How should I test a method that populates a list from a DataReader?

Asked 16 years, 3 months ago Modified 16 years, 3 months ago Viewed 851 times

5

So I'm working on some legacy code that's heavy on the manual database operations. I'm trying to maintain some semblance of quality here, so I'm going TDD as much as possible.

The code I'm working on needs to populate, let's say a `List<Foo>` from a DataReader that returns all the fields required for a functioning Foo. However, if I want to verify that the code in fact returns one list item per one database row, I'm writing test code that looks something like this:

```
Expect.Call(reader.Read()).Return(true);
Expect.Call(reader["foo_id"]).Return((long) 1);
// ....
Expect.Call(reader.Read()).Return(true);
Expect.Call(reader["foo_id"]).Return((long) 2);
// ....
Expect.Call(reader.Read()).Return(false);
```

Which is rather tedious and rather easily broken, too.

How should I be approaching this issue so that the result won't be a huge mess of brittle tests?

Btw I'm currently using Rhino.Mocks for this, but I can change it if the result is convincing enough. Just as long as the alternative isn't TypeMock, because their EULA was a bit too scary for my tastes last I checked.

Edit: I'm also currently limited to C# 2.

c# unit-testing tdd mocking

Share

Improve this question

Follow

edited Aug 31, 2008 at 7:42

Mark Biek

150k ● 54 ● 158 ● 201

asked Aug 27, 2008 at 12:29

Rytmis

32k ● 8 ● 61 ● 69

6 Answers

Sorted by: Highest score (default)

1

To make this less tedious, you will need to encapsulate/refactor the mapping between the DataReader and the Object you hold in the list. There is quite of few steps to encapsulate that logic out. If that is the road you want to take, I can post code for you. I am just not sure how practical it would be to post the code here on StackOverflow, but I can give it a shot to keep it concise and to the point. Otherwise, you are stuck with the tedious task of repeating each expectation on the index accessor for the reader. The encapsulation process will also get rid of the strings and make those strings more reusable through your tests.

Also, I am not sure at this point how much you want to make the existing code more testable. Since this is legacy code that wasn't built with testing in mind.

Share
Improve this answer
Follow

edited Aug 27, 2008 at 17:34

answered Aug 27, 2008 at 17:22
Dale Ragan
18.3k ● 3 ● 55 ● 71

1

I thought about posting some code and then I remembered about JP Boodhoo's Nothin But .NET course. He has a sample project that he is sharing that was created during one of his classes. The project is hosted on Google Code and it is a nice resource. I am sure it has some nice tips for you to use and give you ideas on how to refactor the mapping. The whole project was built with TDD.

Share
Improve this answer
Follow

edited Aug 29, 2008 at 2:52

answered Aug 27, 2008 at 18:42
Dale Ragan
18.3k ● 3 ● 55 ● 71

0

You can put the Foo instances in a list and compare the objects with what you read:

```
var arrFoos = new Foos[]{...}; // what you expect
var expectedFoos = new List<Foo>(arrFoos); // make a list from the hardcoded
array of expected Foos
var readerResult = ReadEntireList(reader); // read everything from reader and
put in List<Foo>
Expect.ContainSameFoos(expectedFoos, readerResult); // compare the two lists
```

Share Improve this answer Follow

answered Aug 27, 2008 at 13:15
kokos
43.6k ● 5 ● 37 ● 32

Kokos,

0

Couple of things wrong there. First, doing it that way means I have to construct the Foos first, then feed their values to the mock reader which does nothing to *reduce* the amount of code I'm writing. Second, if the values pass through the reader, the Foos won't be the *same* Foos (reference equality). They might be *equal*, but even that's assuming too much of the Foo class that I don't dare touch at this point.

Share Improve this answer Follow

answered Aug 27, 2008 at 17:00
Rytmis
32k ● 8 ● 61 ● 69

0

Just to clarify, you want to be able to test your call into SQL Server returned some data, or that if you had some data you could map it back into the model?

If you want to test your call into SQL returned some data checkout my answer found here

Share
Improve this answer
Follow

edited May 23, 2017 at 11:48
Community Bot
1 ● 1

answered Aug 27, 2008 at 17:42
Toran Billups
27.4k ● 41 ● 158 ● 272

0

@Toran: What I'm testing is the programmatic mapping from data returned from the database to quote-unquote domain model. Hence I want to mock out the database connection. For the other kind of test, I'd go for all-out integration testing.

@Dale: I guess you nailed it pretty well there, and I was afraid that might be the case. If you've got pointers to any articles or suchlike where someone has done the dirty job and decomposed it into more easily digestible steps, I'd appreciate it. Code samples wouldn't hurt either. I do have a clue on how to approach that problem, but before I actually dare do that, I'm going to need to get other things done, and if testing that will require tedious mocking, then that's what I'll do.

Share Improve this answer Follow

answered Aug 27, 2008 at 18:01
Rytmis
32k ● 8 ● 61 ● 69