**Difference between Spy and Mock in Mockito**

Share the post "Difference between Spy and Mock in Mockito"

Often you heard developers how to **spy** and **mock** in **Mockito** in unit test but what are the **difference between spy and mock in Mockito** API? Both can be used to mock methods or fields. The difference is that in mock, you are creating a complete mock or fake object while in spy, there is the real object and you just spying or stubbing specific methods of it.

When using **mock objects**, the default behavior of the method when not stub is **do nothing**. Simple means, if its a void method, then it will do nothing when you call the method or if its a method with a return then it may return null, empty or the default value.

While in spy objects, of course, since it is a real method, when you are not stubbing the method, then it will call the real method behavior. If you want to change and mock the method, then you need to stub it.

Consider the example below as a comparison.

[?](https://javapointers.com/tutorial/difference-between-spy-and-mock-in-mockito/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | import org.junit.Test;  import org.junit.runner.RunWith;  import org.mockito.Mock;  import org.mockito.Spy;  import org.mockito.runners.MockitoJUnitRunner;    import java.util.ArrayList;  import java.util.List;    import static org.junit.Assert.assertEquals;  import static org.junit.Assert.assertNull;  import static org.mockito.Mockito.doReturn;  import static org.mockito.Mockito.when;    @RunWith(MockitoJUnitRunner.class)  public class MockSpy {        @Mock      private List<String> mockList;        @Spy      private List<String> spyList = new ArrayList();        @Test      public void testMockList() {          //by default, calling the methods of mock object will do nothing          mockList.add("test");          assertNull(mockList.get(0));      }        @Test      public void testSpyList() {          //spy object will call the real method when not stub          spyList.add("test");          assertEquals("test", spyList.get(0));      }        @Test      public void testMockWithStub() {          //try stubbing a method          String expected = "Mock 100";          when(mockList.get(100)).thenReturn(expected);            assertEquals(expected, mockList.get(100));      }        @Test      public void testSpyWithStub() {          //stubbing a spy method will result the same as the mock object          String expected = "Spy 100";          //take note of using doReturn instead of when          doReturn(expected).when(spyList).get(100);            assertEquals(expected, spyList.get(100));      }  } |

When shoud you use mock or spy? If you want to be safe and avoid calling external services and just want to test the logic inside of the unit, then use mock. If you want to call external service and perform calling of real dependency, or simply say, you want to run the program as it is and just stub specific methods, then use spy. So that’s the difference between spy and mock in mockito.