

Mocks

Ao escrever testes, às vezes precisamos simular partes de nosso sistema para tornar os testes possíveis e os resultados reproduzíveis. Mocks são imitações ou unidades falsas que simulam o comportamento de unidades reais

Nos testes unitários, podemos simular o comportamento de objetos reais complexos. Principalmente, quando estes objetos são difíceis de serem incorporados nos testes de unidade.

A utilização de mocks elimina a necessidade de utilizar alternativas mais lentas. Por exemplo, banco de dados, que precisam ser inicializados antes da execução dos testes e fornece um feedback muito mais rápido ao desenvolvedor.

- **Comparação de Mocks com Stubs**

Stubs

São objetos criados para facilitar os testes dando respostas pré-determinadas e fazendo operações que fornecem informações adicionais do uso do método em teste. É mais importante dar facilidades para o teste acontecer com tranquilidade do que executar o teste, tanto que ele não tem como função fazer o teste falhar. Usado para substituir estados.

Situações para uso

Estabelecido que ambos servem para substituir componentes reais (eles são "dublês" destes componentes) durante os testes, e entendida a diferença entre eles, fica nítido quando usar um e quando usar outro:

Use stub para testar se um código, dada uma determinada entrada (respostas prontas dos métodos do stub), produz determinada saída.

Use mock para testar se um código se comporta da maneira esperada no que tange a interações com o componente que o mock está substituindo.

Diferença

Usamos o Mock quando queremos saber se uma função vai ser chamada corretamente, quantas vezes ela vai ser chamada, se os parâmetros esperados são os corretos, já o Stub vai nos dizer se o resultado do código retorna de acordo com os parâmetros passados, se retorna sucesso, erro ou exceção por exemplo, é previsível.

Um Mock é mais complexo que um Stub.