 sec)

mysql> 
```

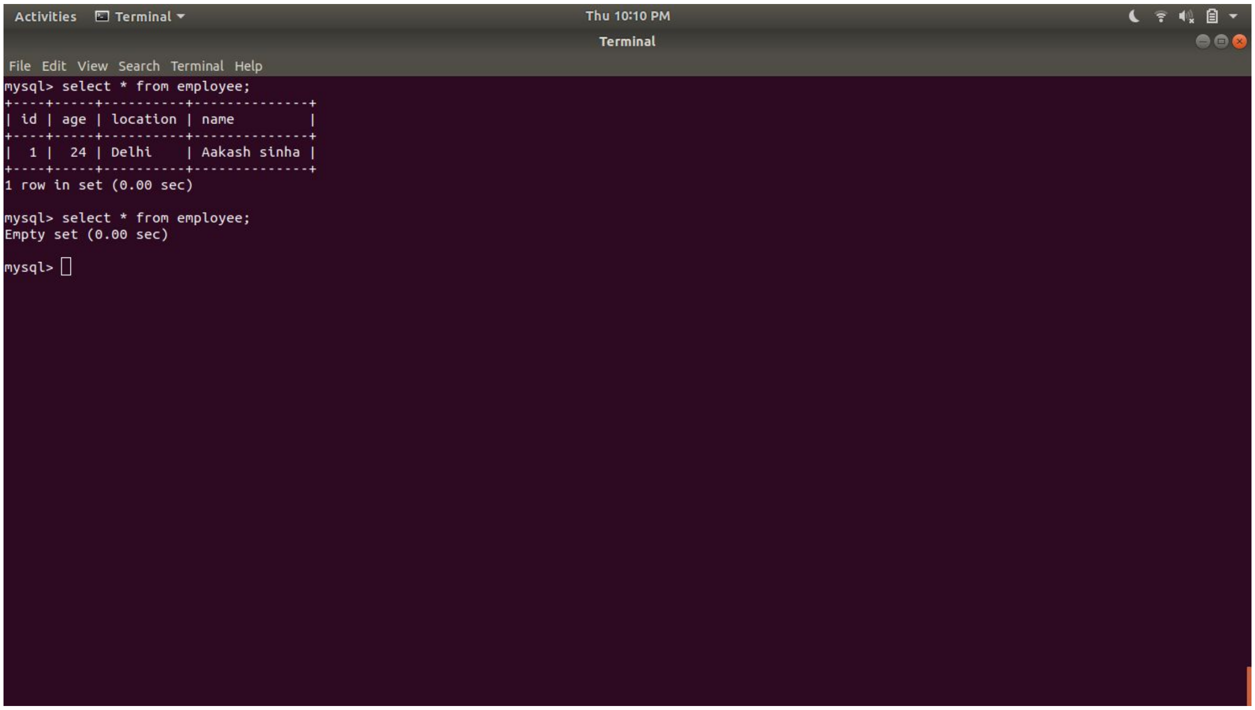


```
DEL   localhost:8080/employees/1
DEL   localhost:8080/employee/1
DEL   localhost:8080/employees/1
GET   localhost:8080/employees
POST  localhost:8080/employees-update
GET   localhost:8080/employees
POST  localhost:8080/employees
GET   localhost:8080/employees
POST  localhost:8080/employees-update
GET   localhost:8080/employees
GET   localhost:8080/employees
GET   localhost:8080/employees
POST  localhost:8080/employees
GET   localhost:8080/employees
POST  localhost:8080/employees
```

Untitled Request

DELETE localhost:8080/employees/1

Status: 200 OK Time: 170ms Size: 123 B



```
mysql> select * from employee;
+----+-----+----------+-------------+
| id | age | location | name        |
+----+-----+----------+-------------+
|  1 |  24 | Delhi    | Aakash sinha |
+----+-----+----------+-------------+
1 row in set (0.00 sec)

mysql> select * from employee;
Empty set (0.00 sec)

