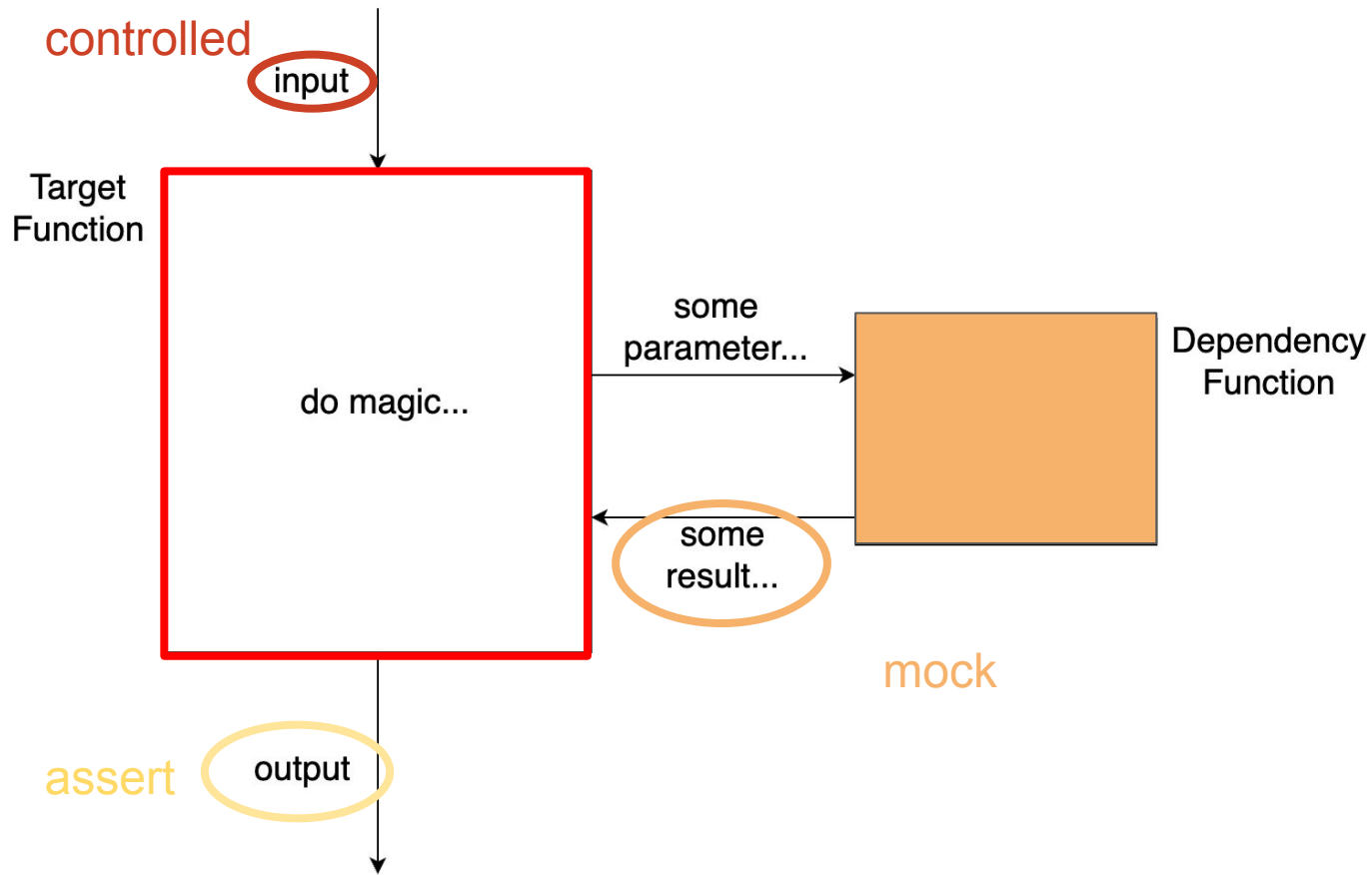


JUnit Test For Spring-boot API

COMS W4156 Fall 2022



Mock

- Five “Test Doubles”: Dummy, Fake, Stub, **Mock**, Spy
- Mock objects created at runtime
- Stubbing (happy case, and exception handling)
- Behavior verification

Spy

- Five “Test Doubles”: Dummy, Fake, Stub, Mock, **Spy**
- Real objects (stubbing specific methods)
- State verification

Spy (*example of when-to-use*)

```
public void transfer( DepositMoneyService depositMoneyService,  
                     WithdrawMoneyService withdrawMoneyService,  
                     double amount, String fromAccount, String toAccount) {  
    withdrawMoneyService.withdraw(fromAccount, amount);  
    depositMoneyService.deposit(toAccount, amount);  
}
```

