2023-04-26-be-mountain

Research Week 12

Cody Schliebe

Mocking a Class

Using JUnit and Mockito, a developer is able to run tests on methods or classes using 'mocking' and 'stubbing'. These terms are often used interchangeably, but they do have distinct differences. A mock, for example, is a fake class that can be examined after a test is run to see whether a method was called or how many times it was called. This could be useful if your method, for instance, sends data to another service when a certain condition is met. A stub, on the other hand, comes programmed with expected return values. You have full control over what is input into a method without the need for any external connections or dependencies. It seems like the mocking we did in the week 12 coding assignment was more stubbing than mocking.

Mocking (or stubbing) is useful when you need to test certain situations in your code without actually running the code. If you have a live application connected to a database that can interact with the public, you don't want to test something and have it send or display something to the outside world that you don't want it to. Mocking allows you to test one or more methods in an isolated, temporary environment that will not affect your database or other external resource. [1]

Dangers of Hardcoding Your Credentials

When a developer is coding a brand-new application, it's easy for them to just put the credentials into the code as they're the only ones who see it. But as the application grows in scope and complexity, a second developer may be brought in to help, and before long you have a full team working on the application. Some of them need to be able to log into the database it's connected to, some don't. Some of them will work on the application for the life of it, and some are temporary hires that will perform one task and leave.

Hardcoding your credentials in plain-text is a fine practice when building and testing the db connection, but once it's established, the credentials should be secured. What if one of the temps, in an effort to gain some street cred, posts the credentials on a dark web forum? Or what if one of your long-term developers quits and goes to work for a competitor? In both situations, your code and the contents of your database could now be open to any number of people that you don't want it open to. Sensitive information could be released to the public or used for illegal purposes, or you could be open to blackmail or penetration attacks. [2]

1 https://semaphoreci.com/community/tutorials/stubbing-and-mocking-with-mockito-2-and-junit

2 https://medium.com/twodigits/keep-passwords-out-of-source-code-why-and-how-e84f9004815a