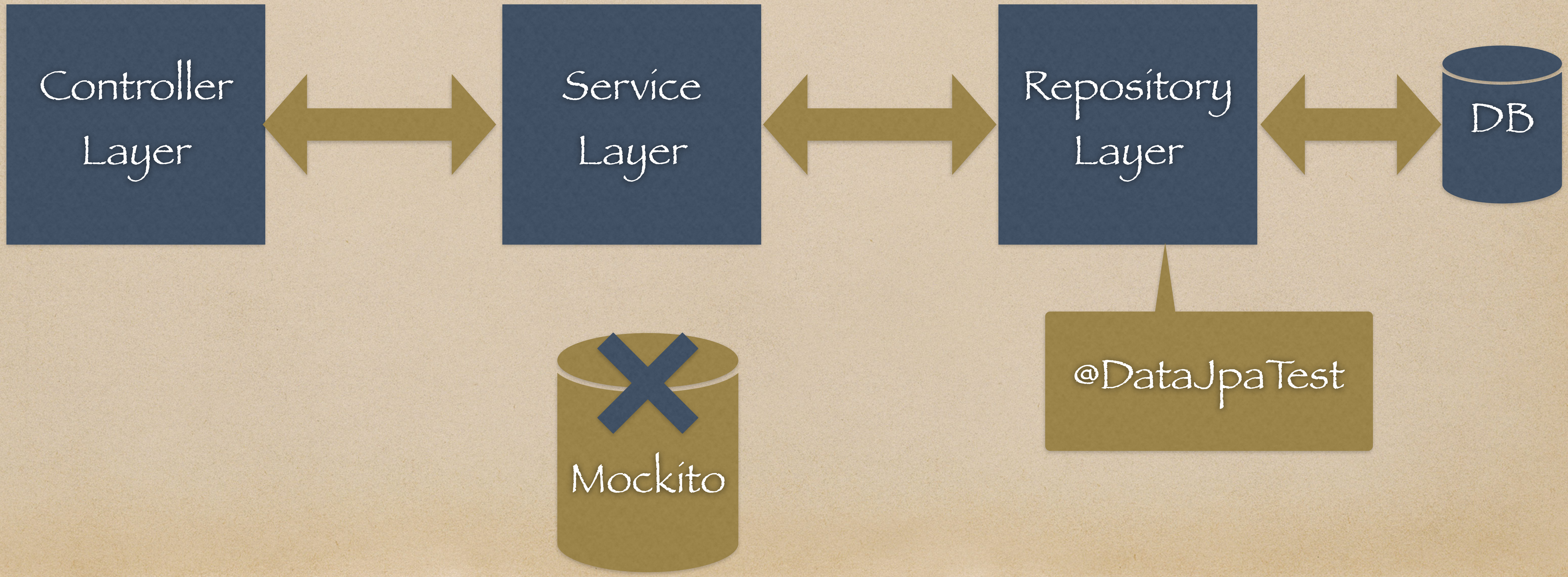
Spring Boot Application Repository Layer Testing

By Ramesh Fadatare (Java Guides)

Spring Boot Application Repository Layer Testing



Repository Testing

1.Overview of @DataJpaTest annotation

2.Unit test for save employee operation

3.Unit test for get all employees operation

4.Unit test for get employee by id operation

5.Unit test for get employee by email operation (custom query)

6.Unit test for update employee operation

7.Unit test for delete employee operation

8.Unit test for custom query using JPQL with index parameters (using @Query annotation)

9.Unit test for custom query using JPQL with named parameters (using @Query annotation)

10.Unit test for custom Native query with index parameters (using @Query annotation)

11.Unit test for custom Native query with Named parameters (using @Query annotation)

