

Spring Data JPA part 1 Exercise

(1) Create an Employee Entity which contains following fields

Name

Id

Age

Location

File=Employee.java

```
package com.tothenew.JPA_Demo.employeeEntities;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table //this will only be used when you have different name in database like -@Table(name=employee)
```

```
public class Employee {
```

```
    private String name;
```

```
    @Id
```

```
    private Integer id;
```

```
    private Integer age;
```

```
    //@Column(name = "give attribute original name as in database table column else will give error")
```

```
    private String location;
```

```
public String getName() {
```

```
    return name;
```

```
}
```

```
public void setName(String name) {
```

```
    this.name = name;
```

```
}
```

```
public Integer getId() {
```

```
    return id;
```

```
}
```

```

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

public Integer getAge() {
    return age;
}

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

public String getLocation() {
    return location;
}

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

public Employee(String name, Integer id, Integer age, String location) {
    this.name = name;
    this.id = id;
    this.age = age;
    this.location = location;
}

@Override
public String toString() {
    return "Employee{" +
        "name=" + name + " " +
        ", id=" + id +
        ", age=" + age +
        ", location=" + location + " " +
        '}';
}
}

```

(2) Set up EmployeeRepository with Spring Data JPA

File=EmployeeRepository.java

```
package com.tothenew.JPA_Demo.employeeEntities;

import org.springframework.data.repository.CrudRepository;

public interface EmployeeRepository extends CrudRepository<Employee,Integer> {

}
```

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

```
package com.tothenew.JPA_Demo;

import com.tothenew.JPA_Demo.employeeEntities.Employee;
import com.tothenew.JPA_Demo.employeeEntities.EmployeeRepository;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
```

```
@SpringBootTest
class JpaDemoApplicationTests {

    @Autowired
    EmployeeRepository employeeRepository;

    @Test
    public void contextLoads(){

}
```

```
@Test
public void testCreate(){
    Employee employee=new Employee();
    employee.setName("Gunjan");
    employee.setAge(22);
    employee.setLocation("Delhi");
    employeeRepository.save(employee);
}
}
```

```
mysql> drop table employee;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from employee;
+----+-----+-----+-----+
| id | name  | age  | location |
+----+-----+-----+-----+
| 1  | Gunjan | 22   | Delhi    |
+----+-----+-----+-----+
1 row in set (0.01 sec)

mysql>
```

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

```
package com.tothenew.JPA_Demo;
```

```
import com.tothenew.JPA_Demo.employeeEntities.Employee;
```

```
import com.tothenew.JPA_Demo.employeeEntities.EmployeeRepository;
```

```
import org.junit.jupiter.api.Test;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.boot.test.context.SpringBootTest;
```

```
@SpringBootTest
```

```
class JpaDemoApplicationTests {
```

```
    @Autowired
```

```
    EmployeeRepository employeeRepository;
```

```
    @Test
```

```
    public void contextLoads(){
```

```
    }
```

```
//Ques4 Perform Update Operation on Entity using Spring Data JPA
```

```
    @Test
```

```
    public void testUpdate() {
```

```
        Employee emp = employeeRepository.findById(1).get();
```

```
        emp.setName("Gaurav");
```

```
        employeeRepository.save(emp);
```

```
    }
```

```
}
```

```
3 rows in set (0.00 sec)

mysql> select * from employee;
+----+-----+-----+-----+
| id | name  | age  | location |
+----+-----+-----+-----+
| 1  | Gunjan | 21   | New Delhi |
| 2  | Gunjan | 22   | Delhi    |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

After Updation:

```
2 rows in set (0.00 sec)

mysql> select * from employee;
+----+-----+-----+-----+
| id | name  | age  | location |
+----+-----+-----+-----+
| 1  | Gaurav | 21   | New Delhi |
| 2  | Gunjan | 22   | Delhi    |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