mysql> 
```

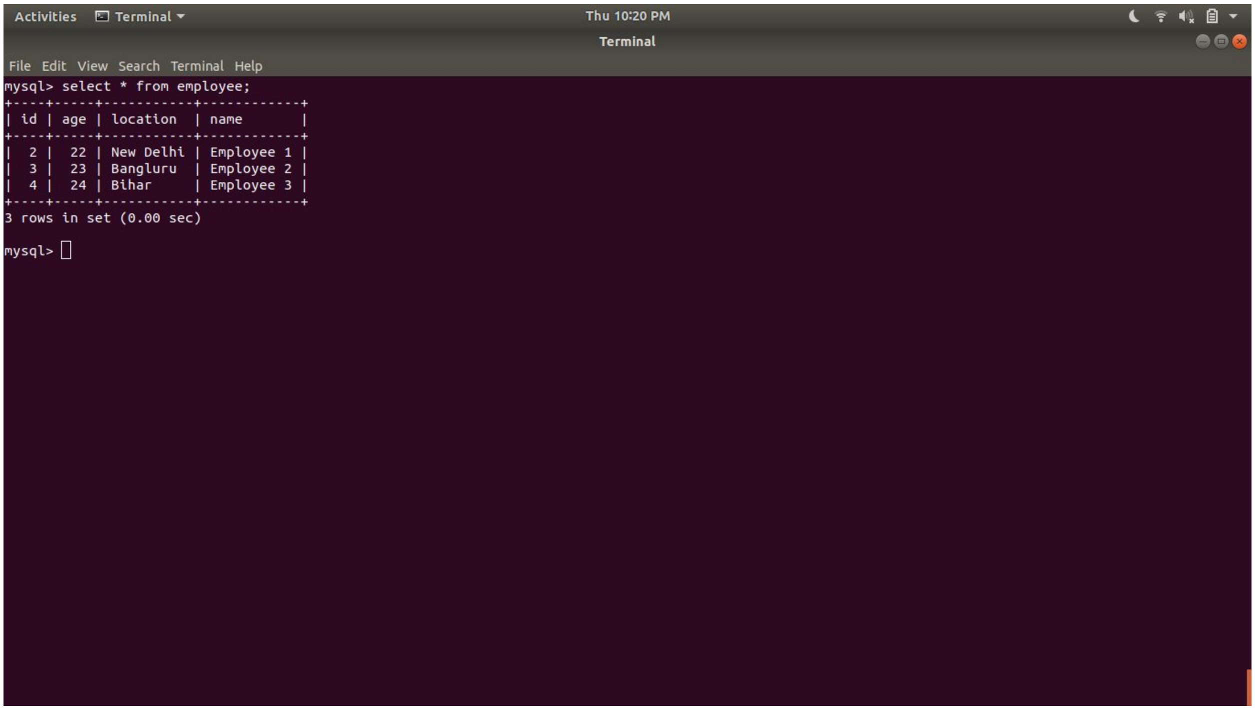
//Ques6: Perform Read Operation on Entity using Spring Data JPA

EmployeeController.java

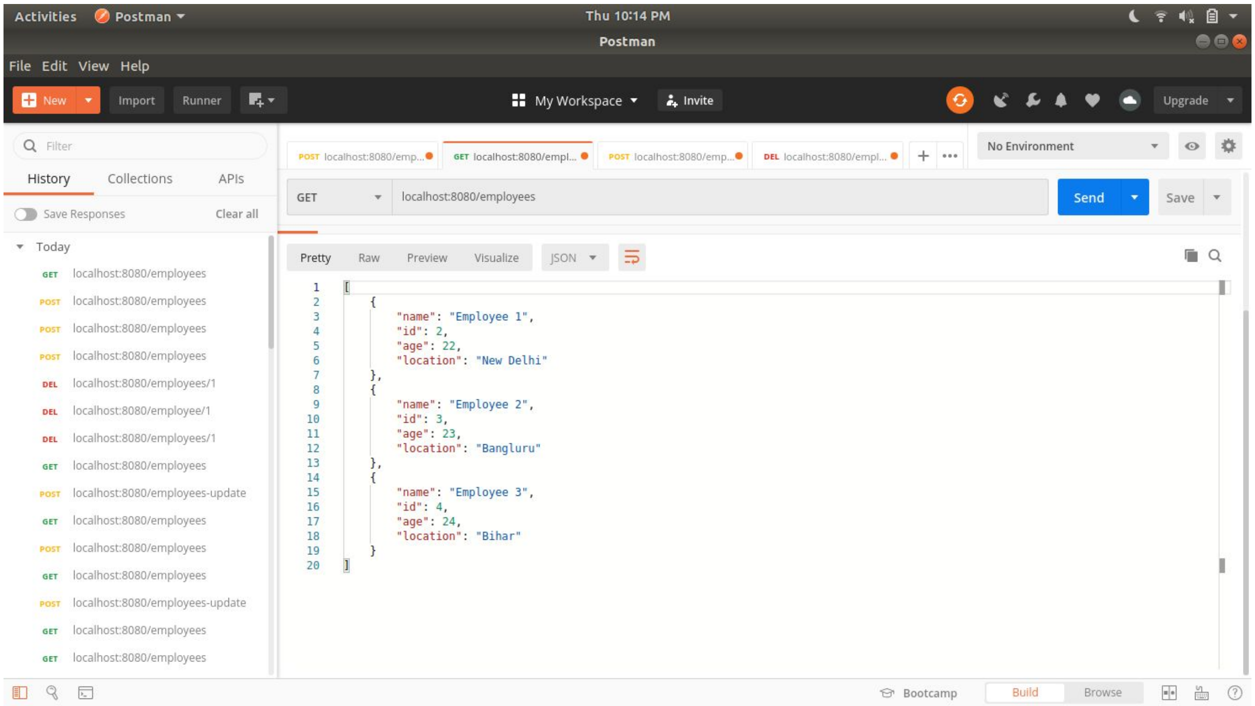
```
@GetMapping(path = "/employees")
public List<Employee> showAllEmployees(){
  List<Employee>tempList = employeeService.getAllEmployee();
  return tempList;
}
```

EmployeeService.java

```
public List<Employee> getAllEmployee(){
  List<Employee> tempList = new ArrayList<>();
  employeeRepository.findAll().forEach(e->tempList.add(e));
  return tempList;
}
```



```
mysql> select * from employee;
+----+-----+-----------+------------+
| id | age | location  | name       |
+----+-----+-----------+------------+
|  2 |  22 | New Delhi | Employee 1 |
|  3 |  23 | Bangluru  | Employee 2 |
|  4 |  24 | Bihar     | Employee 3 |
+----+-----+-----------+------------+
3 rows in set (0.00 sec)

mysql> 
```


Activities
Postman
Thu 10:14 PM
Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
APIs
Save Responses
Clear all
Today
GET localhost:8080/employees
POST localhost:8080/employees
POST localhost:8080/employees
POST localhost:8080/employees
DEL localhost:8080/employees/1
DEL localhost:8080/employee/1
DEL localhost:8080/employees/1
GET localhost:8080/employees
POST localhost:8080/employees-update
GET localhost:8080/employees
POST localhost:8080/employees
GET localhost:8080/employees
POST localhost:8080/employees-update
GET localhost:8080/employees
GET localhost:8080/employees
POST localhost:8080/emp...
GET localhost:8080/empl...
POST localhost:8080/emp...
DEL localhost:8080/empl...
No Environment
Send
Save
Pretty
Raw
Preview
Visualize
JSON
Bootcamp
Build
Browse

GET localhost:8080/employees

