Spring Boot JPA - Custom methods

We've checked the methods available by default in Repository in JPA Methods chapter. Now let's add a method and test it.

Repository - EmployeeRepository.java

Example

Add a method to find an employee by its name.

```
package com.tutorialspoint.repository;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;
import com.tutorialspoint.entity.Employee;

@Repository
public interface EmployeeRepository extends CrudRepository<Employee, Intege
   public List<Employee> findByName(String name);
   public List<Employee> findByAge(int age);
}
```

Now Spring JPA will create the implementation of above methods automatically as we've following the property based nomenclature. Let's test the methods added by adding their test cases in test file. Last two methods of below file tests the custom methods added.

Following is the complete code of EmployeeRepositoryTest.

```
package com.tutorialspoint.repository;

import static org.junit.jupiter.api.Assertions.assertEquals;
import java.util.ArrayList;
import java.util.List;
import javax.transaction.Transactional;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit.jupiter.SpringExtension;
import com.tutorialspoint.entity.Employee;
import com.tutorialspoint.sprintbooth2.SprintBootH2Application;
```

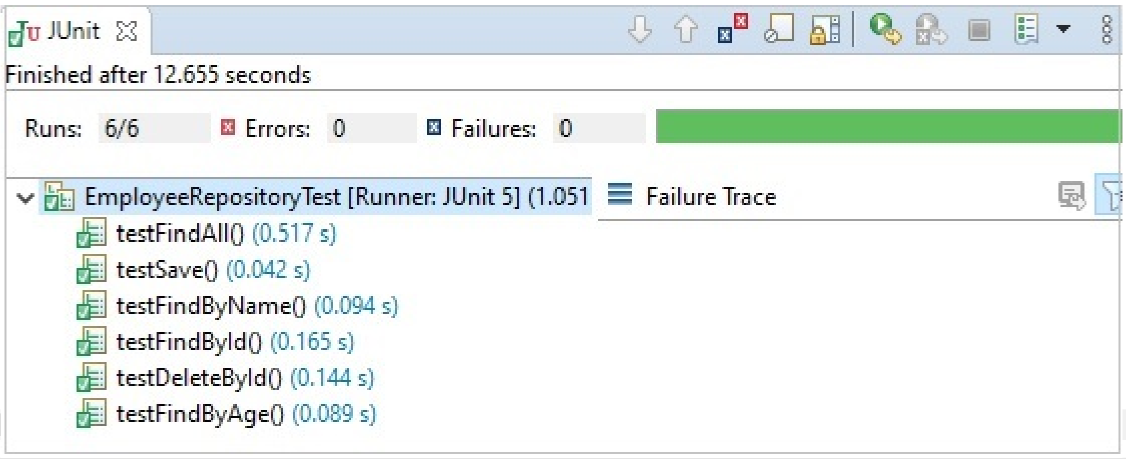
```
@ExtendWith(SpringExtension.class)
@Transactional
@SpringBootTest(classes = SprintBootH2Application.class)
public class EmployeeRepositoryTest {
   @Autowired
   private EmployeeRepository employeeRepository;
   @Test
   public void testFindById() {
      Employee employee = getEmployee();
      employeeRepository.save(employee);
      Employee result = employeeRepository.findById(employee.getId()).get(
      assertEquals(employee.getId(), result.getId());
   }
   @Test
   public void testFindAll() {
      Employee employee = getEmployee();
      employeeRepository.save(employee);
      List<Employee> result = new ArrayList<>();
      employeeRepository.findAll().forEach(e -> result.add(e));
      assertEquals(result.size(), 1);
   }
   @Test
   public void testSave() {
      Employee employee = getEmployee();
      employeeRepository.save(employee);
      Employee found = employeeRepository.findById(employee.getId()).get()
      assertEquals(employee.getId(), found.getId());
   }
   @Test
   public void testDeleteById() {
      Employee employee = getEmployee();
      employeeRepository.save(employee);
      employeeRepository.deleteById(employee.getId());
      List<Employee> result = new ArrayList<>();
      employeeRepository.findAll().forEach(e -> result.add(e));
      assertEquals(result.size(), 0);
   }
   private Employee getEmployee() {
      Employee employee = new Employee();
      employee.setId(1);
      employee.setName("Mahesh");
      employee.setAge(30);
      employee.setEmail("mahesh@test.com");
      return employee;
   }
```

```
    @Test
    public void testFindByName() {
       Employee employee = getEmployee();
       employeeRepository.save(employee);
       List<Employee> result = new ArrayList<>();
       employeeRepository.findByName(employee.getName()).forEach(e -> resul
       assertEquals(result.size(), 1);
    }
    @Test
    public void testFindByAge() {
       Employee employee = getEmployee();
       employeeRepository.save(employee);
       List<Employee> result = new ArrayList<>();
       employeeRepository.findByAge(employee.getAge()).forEach(e -> result.
       assertEquals(result.size(), 1);
    }
 }
```

Run the test cases

Output

Right Click on the file in eclipse and select **Run a JUnit Test** and verify the result.



```
Finished after 12.655 seconds
Runs: 6/6    Errors: 0    Failures: 0
EmployeeRepositoryTest [Runner: JUnit 5] (1.051
   testFindAll() (0.517 s)
   testSave() (0.042 s)
   testFindByName() (0.094 s)
   testFindById() (0.165 s)
   testDeleteById() (0.144 s)
   testFindByAge() (0.089 s)
Failure Trace
```