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

```
package com.tothenew.JPA_Demo;
```

```
import com.tothenew.JPA_Demo.employeeEntities.Employee;
```

```
import com.tothenew.JPA_Demo.employeeEntities.EmployeeRepository;
```

```
import org.junit.jupiter.api.Test;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.boot.test.context.SpringBootTest;
```

```
import java.util.Optional;
```

```
@SpringBootTest
```

```
class JpaDemoApplicationTests {
```

```
    @Autowired
```

```
    EmployeeRepository employeeRepository;
```

```
    @Test
```

```
    public void contextLoads(){
```

```
    }
```

//Ques5 Perform Delete Operation on Entity using Spring Data JPA

@Test

```
public void testDelete(){  
    employeeRepository.deleteById(3);  
}
```

```
2 rows in set (0.01 sec)  
  
mysql> select * from employee;  
+----+-----+-----+-----+  
| id | name  | age  | location |  
+----+-----+-----+-----+  
| 1  | Gunjan | 21   | New Delhi |  
| 2  | Gunjan | 22   | Delhi |  
| 3  | Gaurav | 24   | Gurgaon |  
+----+-----+-----+-----+  
3 rows in set (0.00 sec)  
  
mysql> 
```

After Deletion:

```
mysql> select * from employee;  
+----+-----+-----+-----+  
| id | name  | age  | location |  
+----+-----+-----+-----+  
| 1  | Gunjan | 21   | New Delhi |  
| 2  | Gunjan | 22   | Delhi |  
+----+-----+-----+-----+  
2 rows in set (0.01 sec)  
  
mysql> select * from employee;  
+----+-----+-----+-----+  
| id | name  | age  | location |  
+----+-----+-----+-----+  
| 1  | Gunjan | 21   | New Delhi |  
| 2  | Gunjan | 22   | Delhi |  
+----+-----+-----+-----+  
2 rows in set (0.01 sec)  
  
mysql> select * from employee;  
+----+-----+-----+-----+  
| id | name  | age  | location |  
+----+-----+-----+-----+  
| 1  | Gunjan | 21   | New Delhi |  
| 2  | Gunjan | 22   | Delhi |  
+----+-----+-----+-----+  
2 rows in set (0.01 sec)  
  
mysql> select * from employee;  
+----+-----+-----+-----+  
| id | name  | age  | location |  
+----+-----+-----+-----+  
| 1  | Gunjan | 21   | New Delhi |  
| 2  | Gunjan | 22   | Delhi |  
+----+-----+-----+-----+  
2 rows in set (0.00 sec)  
  
mysql> 
```

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

```
package com.tothenew.JPA_Demo;
```

```
import com.tothenew.JPA_Demo.employeeEntities.Employee;  
import com.tothenew.JPA_Demo.employeeEntities.EmployeeRepository;  
import org.junit.jupiter.api.Test;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.context.SpringBootTest;
```

```
import java.util.Optional;
```

```
@SpringBootTest
```

```
class JpaDemoApplicationTests {
```

```
    @Autowired
```

```
    EmployeeRepository employeeRepository;
```

```
    @Test
```

```
    public void contextLoads(){  
    }  
}
```

```
//Ques5 Perform Read Operation on Entity using Spring Data JPA
```

```
@Test
```

```
public void testRead(){  
    Optional<Employee> emp=employeeRepository.findById(1);  
    if(emp.isPresent())  
        System.out.println(emp.get());  
}
```

```
}
```

```
}
```

Activities IntelliJ IDEA Community Edition Fri 07:46

JPA_Demo [-/Documents/JPA_Demo] - .../src/test/java/com/tothenew/JPA_Demo/JpaDemoApplicationTests.java [JPA_Demo.test]

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

JPA_Demo src test java com tothenew JPA_Demo JpaDemoApplicationTests JpaDemoApplicationTests.testRead

JpaDemoApplicationTests.java

```
45
46 //Ques4 Perform Update Operation on Entity using Spring Data JPA
47
48
49
50
```