```
[
    {
        "name": "Employee 1",
        "id": 2,
        "age": 22,
        "location": "New Delhi"
    },
    {
        "name": "Employee 2",
        "id": 3,
        "age": 23,
        "location": "Bangluru"
    },
    {
        "name": "Employee 3",
        "id": 4,
        "age": 24,
        "location": "Bihar"
    }
]
```

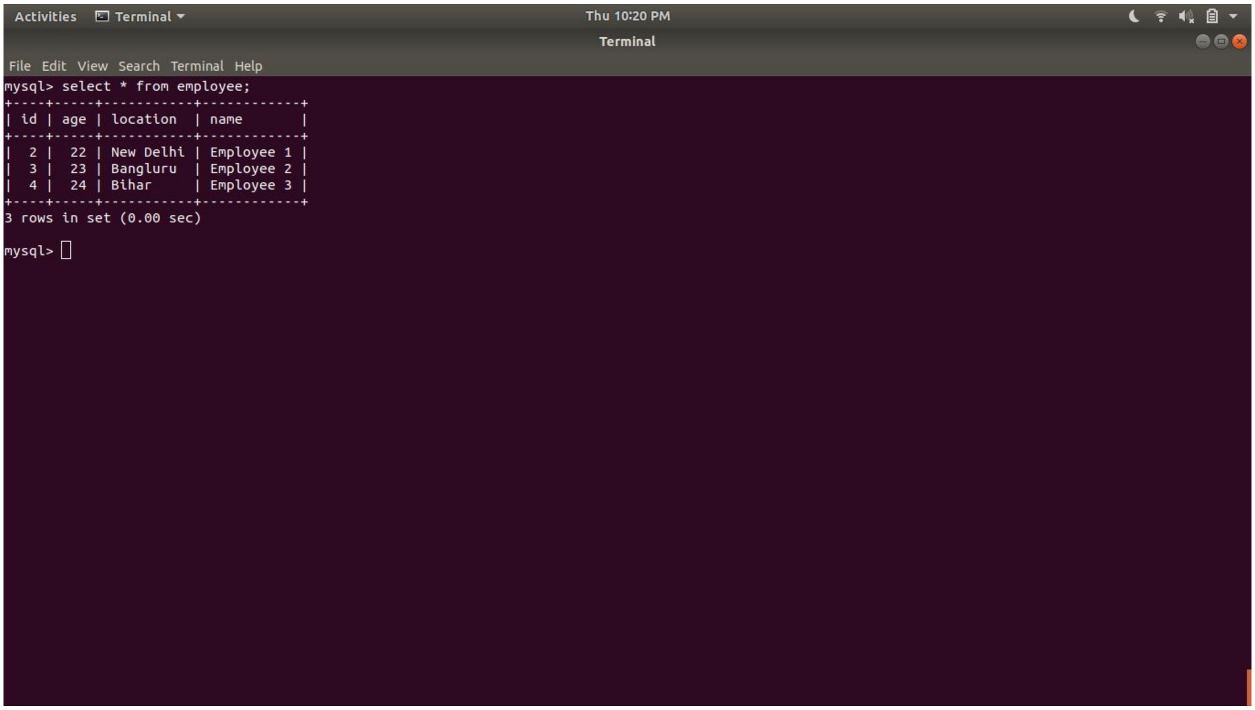
//Ques 7: Get the total count of the number of Employees

EmployeeController.java

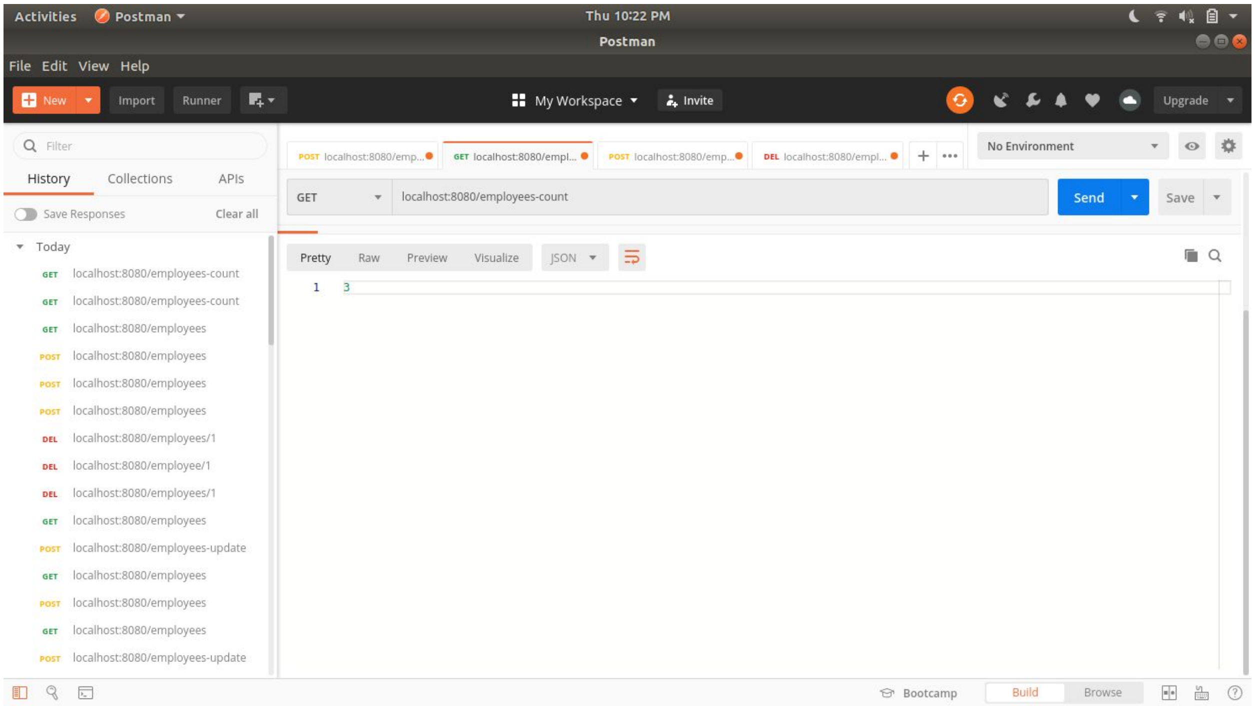
```
@GetMapping(path = "/employees-count")
public Long returnCount(){
  return employeeService.getCount();
}
```

EmployeeService.java