12.Refactoring JUnit tests to use @BeforeEach annotation

JUnit tests in BDD Style

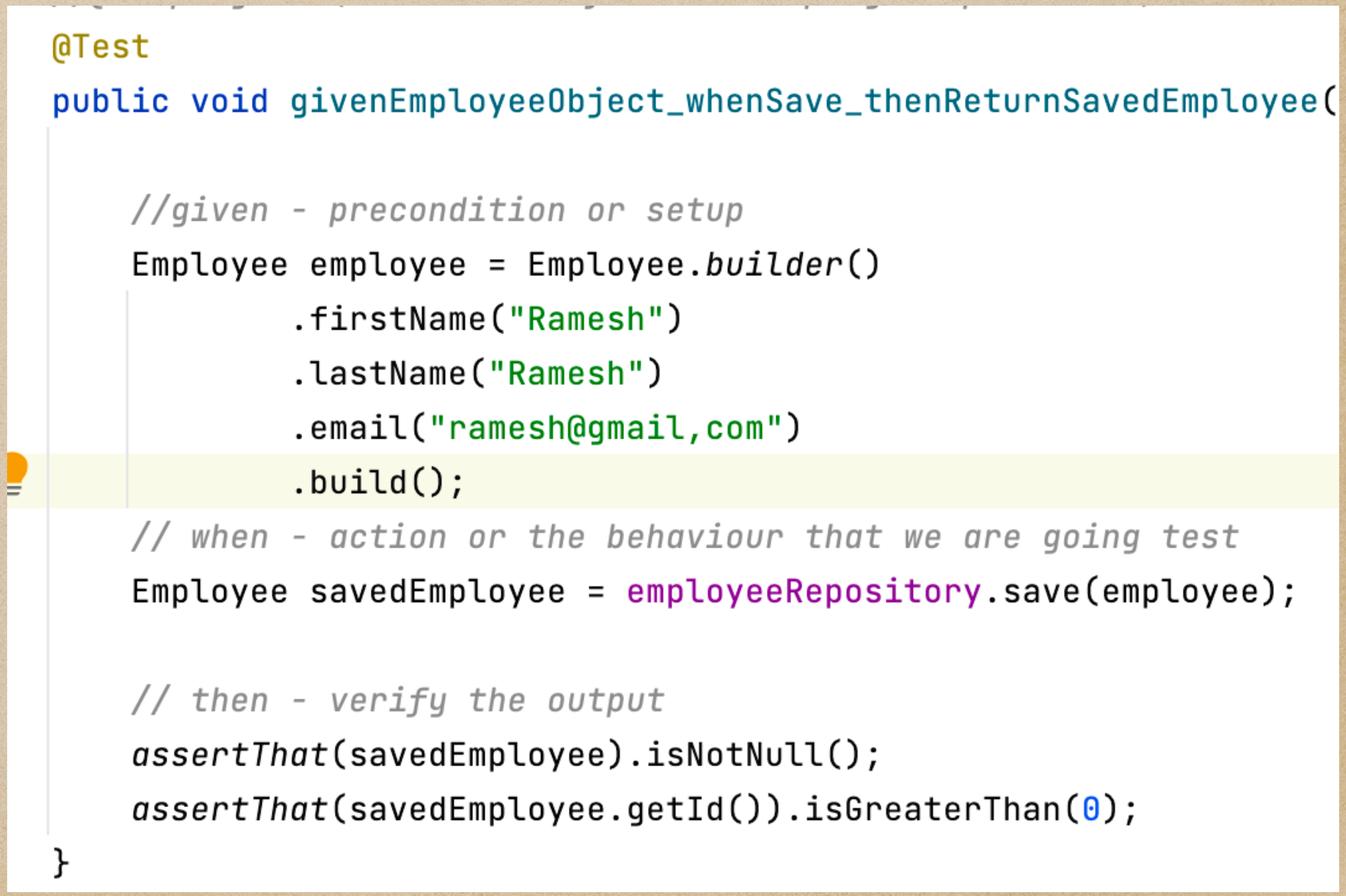
Syntax

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

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

	// then - verify the output
}
```

Example



```
@Test
public void givenEmployeeObject_whenSave_thenReturnSavedEmployee(

    //given - precondition or setup
    Employee employee = Employee.builder()
            .firstName("Ramesh")
            .lastName("Ramesh")
            .email("ramesh@gmail,com")
            .build();
    // when - action or the behaviour that we are going test
    Employee savedEmployee = employeeRepository.save(employee);

    // then - verify the output
    assertThat(savedEmployee).isNotNull();
    assertThat(savedEmployee.getId()).isGreaterThan(0);
}
```