Run: JpaDemoApplicationTests.testRead

Tests passed: 1 of 1 test - 133 ms

Test 133 ms

```
2020-03-13 07:41:25.280 INFO 8326 --- [Test worker] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring data repository scanning in 23ms. Found 0 database entities with the following query: SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'hibernate'
2020-03-13 07:41:26.075 INFO 8326 --- [Test worker] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2020-03-13 07:41:26.506 INFO 8326 --- [Test worker] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2020-03-13 07:41:26.564 INFO 8326 --- [Test worker] o.hibernate.jpa.internal.util.LogHelper : HH000204: Processing PersistenceUnitInfo [name: default]
2020-03-13 07:41:26.624 INFO 8326 --- [Test worker] org.hibernate.Version : HH000412: Hibernate ORM core version 5.4.12.Final
2020-03-13 07:41:26.748 INFO 8326 --- [Test worker] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations [5.1.0.Final]
2020-03-13 07:41:26.838 INFO 8326 --- [Test worker] org.hibernate.dialect.Dialect : HH0000400: Using dialect: org.hibernate.dialect.MySQLDialect
2020-03-13 07:41:27.371 INFO 8326 --- [Test worker] o.h.e.t.j.p.i.JtaPlatformInitiator : HH000490: Using JtaPlatform implementation: [org.hibernate.jpa.internal.util.LogHelper]
2020-03-13 07:41:27.377 INFO 8326 --- [Test worker] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2020-03-13 07:41:27.567 WARN 8326 --- [Test worker] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore this class is automatically loaded. Enable 'spring.jpa.open-in-view=false' in 'spring.jpa.properties.' if you wish to lazily load the classes only when explicitly requested.
2020-03-13 07:41:28.057 INFO 8326 --- [Test worker] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-03-13 07:41:28.353 INFO 8326 --- [Test worker] c.t.JPA_Demo.JpaDemoApplicationTests : Started JpaDemoApplicationTests in 3.569 seconds (JVM running for 3.569s)
Employee(name='Gurjan', id=1, age=22, location='Delhi')
2020-03-13 07:41:28.518 INFO 8326 --- [extShutdownHook] o.s.s.concurrent.ThreadPoolTaskExecutor : Shutting down ExecutorService 'applicationTaskExecutor'
2020-03-13 07:41:28.519 INFO 8326 --- [extShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2020-03-13 07:41:28.520 INFO 8326 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2020-03-13 07:41:28.531 INFO 8326 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
BUILD SUCCESSFUL in 5s
5 actionable tasks: 2 executed, 3 up-to-date
7:41:28 AM: Tasks execution finished 'cleanTest :test --tests 'com.tothenew.JPA_Demo.JpaDemoApplicationTests.testRead''.
```

Run | TODO | Terminal | Event Log

Tests passed: 1 (5 minutes ago) 55 chars 57:56 LF UTF-8 Tab*

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

```
package com.tothenew.JPA_Demo;
```

```
import com.tothenew.JPA_Demo.employeeEntities.Employee;
```

```
import com.tothenew.JPA_Demo.employeeEntities.EmployeeRepository;
```

```
import org.junit.jupiter.api.Test;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.boot.test.context.SpringBootTest;
```

```
import java.util.Optional;
```

```
@SpringBootTest
```

```
class JpaDemoApplicationTests {
```

```
    @Autowired
```

```
    EmployeeRepository employeeRepository;
```

```
    @Test
```

```
    public void contextLoads(){
```

```
    }
```

```
//Ques6 Get the total count of the number of Employees
```

```
@Test
```

```
public void testCount(){
```

```
    System.out.println("Total no of rows are >>>>> "+employeeRepository.count());
```

```
}
```

```
}
```

Activities IntelliJ IDEA Community Edition Fri 08:18

JPA_Demo [-/Documents/JPA_Demo] - .../src/test/java/com/tothenew/JPA_Demo/JpaDemoApplicationTests.java [JPA_Demo.test]

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

JPA_Demo ▾ src ▾ test ▾ java ▾ com ▾ tothenew ▾ JPA_Demo ▾ JpaDemoApplicationTests JpaDemoApplicationTests.testCount

