Testing Void Methods: Techniques and Pitfalls

Techniques for Testing Void Methods

doAnswer() in action

```java
// Example of using doAnswer() to test a void method that sends a message to a queue
doAnswer(invocation -> {
    String message = invocation.getArgument(0, String.class);
    // Assert that the message was sent correctly, for example:
    assertNotNull(message);
    assertTrue(message.startsWith("ExpectedMessagePrefix"));
    // Further actions can be taken here if necessary, such as tracking the messages sent
    return null; // No return value needed since the method is void
}).when(mockedQueue).sendMessage(anyString());



// P.S. you can rewrite this example using ArgumentCaptor
```

doNothing() in action

```
// Example of using doNothing()
// for a method that should not perform its action in the test
doNothing().when(mockedLogger).logError(anyString());
```

doThrow() in action

```
@Test
void testHandleFileWhenFileProcessorThrowsException()
    // Create a mock of FileProcessor
    FileProcessor mockFileProcessor = mock(FileProcessor.class);

    // Setup the mock to throw an exception when processFile is called
    doThrow(new IOException("Disk not found"))
             .when(mockFileProcessor).processFile(anyString());

    // Create an instance of FileHandler with the mocked FileProcessor
    FileHandler fileHandler = new FileHandler(mockFileProcessor);

    // Call handleFile method
    fileHandler.handleFile("important_data.txt");

    // Verify that the FileProcessor's processFile method was called
    verify(mockFileProcessor).processFile("important_data.txt");
}
```

Mockito’s Verification in Void Methods

Example of verify()

```
// Assume we have a mock of our repository
MyRepository mockRepository = mock(MyRepository.class);

// And a service that uses this repository
MyService service = new MyService(mockRepository);

// When we call the saveUser method, it should invoke save on the repository
service.saveUser(new User());

// Now, let's verify the interaction
verify(mockRepository, times(1)).save(any(User.class));
```

Example of verifyNoMoreInteractions()

```
// After our service call...
service.saveUser(new User());

// We verify the expected call
verify(mockRepository).save(any(User.class));

// And then ensure nothing else was done
verifyNoMoreInteractions(mockRepository);
```

Example of verifyZeroInteractions()

```
// If our user shouldn't be saved...
User newUser = new User();

// And the service shouldn't interact with the repository
verifyZeroInteractions(mockRepository);
```

Common Pitfalls and Best Practices