

The logo for Mockito, featuring the word "mockito" in a sans-serif font. The "mock" part is green and the "ito" part is black. To the right of the text is a glass filled with a clear liquid, lime slices, and mint leaves, with two black straws. The entire logo is reflected below it.

mockito

This Photo by Unknown Author is licensed under [CC BY-NC](#)

mockito

A mocking framework

E-Portfolio by Morten Harter

Agenda

- What is a mock object?
- What is Mockito?
- How do I use Mockito? Coding Example
- Other mocking frameworks

What is a mock object?

- Simulates real objects
- Has predefined return values
- Can be realized with interfaces
- Enables testing without dependencies

What is mockito?

- It is a mocking framework
- Creates mock or spy objects based on class files
- Allows you define behavior (return values) of mocked objects
- Lets you verify if and how many times a method was called



How to use mockito

Create and use mock objects

```
//Create mock and define behavior  
Random random = mock(Random.class);  
when(random.nextInt()).thenReturn(1);  
  
//Call method and verify call  
random.nextInt();  
verify(random).nextInt();
```

Create and use spy objects

```
//Create spy  
Random random = new Random();  
Random spy = spy(random);  
when(spy.nextDouble()).thenReturn(1.0);
```

```
//Call methods (method calls can also be verified)  
spy.nextInt(); // uses regular implementation  
spy.nextDouble(); // returns predefined value
```

Argument Matchers

```
List mockedList = mock(List);  
//Use argument matcher  
when(mockedList.get(anyInt())).thenReturn("element");  
//also argThat() can be used with own implementation  
verify(mockedList).add(argThat(string -> string.length() > 5));  
//Now can be called with any Integer  
mockedList.get(1337);  
//anyInt() can also be used for verification  
verify(mockedList).get(anyInt());
```


Verification of Calls

```
List mockedList = mock(List);  
  
//Verify invocations (times(1) is default)  
verify(mockedList, times(2)).add("foo");  
verify(mockedList, never()).add("foo");  
verify(mockedList, atLeastOnce()).add("foo");  
verify(mockedList, atLeast(2)).add("foo");  
verify(mockedList, atMost(10)).add("foo");
```

Stub consecutive calls

```
Random random = mock(Random.class);  
  
//stubbing can be chained  
when(random.nextBoolean()).thenReturn(true, false);  
when(random.nextInt()).thenReturn(1)  
    .thenThrow(new Exception());  
  
//alternate version  
doReturn(2.0, 1.0).when(random).nextDouble();
```



Coding Example

Other mocking frameworks

- JavaScript: Sinon.JS
- Java: Mockito, PowerMock
- C#/.NET: moq
- Python: Mock
- Interfaces can be used for mocking purposes



Questions?