Project ▾ JpaDemoApplicationTests.java ▾ Notes ▾

```
73  
74  
75  
76  
77  
@test  
public void testDelete(){  
    employeeRepository.deleteById(3);  
}
```

com.tothenew.JPA_Demo JpaDemoApplicationTests > testCount()

Run: JpaDemoApplicationTests.testCount

Tests passed: 1 of 1 test - 117 ms

Test 117 ms

```
2020-03-13 08:18:21.829 INFO 9208 --- [Test worker] .s.o.r.c.RepositoryConfigurationDelegate : Multiple Spring data modules found, entering strict  
2020-03-13 08:18:21.830 INFO 9208 --- [Test worker] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT  
2020-03-13 08:18:21.862 INFO 9208 --- [Test worker] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 28ms. Fo  
2020-03-13 08:18:22.480 INFO 9208 --- [Test worker] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...  
2020-03-13 08:18:22.823 INFO 9208 --- [Test worker] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.  
2020-03-13 08:18:22.877 INFO 9208 --- [Test worker] o.hibernate.jpa.internal.util.LogHelper : HH000204: Processing PersistenceUnitInfo [name: def  
2020-03-13 08:18:22.922 INFO 9208 --- [Test worker] org.hibernate.Version : HH000412: Hibernate ORM core version 5.4.12.Final  
2020-03-13 08:18:23.042 INFO 9208 --- [Test worker] org.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations (5.1.0.Fi  
2020-03-13 08:18:23.151 INFO 9208 --- [Test worker] org.hibernate.dialect.Dialect : HH000400: Using dialect: org.hibernate.dialect.MySQ  
2020-03-13 08:18:23.701 INFO 9208 --- [Test worker] o.h.e.t.j.p.i.JtaPlatformInitiator : HH000490: Using JtaPlatform implementation: [org.hi  
2020-03-13 08:18:23.708 INFO 9208 --- [Test worker] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence  
2020-03-13 08:18:23.880 WARN 9208 --- [Test worker] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. There  
2020-03-13 08:18:24.422 INFO 9208 --- [Test worker] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecuto  
2020-03-13 08:18:24.729 INFO 9208 --- [Test worker] c.t.JPA_Demo.JpaDemoApplicationTests : Started JpaDemoApplicationTests in 3.663 seconds (JVM  
Total no of rows are >>>>>2  
2020-03-13 08:18:24.885 INFO 9208 --- [extShutdownHook] o.s.s.concurrent.ThreadPoolTaskExecutor : Shutting down ExecutorService 'applicationTaskExecut  
2020-03-13 08:18:24.887 INFO 9208 --- [extShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence uni  
2020-03-13 08:18:24.889 INFO 9208 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...  
2020-03-13 08:18:24.899 INFO 9208 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.  
BUILD SUCCESSFUL in 5s  
5 actionable tasks: 3 executed, 2 up-to-date  
8:18:24 AM: Tasks execution finished 'cleanTest :test --tests "com.tothenew.JPA_Demo.JpaDemoApplicationTests.testCount"'.
```

Run ▾ Run ▾ TODO ▾ Terminal ▾ Event Log

Tests passed: 1 (moments ago) 28 chars 57:1 LF UTF-8 Tab*

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

```
package com.tothenew.JPA_Demo.employeeEntities;

import org.springframework.data.repository.CrudRepository;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;
import org.springframework.data.domain.Pageable;
import org.springframework.data.domain.Sort;

import java.util.List;
import java.util.Optional;

public interface EmployeeRepository extends CrudRepository<Employee,Integer> {

    List<Employee> findByName(String name);

}

//Ques8 Create and use finder to find Employee by Name
@Test
public void testfindByName(){
    List<Employee> emp= employeeRepository.findByName("Bhawna");
    System.out.println(">>>>>>>>>>>>>>>");
    System.out.println(emp);
    System.out.println(">>>>>>>>>>>>>>>");
}

//(7) Implement Pagination and Sorting on the bases of Employee Age
@Test
void testPaging() {

    Pageable pageable = PageRequest.of(0, 3, Sort.by(Sort.Direction.DISC,"age"));
    Page<Employee> page = empRepo.findAll(pageable);
    page.forEach(p -> System.out.println(p.getAge()));
}
```

```
package com.tothenew.JPA_Demo.employeeEntities;

import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.repository.PagingAndSortingRepository;

public interface EmpRepo extends PagingAndSortingRepository<Employee,Integer> {

    Page<Employee> findAll(Pageable pageable);

}
```

