Mockito Verify Cookbook

www.baeldung.com/mockito-verify

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 illustrates **how to use Mockito verify** in a variety of 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:

```
public class MyList extends AbstractList<String>
{

    @Override
    public String get(final int index) {
        return null;
    }
    @Override
    public int size() {
        return 0;
    }
}
```

2. The Cookbook

verify simple invocation on mock

```
List<String> mockedList =
mock(MyList.class);
mockedList.size();
verify(mockedList).size();
```

verify number of interactions with mock

```
List<String> mockedList =
mock(MyList.class);
mockedList.size();
verify(mockedList, times(1)).size();
```

verify no interaction with the whole mock occurred

```
List<String> mockedList =
mock(MyList.class);
verifyZeroInteractions(mockedList);
```

verify no interaction with a specific method occurred

```
List<String> mockedList =
mock(MyList.class);
verify(mockedList, times(0)).size();
```

verify there are no unexpected interactions – this should fail:

```
List<String> mockedList =
mock(MyList.class);
mockedList.size();
mockedList.clear();
verify(mockedList).size();
verifyNoMoreInteractions(mockedList);
```

verify order of interactions

```
List<String> mockedList = mock(MyList.class);
mockedList.size();
mockedList.add("a parameter");
mockedList.clear();

InOrder inOrder =
Mockito.inOrder(mockedList);
inOrder.verify(mockedList).size();
inOrder.verify(mockedList).add("a parameter");
inOrder.verify(mockedList).clear();
```

verify an interaction has not occurred

```
List<String> mockedList =
mock(MyList.class);
mockedList.size();
verify(mockedList, never()).clear();
```

verify an interaction has occurred at least certain number of times

```
List<String> mockedList =
mock(MyList.class);
mockedList.clear();
mockedList.clear();
mockedList.clear();

verify(mockedList, atLeast(1)).clear();
verify(mockedList, atMost(10)).clear();
```

verify interaction with exact argument

```
List<String> mockedList =
mock(MyList.class);
mockedList.add("test");
verify(mockedList).add("test");
```

verify interaction with flexible/any argument

```
List<String> mockedList =
mock(MyList.class);
mockedList.add("test");
verify(mockedList).add(anyString());
```

verify interaction using argument capture

```
List<String> mockedList = mock(MyList.class);
mockedList.addAll(Lists.<String> newArrayList("someElement"));
ArgumentCaptor<List> argumentCaptor =
ArgumentCaptor.forClass(List.class);
verify(mockedList).addAll(argumentCaptor.capture());
List<String> capturedArgument = argumentCaptor.<List<String>> getValue();
assertThat(capturedArgument, hasItem("someElement"));
```

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