**PROJECT: EMPLOYEE PAYROLL SYSTEM**

**Unit Testing:**

**APP Config file:**

**package** com.mph.controller;

**import** org.hibernate.SessionFactory;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Configuration;

@Configuration

@ComponentScan(basePackages = {"com.mph.\*"})

**public** **class** AppConfig {

@Bean

SessionFactory sessionFactory() {

**return** **null**;

}

}

**Employee Rest Controller Test:**

**package** com.mph.controller;

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

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.junit.Assert;

**import** org.junit.Test;

**import** org.junit.runner.RunWith;

**import** org.mockito.InjectMocks;

**import** org.mockito.Mock;

**import** org.mockito.runners.MockitoJUnitRunner;

**import** org.springframework.test.context.ContextConfiguration;

**import** com.mph.entity.Employee;

**import** com.mph.entity.Salary;

**import** com.mph.service.EmployeeServiceImpl;

@RunWith(MockitoJUnitRunner.**class**)

@ContextConfiguration(classes = { AppConfig.**class** })

**public** **class** EmployeeRestControllerTest {

@InjectMocks

EmployeeRestController employeeRestController;

@Mock

EmployeeServiceImpl employeeService;

Salary sal,sal1;

@Test

**public** **void** testAllEmployee() {

Employee emp = **new** Employee(101, "Chandra", "30-08-1989", 9620500984L, "chandra@gmail.com", "30-10-2020", "M",

"Lead", "kurnool", "testing",sal);

List<Employee> employees = **new** ArrayList<>();

employees.add(emp);

Assert.assertNotNull(employeeRestController.allEmployee());

Assert.assertEquals(204, employeeRestController.allEmployee().getStatusCodeValue());

*when*(employeeService.getEmployeeList()).thenReturn(employees);

Assert.assertNotNull(employeeRestController.allEmployee());

Assert.assertEquals(1, employeeRestController.allEmployee().getBody().size());

}

@Test

**public** **void** testGetEmployee() {

Employee emp = **new** Employee(101, "Chandra", "30-08-1989", 9620500984L, "chandra@gmail.com", "30-10-2020", "M",

"Lead", "kurnool", "testing",sal);

Employee emp1 = **new** Employee(102, "sekhar", "30-08-1989", 9620500984L, "chandra@gmail.com", "30-10-2020", "M",

"Lead", "kurnool", "testing",sal1);

*when*(employeeService.getEmployeeById(101)).thenReturn(emp);

*when*(employeeService.getEmployeeById(102)).thenReturn(emp1);

Assert.assertEquals("Chandra", employeeRestController.getEmployee(101).getBody().getEname());

Assert.assertEquals("sekhar", employeeRestController.getEmployee(102).getBody().getEname());

}

}

**Employee Service Implementation Test:**

**package** com.mph.service;

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

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.junit.Assert;

**import** org.junit.Test;

**import** org.junit.runner.RunWith;

**import** org.mockito.InjectMocks;

**import** org.mockito.Mock;

**import** org.mockito.runners.MockitoJUnitRunner;

**import** org.springframework.test.context.ContextConfiguration;

**import** com.mph.controller.AppConfig;

**import** com.mph.dao.EmployeeDao;

**import** com.mph.entity.Employee;

**import** com.mph.entity.Salary;

@RunWith(MockitoJUnitRunner.**class**)

@ContextConfiguration(classes = { AppConfig.**class** })

**public** **class** EmployeeServiceImplTest {

@InjectMocks

EmployeeServiceImpl employeeService;

@Mock

EmployeeDao employeeDao;

Salary sal,sal1;

@Test

**public** **void** testGetEmployeeList() {

Employee emp = **new** Employee(101, "Chandra", "30-08-1989", 9620500984L, "chandra@gmail.com", "30-10-2020", "M",

"Lead", "kurnool", "testing",sal);

List<Employee> employees = **new** ArrayList<>();

employees.add(emp);

*when*(employeeDao.getEmployeeList()).thenReturn(employees);

Assert.*assertNotNull*(employeeService.getEmployeeList());

Assert.*assertEquals*(1, employeeService.getEmployeeList().size());

}

@Test

**public** **void** getEmployeeById() {

Employee emp = **new** Employee(101, "Chandra", "30-08-1989", 9620500984L, "chandra@gmail.com", "30-10-2020", "M",

"Lead", "kurnool", "testing",sal);

Employee emp1 = **new** Employee(102, "sekhar", "30-08-1989", 9620500984L, "chandra@gmail.com", "30-10-2020", "M",

"Lead", "kurnool", "testing",sal1);

*when*(employeeDao.getEmployeeById(101)).thenReturn(emp);

*when*(employeeDao.getEmployeeById(102)).thenReturn(emp1);

Assert.*assertEquals*("Chandra", employeeService.getEmployeeById(101).getEname());

Assert.*assertEquals*("sekhar", employeeService.getEmployeeById(102).getEname());

}

}