}

The screenshot shows an IDE window for a project named 'JPA_Demo'. The main editor displays the file 'JpaDemoApplicationTests.testPaging', which contains the following code:

```
1 package com.tothenew.JPA_Demo.employeeEntities;
2
3 import org.springframework.data.domain.Page;
4 import org.springframework.data.domain.Pageable;
5 import org.springframework.data.repository.PagingAndSortingRepository;
6
7 public interface EmpRepo extends PagingAndSortingRepository<Employee,Integer> {
8     Page<Employee> findAll(Pageable pageable);
9 }
10
```

The left sidebar shows the project structure:

- com.tothenew.JPA_Demo
 - employeeEntities
 - Employee
 - EmployeeRepository
 - EmpRepo
 - JpaDemoApplication
 - Notes
 - resources

The bottom panel shows the 'Run' output for 'JpaDemoApplicationTests.testPaging'. The output indicates that the test passed successfully in 231 ms. The log includes various startup messages from Spring and Hibernate, and a shutdown message from the HikariPool.

Run: JPA_Demo [JpaDemoApplication.m... x JpaDemoApplicationTests.testPaging x

Tests passed: 1 of 1 test - 231 ms

2020-03-14 16:09:03.573 INFO 31325 --- [Test worker] o.hibernate.annotations.common.Version : HLA00000001: Hibernate Commons Annotations {5.1.0.Final}

2020-03-14 16:09:03.666 INFO 31325 --- [Test worker] org.hibernate.dialect.Dialect : HH0000400: Using dialect: org.hibernate.dialect.MySQL5Dialect

2020-03-14 16:09:04.130 INFO 31325 --- [Test worker] o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000490: Using JtaPlatform implementation: [org.hibernate.jta.platform.internal.NoJtaPlatform]

2020-03-14 16:09:04.136 INFO 31325 --- [Test worker] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'

2020-03-14 16:09:04.335 WARN 31325 --- [Test worker] JpaBaseConfiguration\$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, SessionAttributes and OpenInViewConfiguration are not required classes in JPA applications. Please consider disabling this behavior by setting spring.jpa.open-in-view to false in your application configuration files.

2020-03-14 16:09:04.818 INFO 31325 --- [Test worker] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'

2020-03-14 16:09:05.213 INFO 31325 --- [Test worker] c.t.JPA_Demo.JpaDemoApplicationTests : Started JpaDemoApplicationTests in 3.521 seconds (JVM running for 10.15)

2020-03-14 16:09:05.476 INFO 31325 --- [Test worker] o.s.s.concurrent.ThreadPoolTaskExecutor : Shutting down ExecutorService 'applicationTaskExecutor'

2020-03-14 16:09:05.477 INFO 31325 --- [Test worker] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'

2020-03-14 16:09:05.479 INFO 31325 --- [Test worker] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...

2020-03-14 16:09:05.496 INFO 31325 --- [Test worker] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.

BUILD SUCCESSFUL in 5s

5 actionable tasks: 3 executed, 2 up-to-date

4:09:05 PM: Tasks execution finished 'cleanTest :test --tests "com.tothenew.JPA_Demo.JpaDemoApplicationTests.testPaging"'.