Easy mock!!

```
public void transfer(String fromAccount, String toAccount, double amount) {  
    this.depositMoneyService = new DepositMoneyService();  
    this.withdrawMoneyService = new WithdrawMoneyService();  
    withdrawMoneyService.withdraw(fromAccount, amount);  
    depositMoneyService.deposit(toAccount, amount);  
}
```

```
public void transfer(String fromAccount, String toAccount, double amount){  
    this.depositMoneyService = proxyDepositMoneyServiceCreator();  
    this.withdrawMoneyService = proxyWithdrawMoneyServiceCreator();  
    withdrawMoneyService.withdraw(fromAccount, amount);  
    depositMoneyService.deposit(toAccount, amount);  
}
```

```
DepositMoneyService proxyDepositMoneyServiceCreator() {  
    return new DepositMoneyService();  
}
```

```
WithdrawMoneyService proxyWithdrawMoneyServiceCreator() {  
    return new WithdrawMoneyService();  
}
```

```
DepositMoneyService mockDepositMoneyService = mock(DepositMoneyService.class);  
WithdrawMoneyService mockWithdrawMoneyService = mock(WithdrawMoneyService.class);
```

```
TransferMoneyService target = spy(new TransferMoneyService());
```

```
doReturn(mockDepositMoneyService)  
    .when(target)  
    .proxyDepositMoneyServiceCreator();
```

```
doReturn(mockWithdrawMoneyService)  
    .when(target)  
    .proxyWithdrawMoneyServiceCreator();
```

Mockito

- Mock
- Spy (not supported by EasyMock)
- InjectMocks
- verify()

```

@Mock Echo mockEcho;

@Test
public void verificationInOrder() {
    mockEcho.echo("1");
    mockEcho.echo("2");
    mockEcho.echo("3");

    // verify without order
    verify(mockEcho).echo("3");
    verify(mockEcho).echo("2");
    verify(mockEcho).echo("1");

    // verify order
    InOrder inOrder = inOrder(mockEcho);
    inOrder.verify(mockEcho).echo("1");
    inOrder.verify(mockEcho).echo("2");
    inOrder.verify(mockEcho).echo("3");

    assertEquals(echoResponse("0003"), mockEcho.echo("42"));
}

```



```
@Test
public void capturingArguments() {
    ArgumentCaptor<EchoRequest> captorRequest =
        ArgumentCaptor.forClass(EchoRequest.class);
    EchoRequest echoRequest = new EchoRequest();
    echoRequest.id = "23";
    mockEcho.echo(echoRequest, "0000");
    verify(mockEcho).echo(captorRequest.capture(), eq("0000"));
    assertEquals("23", captorRequest.getValue().id);
}
```

- What about static methods?
- ... and private ones?

Highly NOT-recommended!!! But things happen...

PowerMock

- Serves as an extension to other libraries, such as Mockito and EasyMock
- Static, private, final

test2.publicMethod();

```
@RunWith(PowerMockRunner.class)
@PrepareForTest(Test1.class)
public class PowerMockTest {
    @Test
    public void staticVerifyTest() {
        PowerMockito.mockStatic(Test1.class);
        // Test
        Test1.staticMethod();
        PowerMockito.verifyStatic();
        Test1.staticMethod();
    }
}
```

```
@RunWith(PowerMockRunner.class)
@PrepareForTest(MyClient.class)
public class MyClientTest {

    @Test(expected = RuntimeException.class)
    public void testPublicAPI() throws Exception {
        MyClient classUnderTest = PowerMockito.spy(new MyClient());

        // Change to this

        PowerMockito.doReturn(20).when(classUnderTest, "privateApi", anyString(), anyInt());

        classUnderTest.publicApi();
    }
}
```

Now let's write a small test together!

Useful Links

- [Beginner fast wrap-up guide on JUnit with Spring Boot on REST API](#)
- ... [and combine the above with this tutorial from Spring.io, which is more up-to-dated](#)
- [Add test coverage report into your pom.xml dependency](#)
- IDE related:
 - [VisualStudio](#)
 - [Eclipse](#)
 - [IntelliJ](#)
- [Maven official guide on testing commands](#)
- ... [or you can refer to this more developer-friendly tutorial](#)
- [All kinds of assertions you may want to use to verify the function results](#)
- ... [and of course, more friendly tutorial here \(yes, unofficial version\)](#)
- [Mockito cheat sheet](#)
- [PowerMock with Mockito \(official but easy to follow\)](#)
- [Test with expected error](#)