Configuring Return Values of Mocked Methods

Nicolae Caprarescu
FULL-STACK SOFTWARE DEVELOPMENT CONSULTANT

www.properjava.com

Overview

Configuring your first mocked methods

- Also known as method stubbing
- Or setting expectations

Solidifying the knowledge gained so far

Using argument matchers

Configuring exception throwing

Demo

This module is 100% demo

Let’s write some code

Argument Matchers

Increased flexibility for method stubbing (verification too; more later)

Useful when you're not interested in the values of arguments

Or when you want to a return value for a range of arguments (or for an unknown argument)

Summary

Unconfigured mocks

Some more tests

Exceptions

Method stubbing

Argument matchers

Configured mocks

Summary

Unconfigured mocks

Some more tests

Exceptions

Method stubbing

Argument matchers

Configured mocks

Recap: Method Stubbing

when...thenReturn

Mockito has a default "loose" mocking behavior

This means that a non-void method of a mock, without any set expectations, will still return a value: null, primitive (or the equivalent wrapper), or an empty collection

Summary

Unconfigured mocks

Some more tests

Exceptions

Method stubbing

Argument matchers

Configured mocks

Summary

Unconfigured mocks

Some more tests

Exceptions

Method stubbing

Argument matchers

Configured mocks

Recap: Argument Matchers

any*(...), eq(...), isNull() etc.

They add flexibility to method stubbing (and verification; more later)

For readable tests, prefer Mockito's natural style of using actual values combined with limited any*() usage

Summary

Unconfigured mocks

Some more tests

Exceptions

Method stubbing

Argument matchers

Configured mocks

Recap: Configuring Exceptions

when...thenThrow and doThrow...when

It's important to also test these exceptional paths

Can stub methods with both checked and unchecked exceptions

Summary

Unconfigured mocks

Some more tests

Exceptions

Method stubbing

Argument matchers

Configured mocks