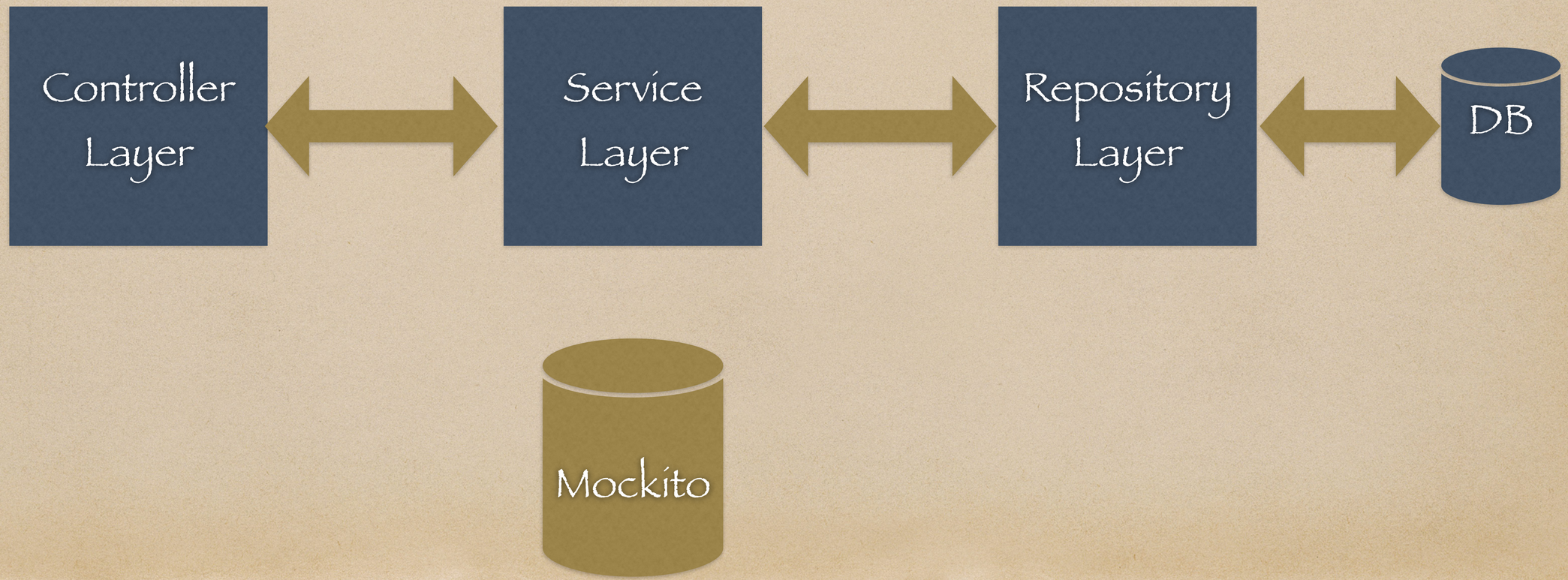
Spring Boot Application Service Layer Testing

By Ramesh Fadatare (Java Guides)

Spring Boot Application
Service Layer Testing



Service Layer Testing

1. **Create EmployeeService with saveEmployee method**
2. **Quick Recap of Mockito basics (before writing Unit tests)**
3. **Unit test for EmployeeService saveEmployee method**
4. **Using @Mock and @InjectMocks annotations to mock the object**
5. **Unit test for saveEmployee method which throws Exception**
6. **Unit test for EmployeeService getAllEmployees method - Positive Scenario**
7. **Unit test for EmployeeService getAllEmployees method - Negative Scenario**
8. **Unit test for EmployeeService getEmployeeById method**
9. **Unit test for EmployeeService updateEmployee method**
10. **Unit test for EmployeeService deleteEmployee method**

JUnit tests in BDD Style

Syntax

```
@Test
public void given_when_then() {
    // given - precondition or setup

    // when - action or the behaviour we're testing

    // then - verify the output
}
```

Example

```
// JUnit test for saveEmployee method
@DisplayName("JUnit test for saveEmployee method which throws exception")
@Test
public void givenExistingEmail_whenSaveEmployee_thenThrowsException(){
    // given - precondition or setup
    given(employeeRepository.findByEmail(employee.getEmail()))
            .willReturn(Optional.of(emplo

    System.out.println(employeeRepository);
    System.out.println(employeeService);

    // when -  action or the behaviour that we are going test
    org.junit.jupiter.api.Assertions.assertThrows(ResourceNotFoundException.class, () -> {
        employeeService.saveEmployee(employee);
    });

    // then
    verify(employeeRepository, never()).save(any(Employee.class));
}
```

```
net.javaguides.springboot.service.EmployeeServiceTes
private Employee employee
```