```
public Long getCount(){
  return employeeRepository.count();
}
```



```
mysql> select * from employee;
+----+-----+-----------+------------+
| id | age | location  | name       |
+----+-----+-----------+------------+
|  2 |  22 | New Delhi | Employee 1 |
|  3 |  23 | Bangluru  | Employee 2 |
|  4 |  24 | Bihar     | Employee 3 |
+----+-----+-----------+------------+
3 rows in set (0.00 sec)

mysql> 
```



GET localhost:8080/employees-count

```
3
```

```
GET localhost:8080/employees-count
GET localhost:8080/employees-count
GET localhost:8080/employees
POST localhost:8080/employees
POST localhost:8080/employees
POST localhost:8080/employees
DEL localhost:8080/employees/1
DEL localhost:8080/employee/1
DEL localhost:8080/employees/1
GET localhost:8080/employees
POST localhost:8080/employees-update
GET localhost:8080/employees
POST localhost:8080/employees
GET localhost:8080/employees
POST localhost:8080/employees-update
```

//ques 8: Implement Pagination and Sorting on the bases of Employee Age

EmployeeRepository.java

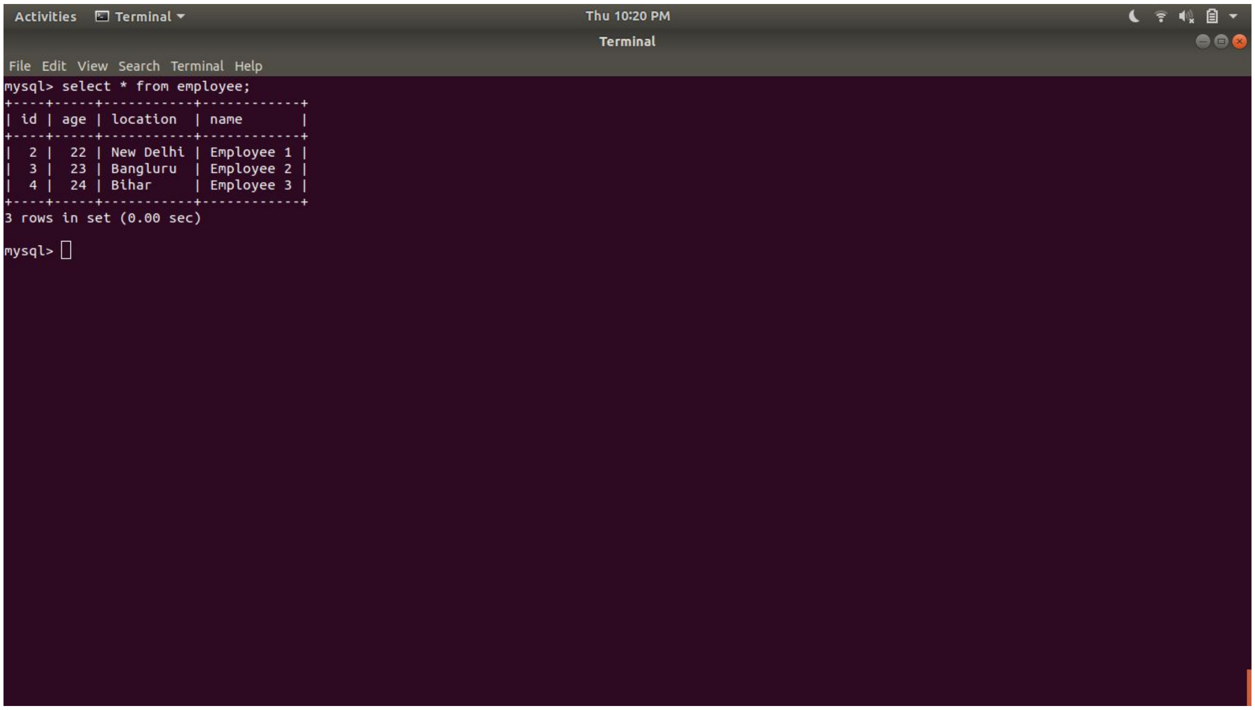
```
@Repository
public interface EmployeeRepository extends PagingAndSortingRepository<Employee,Integer> {

