

# Mockito When/Then Cookbook

 [www.baeldung.com/mockito-behavior](http://www.baeldung.com/mockito-behavior)

By Eugen Paraschiv

If you're new here, [join the next "CQRS and Event Sourcing with Spring" Webinar](#). Thanks for visiting!

## 1. Overview

This cookbook shows **how to use Mockito to configure behavior** in a variety of examples and usecases.

The **format of the cookbook is example focused** and practical – no extraneous details and explanations necessary.

We're going to be **mocking a simple list** implementation – the same implementation we used in the previous cookbook:

```
public class MyList extends AbstractList<String>
{
    @Override
    public String get(final int index) {
        return null;
    }
    @Override
    public int size() {
        return 1;
    }
}
```

## 2. Cookbook

### configure simple return behavior for mock

```
MyList listMock = Mockito.mock(MyList.class);
when(listMock.add(anyString())) .thenReturn(false);
```

```
boolean added =
listMock.add(randomAlphabetic(6));
assertThat(added, is(false));
```

### configure return behavior for mock in alternative way

```
MyList listMock = Mockito.mock(MyList.class);
doReturn(false).when(listMock).add(anyString());
```

```
boolean added =
listMock.add(randomAlphabetic(6));
assertThat(added, is(false));
```

### configure mock to throw exception on method call

```

@Test(expected = IllegalStateException.class)
public void
givenMethodIsConfiguredToThrowException_whenCallingMethod_thenExceptionIsThrown() {
    MyList listMock = Mockito.mock(MyList.class);
    when(listMock.add(anyString())) .thenThrow(IllegalStateException.class);

    listMock.add(randomAlphabetic(6));
}

```

### **configure behavior of a method with void return type – to throw an exception**

```

MyList listMock = Mockito.mock(MyList.class);
doThrow(NullPointerException.class).when(listMock).clear();

listMock.clear();

```

### **configure behavior of multiple calls**

```

MyList listMock = Mockito.mock(MyList.class);
when(listMock.add(anyString())) .thenReturn(false) .thenThrow(IllegalStateException.class);

listMock.add(randomAlphabetic(6));
listMock.add(randomAlphabetic(6)); // will throw the exception

```

### **configure behavior of spy**

```

MyList instance = new MyList();
MyList spy = Mockito.spy(instance);

doThrow(NullPointerException.class).when(spy).size();
spy.size(); // will throw the exception

```

### **configure method to call the real, underlying method on a mock**

```

MyList listMock =
Mockito.mock(MyList.class);
when(listMock.size()).thenCallRealMethod();

assertThat(listMock.size(), equalTo(1));

```

### **configure mock method call with custom Answer**

```
MyList listMock = Mockito.mock(MyList.class);
doAnswer(new Answer<String>() {
    @Override
    public String answer(InvocationOnMock invocation)
    {
        return "Always the same";
    }
}).when(listMock).get(anyInt());

String element = listMock.get(1);
assertThat(element, is(equalTo("Always the same")));
```

### 3. Conclusion

This format is an experiment – I'm publishing some of my internal development cookbooks on a given topic – on [Google Guava](#), [Hamcrest](#) and [now Mockito](#). The goal is to have this information readily available online – and to add to it whenever I run into a new useful example.

The implementation of all these examples and code snippets **can be found in [my Mockito github project](#)** – this is an Eclipse based project, so it should be easy to import and run as it is.

**Get the early-bird price (20% Off) of my upcoming "*Learn Spring Security*" Course:**

**[>> CHECK OUT THE COURSE](#)**