**package** com.example.onetomany;

**import** com.example.onetomany.model.Department;

**import** com.example.onetomany.model.Employee;

**import** com.example.onetomany.repo.Departmentrepo;

**import** com.example.onetomany.service.Departmentservice;

**import** org.junit.jupiter.api.BeforeEach;

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

**import** org.mockito.InjectMocks;

**import** org.mockito.Mock;

**import** org.mockito.MockitoAnnotations;

**import** java.util.Arrays;

**import** java.util.Optional;

**import** java.util.List;

**import** **static** org.junit.jupiter.api.Assertions.\*;

**import** **static** org.mockito.Mockito.\*;

**class** DepartmentserviceTest {

@InjectMocks

**private** Departmentservice departmentservice;

@Mock

**private** Departmentrepo departmentrepo;

@BeforeEach

**void** setUp() {

MockitoAnnotations.*openMocks*(**this**);

}

@Test

**void** testSaveDepartmentWithEmployees() {

// Arrange

Employee emp1 = **new** Employee();

emp1.setEname("John");

Employee emp2 = **new** Employee();

emp2.setEname("Alice");

Department dept = **new** Department();

dept.setName("CSE");

dept.setEmployees(Arrays.*asList*(emp1, emp2));

*when*(departmentrepo.save(dept)).thenReturn(dept);

// Act

Department result = departmentservice.saveDepartment(dept);

// Assert

*assertEquals*("CSE", result.getName());

*assertEquals*(2, result.getEmployees().size());

*verify*(departmentrepo, *times*(1)).save(dept);

}

// Test: Get all departments

@Test

**void** testGetAllDepartments() {

// Arrange

Department dept1 = **new** Department();

dept1.setName("CSE");

Department dept2 = **new** Department();

dept2.setName("ECE");

*when*(departmentrepo.findAll()).thenReturn(Arrays.*asList*(dept1, dept2));

// Act

List<Department> result = departmentservice.getAllDepartments();

// Assert

*assertEquals*(2, result.size());

*verify*(departmentrepo, *times*(1)).findAll();

}

// Test: Get department by ID

@Test

**void** testGetDepartmentById() {

// Arrange

Department dept = **new** Department();

dept.setId(1);

dept.setName("Mechanical");

*when*(departmentrepo.findById(1)).thenReturn(Optional.*of*(dept));

// Act

Department result = departmentservice.getDepartmentById(1);

// Assert

*assertNotNull*(result);

*assertEquals*("Mechanical", result.getName());

*verify*(departmentrepo, *times*(1)).findById(1);

}

// Test: Delete department by ID

@Test

**void** testDeleteDepartment() {

// Arrange

**int** id = 2;

// Act

departmentservice.deleteDepartment(id);

// Assert

*verify*(departmentrepo, *times*(1)).deleteById(id);

}

}