}
```

EmployeeService.java

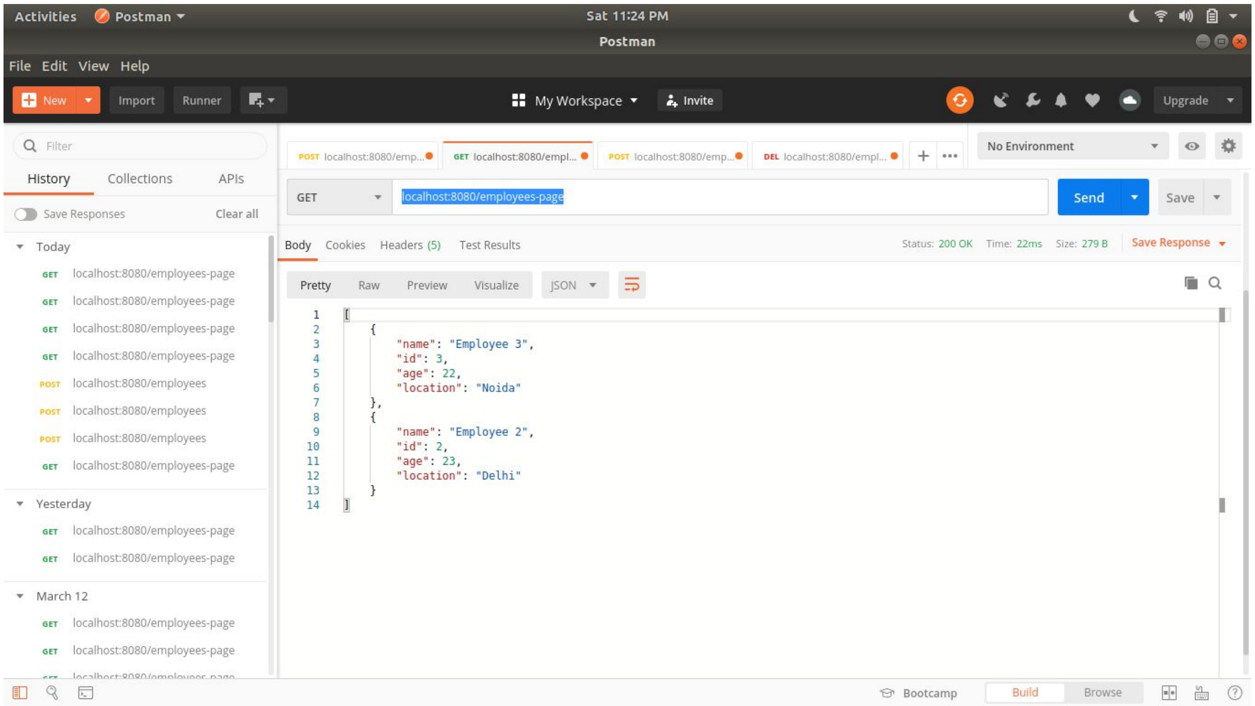
```
public List<Employee> getPage(){
    Pageable pageable = PageRequest.of(0,2, Sort.by("age"));
    List<Employee> employeeList = employeeRepository.findAll(pageable).toList();
    return employeeList;
  }
```

EmplooyeeController.java

```
@GetMapping(path = "/employees-page")
public List<Employee> getPage(){
  List<Employee> tempList= employeeService.getPage();
  return tempList;
}
```



```
mysql> select * from employee;
+----+-----+-----------+------------+
| id | age | location  | name       |
+----+-----+-----------+------------+
|  2 |  22 | New Delhi | Employee 1 |
|  3 |  23 | Bangluru  | Employee 2 |
|  4 |  24 | Bihar     | Employee 3 |
+----+-----+-----------+------------+
3 rows in set (0.00 sec)

mysql> 
```


Activities
Postman
Sat 11:24 PM
Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
APIs
Save Responses
Clear all
Today
GET localhost:8080/employees-page
GET localhost:8080/employees-page
GET localhost:8080/employees-page
GET localhost:8080/employees-page
POST localhost:8080/employees
POST localhost:8080/employees
POST localhost:8080/employees
GET localhost:8080/employees-page
Yesterday
GET localhost:8080/employees-page
GET localhost:8080/employees-page
March 12
GET localhost:8080/employees-page
GET localhost:8080/employees-page
POST localhost:8080/emp...
GET localhost:8080/empl...
POST localhost:8080/emp...
DEL localhost:8080/empl...
No Environment
Send
Save
Body
Cookies
Headers (5)
Test Results
Save Response
Pretty
Raw
Preview
Visualize
JSON
Bootcamp
Build
Browse

GET localhost:8080/employees-page

Status: 200 OK Time: 22ms Size: 279 B

```
[
    {
        "name": "Employee 3",
        "id": 3,
        "age": 22,
        "location": "Noida"
    },
    {
        "name": "Employee 2",
        "id": 2,
        "age": 23,
        "location": "Delhi"
    }
]
```

//Ques 9: Create and use finder to find Employee by Name

EmployeeRepository.java

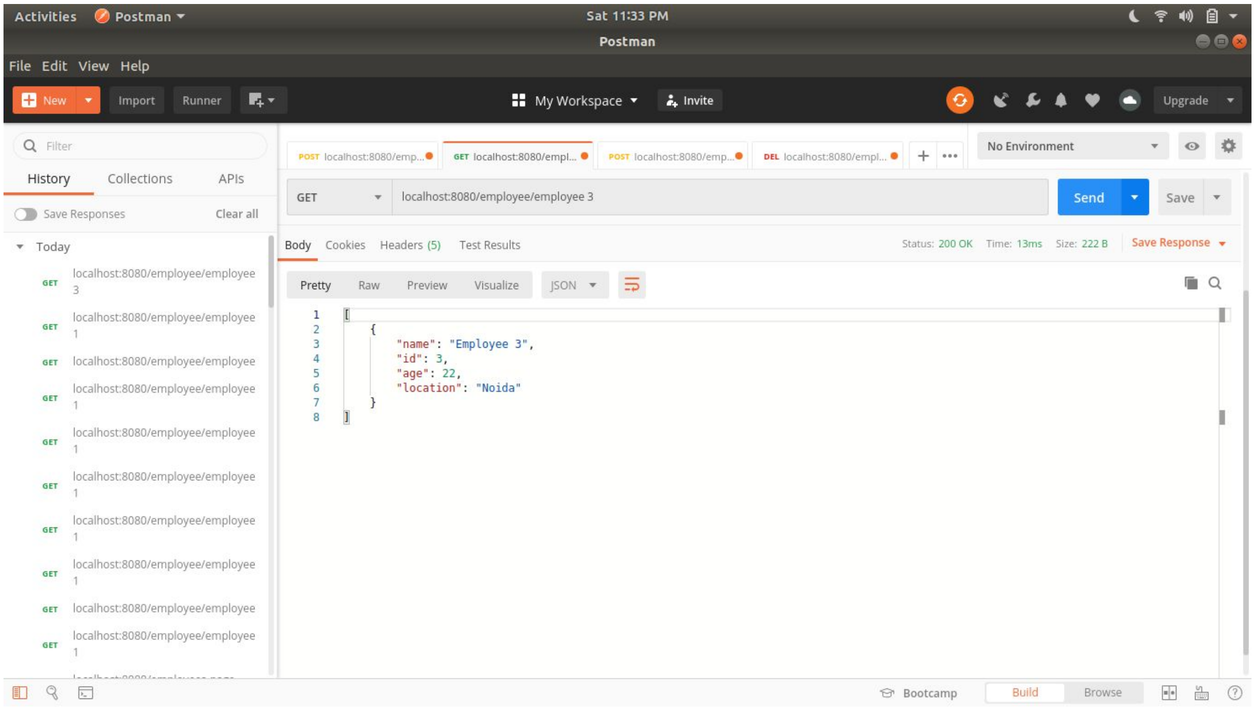
```
List<Employee> findAllByName(String name);
```

EmployeeService.java

