In a layered design with a separate DataAccess layer in .NET where should connection string be managed?

Asked 16 years, 2 months ago Modified 15 years, 5 months ago Viewed 1k times

3

There is a long running habit here where I work that the connection string lives in the web.config, a Sql Connection object is instantiated in a using block with that connection string and passed to the DataObjects constructor (via a CreateInstance Method as the constructor is private). Something like this:

```
using(SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].Connecti
{
    DataObject foo = DataObject.CreateInstance(conn);
    foo.someProperty = "some value";
    foo.Insert();
}
```

This all smells to me.. I don't know. Shouldn't the DataLayer class library be responsible for Connection objects and Connection strings? I'd be grateful to know what others are doing or any good online articles about these kind of design decisions.

Consider that the projects we work on are always Sql Server backends and that is extremely unlikely to change. So factory and provider pattern is not what I'm after. It's more about where responsibility lies and where config settings should be managed for data layer operation.

c# asp.net database

Share

Improve this question

Follow

edited Jul 21, 2009 at 16:40

Rich Seller

84k ● 23 ● 175 ● 179

asked Oct 8, 2008 at 9:14

hollystyles

5,029 ● 2 ● 37 ● 38

3 Answers

Sorted by: Highest score (default)

I like to code the classes in my data access layer so that they have one constructor that takes an IDbConnection as a parameter, and another that takes a (connection) string.

4

That way the calling code can either construct its own SqlConnection and pass it in (handy for integration tests), mock an IDbConnection and pass that in (handy for unit tests) or read a connection string from a configuration file (eg web.config) and pass that in.

Share Improve this answer Follow

answered Oct 8, 2008 at 9:20
Matt Hamilton
204k ● 61 ● 392 ● 321

In this case the main/startup module has to refer the DAL as well, do you still suggest the same after these years or new patterns/approaches have emerged? – bjan Jun 19, 2018 at 4:37

@bjan Nowadays I just use EF and let it read the connection string from the web.config. The DAL is the web API as far as I care. – Matt Hamilton Jun 20, 2018 at 6:11

1

Hm, I think I agree that the datalayer should be responsible for managing such connection strings so the higher layers don't need to worry about this. However, I do not think that the SQLConnection should worry where the connection string comes from.

I think, I would have a datalayer which provides certain DataInputs, that is, things that take a condition and return DataObjects. Such a DataInput now knows "hey, this DataObjects are stored in THAT Database, and using the Configurations, I can use some connection-string to get an SQL-Connection from over there.

That way you have encapsulated the entire process of "How and where do the data objects come from?" and the internals of the datalayer can still be tested properly. (And, as a side effect, you can easily use different databases, or even multiple different databases at the same time. Such flexibility that just pops up is a good sign(tm))

Share Improve this answer Follow

answered Oct 8, 2008 at 9:27
Tetha
4,904 ● 1 ● 18 ● 17

0

this being a "smell" is relative. if you are pretty sure about coupling this particular piece of code to SQL Server and a web.config connection string entry then it's perfectly OK. if you are not into this kind of coupling, I agree that it is a code smell and is undesirable.

Share Improve this answer Follow

answered Oct 8, 2008 at 9:21

cruizer

6,151 ● 2 ● 29 ● 34