

# Testing Static Classes: Approaches and Challenges



# Mockito Static Methods Mocking

# Using mockStatic to Mock a Static Method

```
// Start of the try-with-resources block to scope the mock's lifecycle
try (MockedStatic<UtilityClass> mockedStatic = mockStatic(UtilityClass.class)) {

    // Here we stub the static method staticMethod of UtilityClass.
    // We define it to return "Mocked Response" when called with "Expected Input".
    mockedStatic.when(() -> UtilityClass.staticMethod("Expected Input")).thenReturn("Mocked Response");

    // We call the static method as we would in our application code.
    String result = UtilityClass.staticMethod("Expected Input");

    // Now we assert that the result is as expected after our stubbing.
    assertEquals("Mocked Response", result);

    // This is where we verify that our static method was called exactly once.
    mockedStatic.verify(() -> UtilityClass.staticMethod("Expected Input"), times(1));
}
```

# Verifying Static Method Calls with Arguments

```
// Again, we use a try-with-resources block to ensure the mock is closeable and scoped.
try (MockedStatic<UtilityClass> mockedStatic = mockStatic(UtilityClass.class)) {

    // We don't stub any behavior here; we'll simply verify the call.
    UtilityClass.staticMethod("Any Input");

    // Verification with argument matchers. This verifies that our static method
    // was called with any string, showing the flexibility of Mockito's matchers.
    mockedStatic.verify(() -> UtilityClass.staticMethod(anyString()));
}
```

# PowerMock Integration for Stubbing Static Methods

# Annotations Setup for PowerMock

```
import org.junit.runner.RunWith;
import org.powermock.core.classloader.annotations.PrepareForTest;
import org.powermock.modules.junit4.PowerMockRunner;

@RunWith(PowerMockRunner.class) // Instructs JUnit to use PowerMock's runner
@PrepareForTest({YourClassWithStaticMethods.class}) // Prepares the class with static methods for testing
public class YourTestClass {
    // Your test methods will go here
}
```

# Mocking a Static Method with PowerMock

```
import org.powermock.api.mockito.PowerMockito;
import static org.powermock.api.mockito.PowerMockito.mockStatic;

public class YourTestClass {
    @Test
    public void testStaticMethodMocking() {
        mockStatic(YourClassWithStaticMethods.class); // Tells PowerMock to mock all static methods of this class
        PowerMockito.when(YourClassWithStaticMethods.staticMethodName("expectedInput"))
            .thenReturn("mockedOutput"); // Stubbing the static method

        // ... perform your test logic
    }
}
```

# Verification of Static Method Calls with PowerMock

```
import org.powermock.api.mockito.PowerMockito;  
import static org.powermock.api.mockito.PowerMockito.verifyStatic;  
  
public class YourTestClass {  
    @Test  
    public void testStaticMethodVerification() {  
        mockStatic(YourClassWithStaticMethods.class); // Mock the static methods first  
  
        // ... call some method that should, in turn, call the static method  
  
        // Verify that the static method was called with the right argument  
        verifyStatic(YourClassWithStaticMethods.class);  
        YourClassWithStaticMethods.staticMethodName("expectedInput");  
    }  
}
```