Session: Spring Data JPA with Hibernate Part 1

Assignment

(1) Create an Employee Entity which contains following fields

Name, Id, Age, Location

```
@Entity
public class Employee {

    @Id
    @Column(name = "empid")
    private int id;
    @Column(name = "empname")
    private String name;
    private int age;
    private String location;

    public Employee() {
    }

    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

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

    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;
    }
}
```

(2) Set up Employee Repository with Spring Data JPA

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Employee.java | application.properties | EmployeeDaoService.java | EmployeeResource.java

```
spring.datasource.url=jdbc:mysql://localhost:3306/newdb
spring.datasource.username=root
spring.datasource.password=root
```

(3) Perform Create Operation on Entity using Spring Data JPA

POST http://localhost:8080/employee

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

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

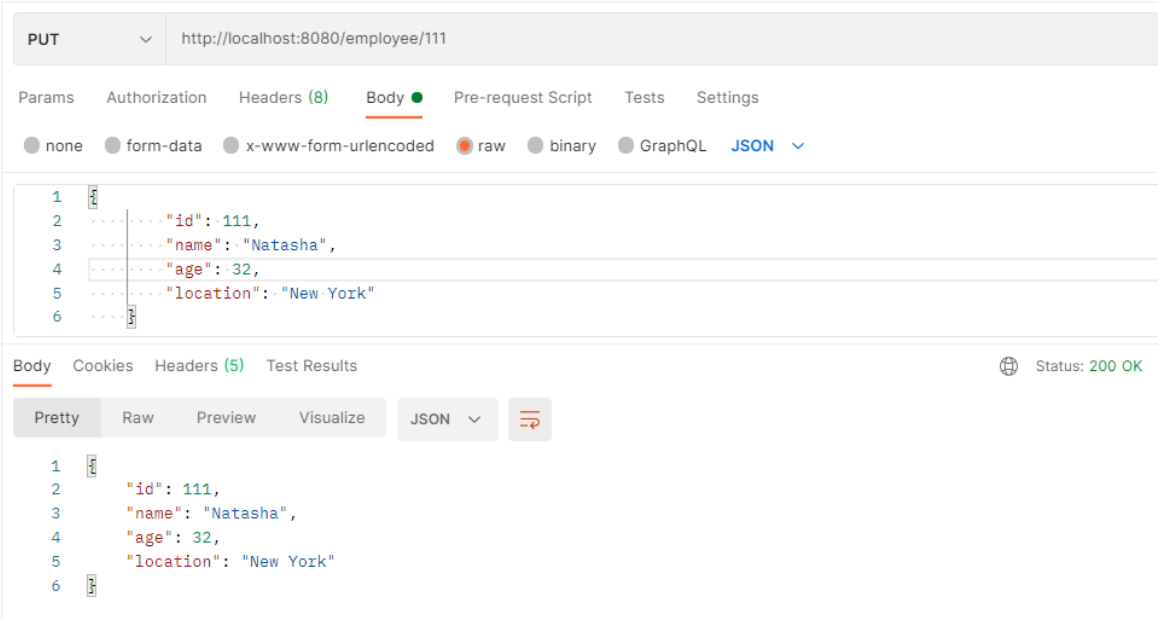
```
{
    "id": 111,
    "name": "Natasha",
    "age": 30,
    "location": "New York"
}
```

Body | Cookies | Headers (5) | Test Results | Status: 200 OK

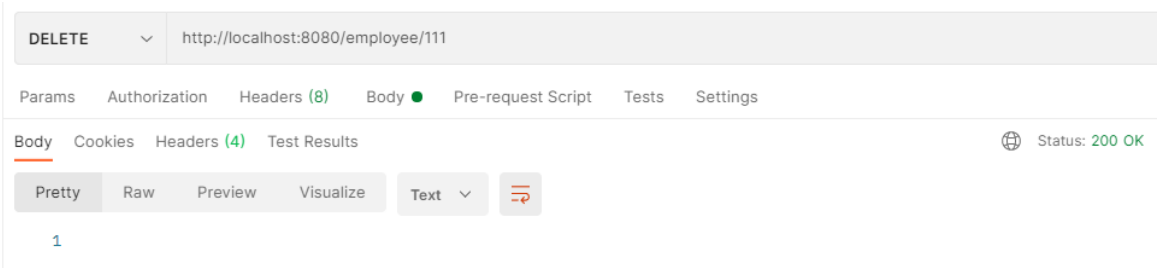
Pretty | Raw | Preview | Visualize | JSON

```
{
    "id": 111,
    "name": "Natasha",
    "age": 30,
    "location": "New York"
}
```

(4) Perform Update Operation on Entity using Spring Data JPA



(5) Perform Delete Operation on Entity using Spring Data JPA



(6) Perform Read Operation on Entity using Spring Data JPA

localhost:8080/employee


```
[
  {
    "id": 101,
    "name": "John",
    "age": 32,
    "location": "New York"
  },
  {
    "id": 102,
    "name": "Alex",
    "age": 31,
    "location": "Dallas"
  },
  {
    "id": 103,
    "name": "Clark",
    "age": 22,
    "location": "Chicago"
  },
  {
    "id": 104,
    "name": "Scott",
    "age": 25,
    "location": "Boston"
  },
  {
    "id": 105,
    "name": "Adams",
    "age": 29,
    "location": "London"
  },
  {
    "id": 106,
    "name": "Justin",
    "age": 30,
    "location": "Paris"
  },
```

(7) Get the total count of the number of Employees

localhost:8080/employee/count

```
10
```

(8) Implement Pagination and Sorting on the bases of Employee Age

localhost:8080/employee/paging?no=0&size=2


```
[
  {
    "id": 103,
    "name": "Clark",
    "age": 22,
    "location": "Chicago"
  },
  {
    "id": 104,
    "name": "Scott",
    "age": 25,
    "location": "Boston"
  }
]
```


localhost:8080/employee/paging?no=1&size=2


```
[
  {
    "id": 110,
    "name": "Johnny",
    "age": 27,
    "location": "Barcelona"
  },
  {
    "id": 105,
    "name": "Adams",
    "age": 29,
    "location": "London"
  }
]
```

(9) Create and use finder to find Employee by Name

localhost:8080/employee/Scott


```
[
    {
        "id": 104,
        "name": "Scott",
        "age": 25,
        "location": "Boston"
    }
]
```


(10) Create and use finder to find Employees starting with A character

localhost:8080/employee/likeA


```
[
    {
        "id": 102,
        "name": "Alex",
        "age": 31,
        "location": "Dallas"
    },
    {
        "id": 105,
        "name": "Adams",
        "age": 29,
        "location": "London"
    },
    {
        "id": 107,
        "name": "Andrew",
        "age": 31,
        "location": "Los Angeles"
    }
]
```

(11) Create and use finder to find Employees Between the age of 28 to 32

localhost:8080/employee/age


```
[
  {
    "id": 101,
    "name": "John",
    "age": 32,
    "location": "New York"
  },
  {
    "id": 102,
    "name": "Alex",
    "age": 31,
    "location": "Dallas"
  },
  {
    "id": 105,
    "name": "Adams",
    "age": 29,
    "location": "London"
  },
  {
    "id": 106,
    "name": "Justin",
    "age": 30,
    "location": "Paris"
  },
  {
    "id": 107,
    "name": "Andrew",
    "age": 31,
    "location": "Los Angeles"
  },
  {
    "id": 108,
    "name": "Smith",
    "age": 30,
    "location": "Madrid"
  }
]
```