8 chars, 2 line breaks 56:1 LF UTF-8 4 spaces

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

```
package com.tothenew.JPA_Demo.employeeEntities;

import org.springframework.data.repository.CrudRepository;

import java.util.List;
import java.util.Optional;

public interface EmployeeRepository extends CrudRepository<Employee,Integer> {

    List<Employee> findByName(String name);

}
```

//Ques8 Create and use finder to find Employee by Name

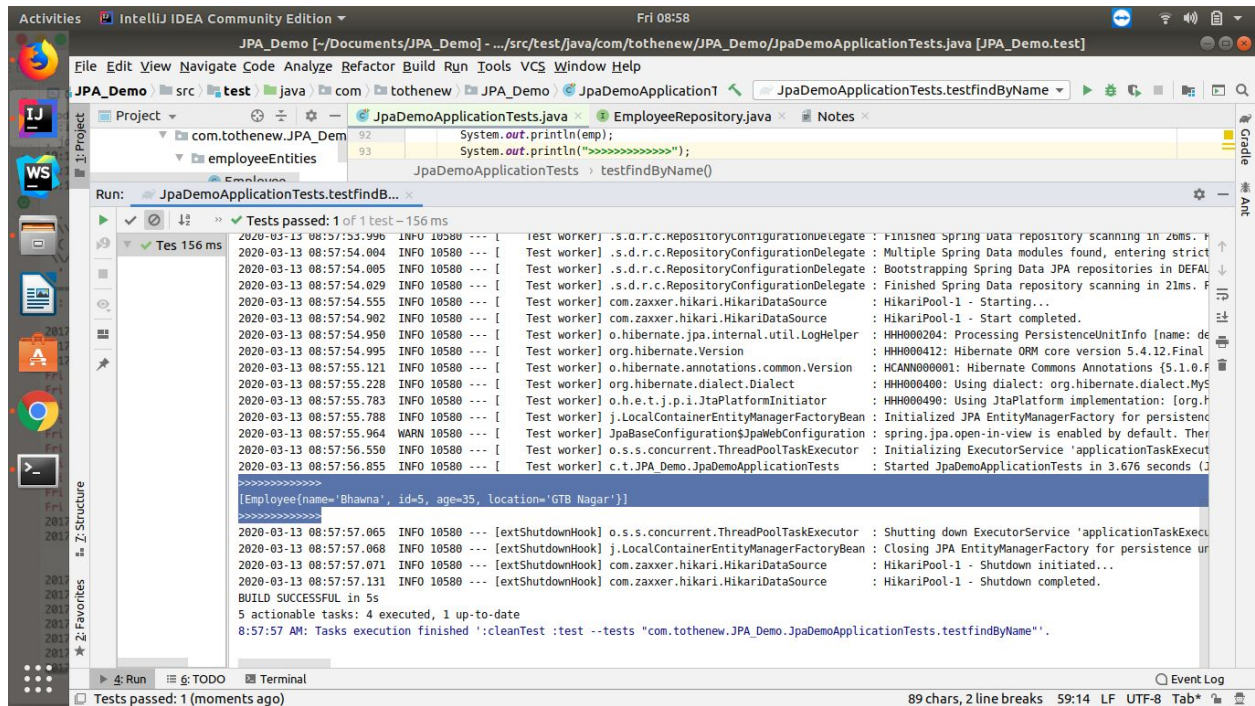
```
@Test
public void testfindByName(){
List<Employee> emp= employeeRepository.findByName("Bhawna");
    System.out.println(">>>>>>>>>>>>>>>>");
    System.out.println(emp);
    System.out.println(">>>>>>>>>>>>>>>>");

}
```

//(7) Implement Pagination and Sorting on the bases of Employee Age

```
@Test
void testPaging() {

    Pageable pageable = PageRequest.of(0, 3, Sort.by(Sort.Direction.DESC, "age"));
    Page<Employee> page = empRepo.findAll(pageable);
    page.forEach(p -> System.out.println(p.getAge()));
}
```



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

```
//Ques9 Create and use finder to find Employees starting with A character
```

@Test

```
public void testfindByCharacter() {
    List<Employee> emp = employeeRepository.findByNameLike("A%");
    System.out.println(">>>>>>>>>>>>>>>>");
    System.out.println(emp);
    System.out.println(">>>>>>>>>>>>>>>>");
}
}
```

```
package com.tothenew.JPA_Demo.employeeEntities;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import java.util.List;
```