```
public List<Employee> findByName(String name){
  return employeeRepository.findAllByName(name);
}
```

EmployeeController.java

```
@GetMapping(path = "/employee/{name}")
public List<Employee> getEployeeByName(@PathVariable String name){
  return employeeService.findByName(name);
}
```



GET localhost:8080/employee/employee 3

Status: 200 OK Time: 13ms Size: 222 B

```
[
    {
        "name": "Employee 3",
        "id": 3,
        "age": 22,
        "location": "Noida"
    }
]
```

GET localhost:8080/employee/employee 3

GET localhost:8080/employee/employee 1

GET localhost:8080/employee/employee

GET localhost:8080/employee/employee 1

GET localhost:8080/employee/employee 1

GET localhost:8080/employee/employee 1

GET localhost:8080/employee/employee 1

GET localhost:8080/employee/employee 1

GET localhost:8080/employee/employee

GET localhost:8080/employee/employee 1

//ques 10: Create and use finder to find Employees starting with A character

EmployeeRepository.java

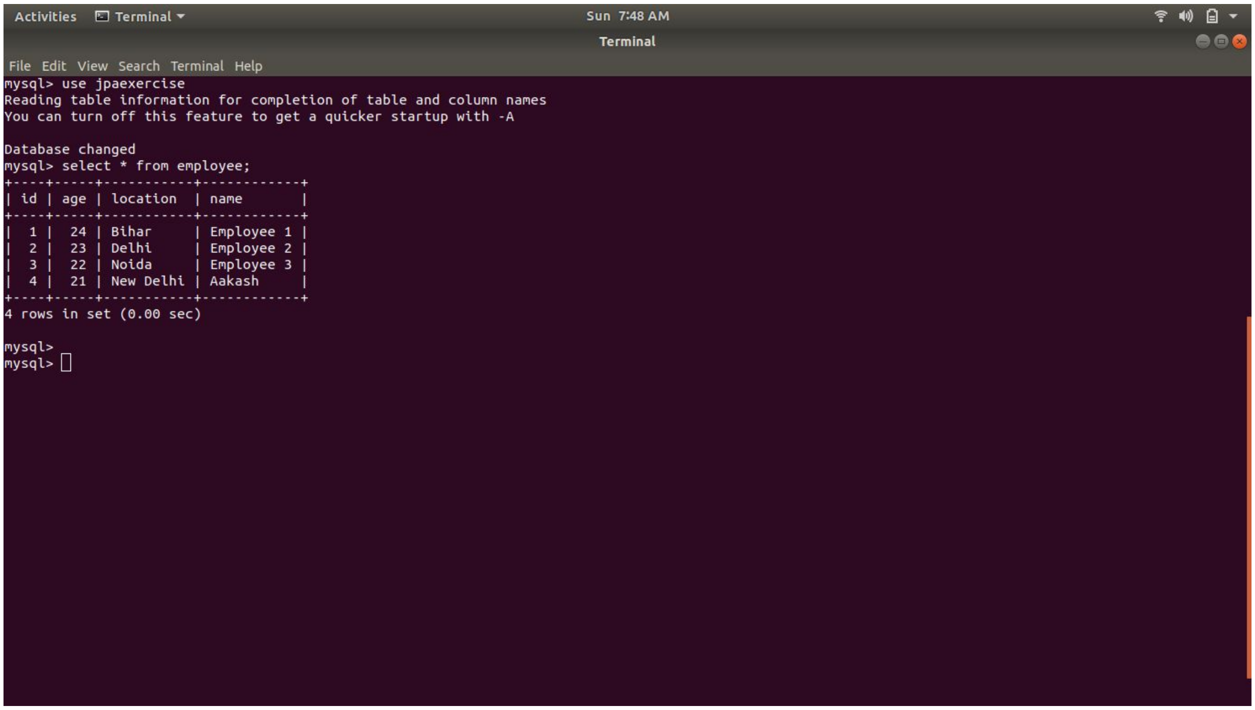
```
List<Employee> findAllByNameStartingWith(Character c);
```

EmployeeService.java

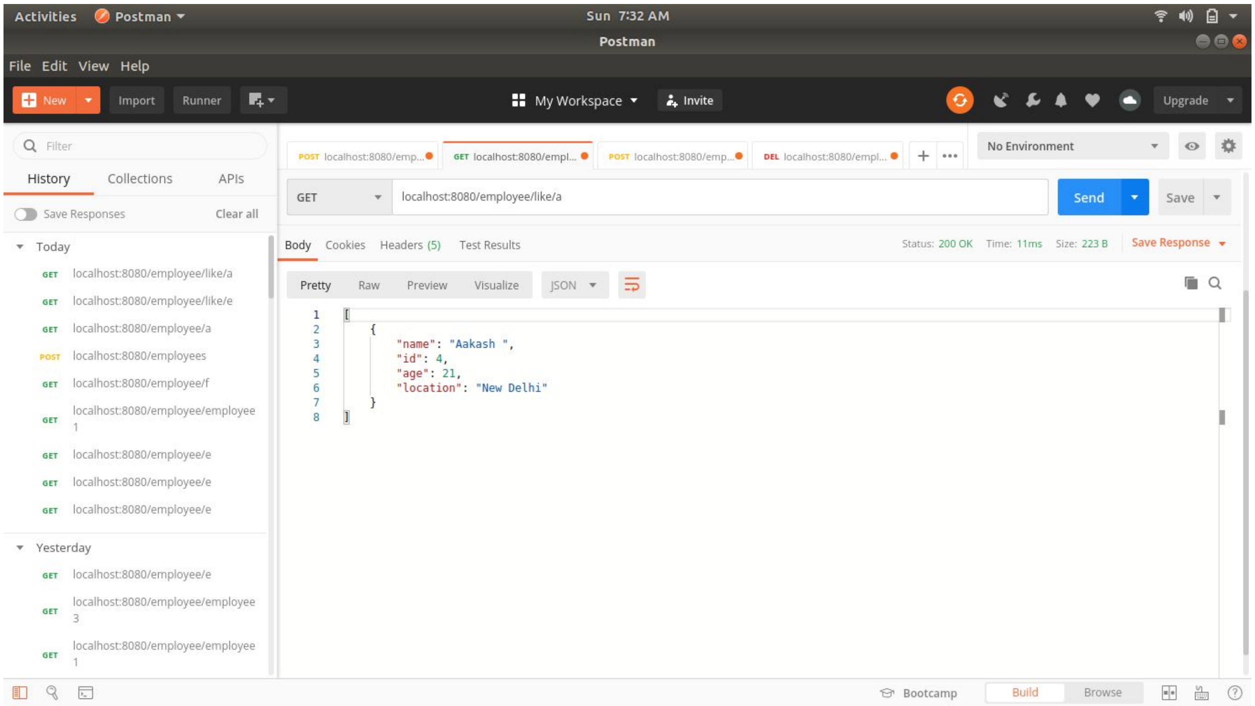
```
public List<Employee> findEmployeeStartWith(Character c){
  return employeeRepository.findAllByNameStartingWith(c);
}
```

EmployeeController.java

