

Spring Boot JPA - Unit Test Repository

To test a Repository, we need the following annotation and classes –

- **@ExtendWith(SpringExtension.class)** – Mark the class to run as test case using SpringExtension class.
- **@SpringBootTest(classes = SprintBootH2Application.class)** – Configure the Spring Boot application.
- **@Transactional** – To mark repository to do CRUD Operation capable.
- **@Autowired private EmployeeRepository employeeRepository** – EmployeeRepository object to be tested.

Example

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.SprintBoothH2Application;

@ExtendWith(SpringExtension.class)
@Transactional
@SpringBootTest(classes = SprintBoothH2Application.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;
    }
}
```

Run the test cases

Output

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