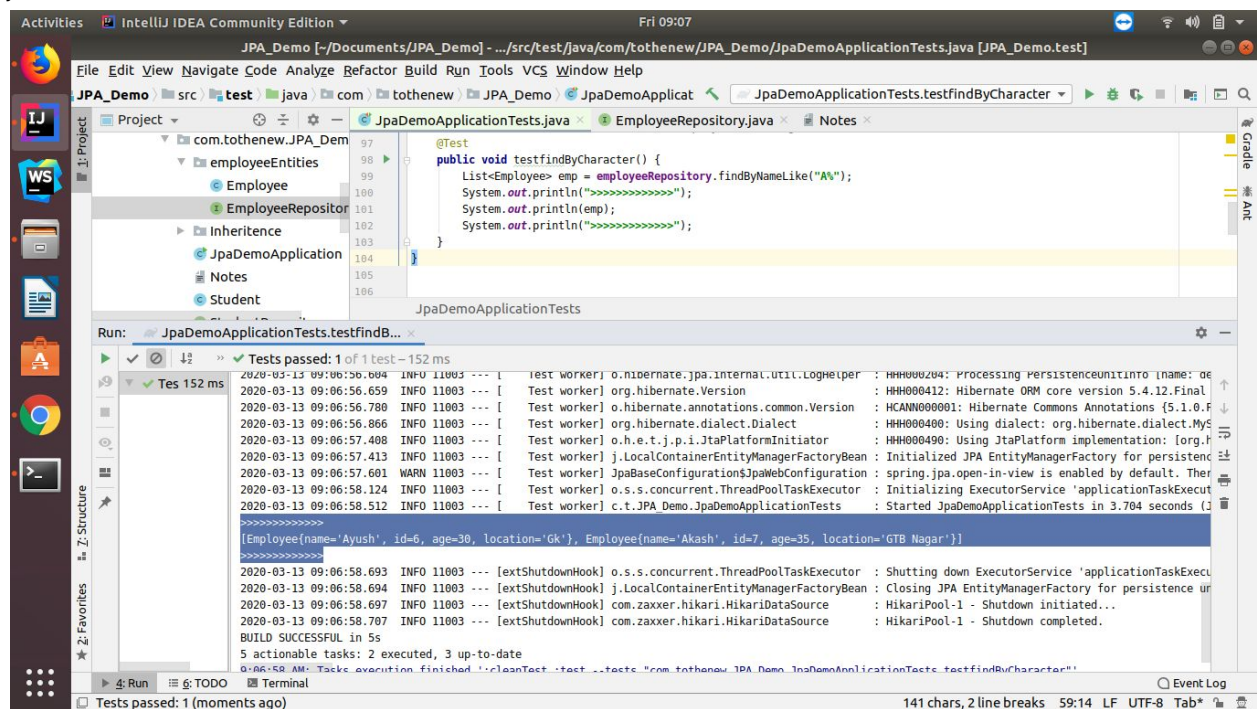
```
import java.util.Optional;
```

```
public interface EmployeeRepository extends CrudRepository<Employee,Integer> {
```

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

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

}



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

```
package com.tothenew.JPA_Demo.employeeEntities;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import java.util.List;
```

```
import java.util.Optional;
```

```
public interface EmployeeRepository extends CrudRepository<Employee,Integer> {
```

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

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

```
    List<Employee> findByAgeBetween(Integer age1,Integer age2);
```

```
}
```

```
-----  
---
```

```
//Ques10 Create and use finder to find Employees Between the age of 28 to 32
```

```
@Test
```

```
public void testfindAgeBetween() {
```

```
    List<Employee> emp = employeeRepository.findByAgeBetween(28,32);
```

```
    System.out.println(">>>>>>>>>>>>");
```

```
    System.out.println(emp);
```

```
    System.out.println(">>>>>>>>>>>>");
```

```
}
```