```
@GetMapping(path = "/employee/like/{c}")
public List<Employee> getEmployeeByNameLike(@PathVariable Character c){
  return employeeService.findEmployeeStartWith(c);
}
```



```
mysql> use jpaexercise
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from employee;
+----+-----+-----------+------------+
| id | age | location  | name       |
+----+-----+-----------+------------+
|  1 |  24 | Bihar     | Employee 1 |
|  2 |  23 | Delhi     | Employee 2 |
|  3 |  22 | Noida     | Employee 3 |
|  4 |  21 | New Delhi | Aakash     |
+----+-----+-----------+------------+
4 rows in set (0.00 sec)

mysql>
mysql> 
```


Activities
Postman
Sun 7:32 AM
Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
APIs
Save Responses
Clear all
Today
GET localhost:8080/employee/like/a
GET localhost:8080/employee/like/e
GET localhost:8080/employee/a
POST localhost:8080/employees
GET localhost:8080/employee/f
GET localhost:8080/employee/employee1
GET localhost:8080/employee/e
GET localhost:8080/employee/e
GET localhost:8080/employee/e
Yesterday
GET localhost:8080/employee/e
GET localhost:8080/employee/employee3
GET localhost:8080/employee/employee1
POST localhost:8080/emp...
GET localhost:8080/empl...
POST localhost:8080/emp...
DEL localhost:8080/empl...
No Environment
Send
Save
Body
Cookies
Headers (5)
Test Results
Save Response
Pretty
Raw
Preview
Visualize
JSON
Bootcamp
Build
Browse

GET localhost:8080/employee/like/a

Status: 200 OK Time: 11ms Size: 223 B

```
[
    {
        "name": "Aakash ",
        "id": 4,
        "age": 21,
        "location": "New Delhi"
    }
]
```

//ques11: Create and use finder to find Employees Between the age of 28 to 32

EmployeeRepository.java

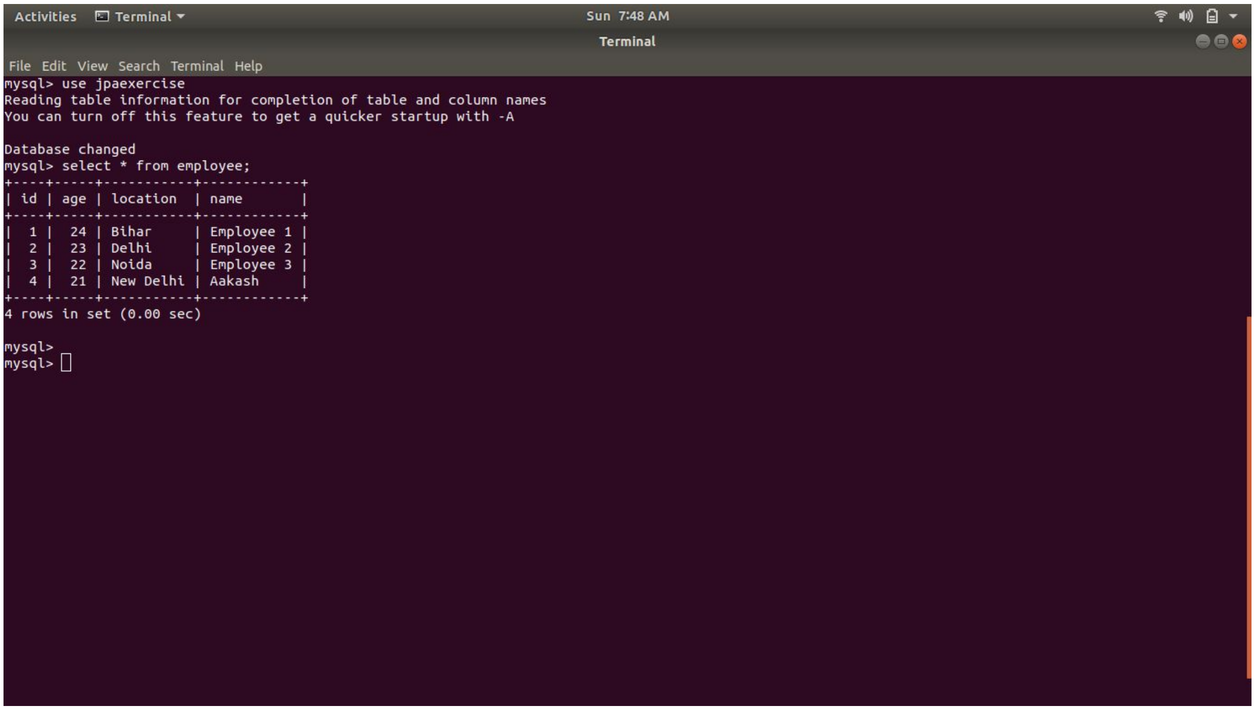
```
List<Employee> findAllByAgeBetween(Integer min, Integer max);
```

EmployeeService.java

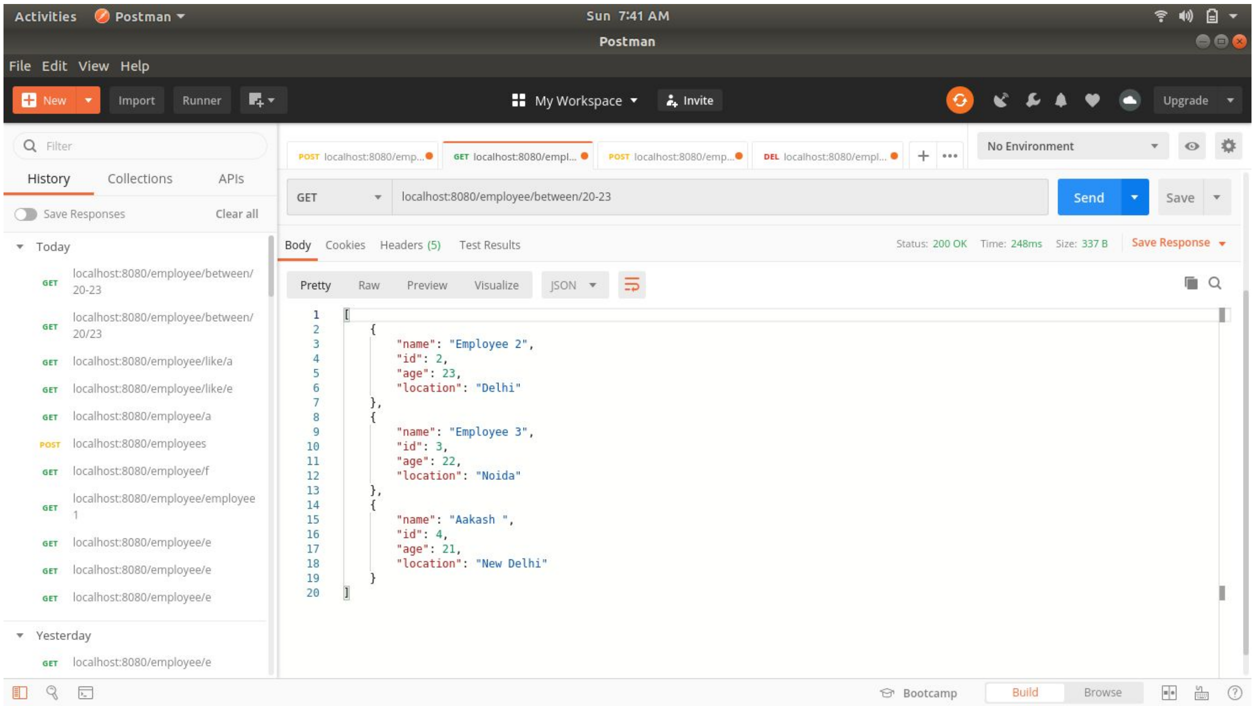
```
public List<Employee> findEmployeeBetweenAge(Integer min, Integer max){
  return employeeRepository.findAllByAgeBetween(min, max);
}
```

EmployeeController.java

```
@GetMapping(path = "/employee/between/{min}-{max}")
public List<Employee> getEmployeeByAge(@PathVariable Integer min, @PathVariable Integer max){
  return employeeService.findEmployeeBetweenAge(min, max);
}
```



```
mysql> use jpaexercise
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from employee;
+----+-----+-----------+------------+
| id | age | location  | name       |
+----+-----+-----------+------------+
|  1 |  24 | Bihar     | Employee 1 |
|  2 |  23 | Delhi     | Employee 2 |
|  3 |  22 | Noida     | Employee 3 |
|  4 |  21 | New Delhi | Aakash     |
+----+-----+-----------+------------+
4 rows in set (0.00 sec)

mysql>
mysql> 
```



Activities
Postman
Sun 7:41 AM
Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
APIs
Save Responses
Clear all
Today
GET localhost:8080/employee/between/20-23
GET localhost:8080/employee/between/20/23
GET localhost:8080/employee/like/a
GET localhost:8080/employee/like/e
GET localhost:8080/employee/a
POST localhost:8080/employees
GET localhost:8080/employee/f
GET localhost:8080/employee/employee 1
GET localhost:8080/employee/e
GET localhost:8080/employee/e
GET localhost:8080/employee/e
Yesterday
GET localhost:8080/employee/e
POST localhost:8080/emp...
GET localhost:8080/empl...
POST localhost:8080/emp...
DEL localhost:8080/empl...
No Environment
Send
Save
Body
Cookies
Headers (5)
Test Results
Save Response
Pretty
Raw
Preview
Visualize
JSON
Bootcamp
Build
Browse

GET localhost:8080/employee/between/20-23

Status: 200 OK Time: 248ms Size: 337 B

```
[
    {
        "name": "Employee 2",
        "id": 2,
        "age": 23,
        "location": "Delhi"
    },
    {
        "name": "Employee 3",
        "id": 3,
        "age": 22,
        "location": "Noida"
    },
    {
        "name": "Aakash ",
        "id": 4,
        "age": 21,
        "location": "New Delhi"
    }
]
```