

Re: Implementing a DataReader : where to store the result from DB ?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.adonet/2004-04/1270.html>

From: William Ryan eMVP (*dotnetguru_at_comcast.nospam.net*)

Date: 04/16/04

Date: Fri, 16 Apr 2004 08:54:14 -0400

Hi Craig:

> *I was wondering how efficient this could be ... it really doesn't seem to
> be. I mean, an array of array of objects seems to lead to unnecessary type
> casts.*

Sure, and Array or Objects would lead to many type casts but unless you know the types in advance, I don't see any way around it.

> *Also in other .Net providers, I've seen that a DataTable is used to store
> the values retrived from the DB.*

Not necessarily. Depends on what you use. A dataset is also used and sometime nothing is used, for instance, cmd.ExecuteScalar, cmd.ExecuteReader

Datatables are headless and don't have anything to do with a provider nor do they care. If a provider chooses to use one it can, but it's not obligatory.

> *Moreover, it seems that other .Net providers, does not store the data in
> anyplace and instead they keep a connection open to the database to
retrieve
> the data on demand.*

No, not at all. A dataReader needs a persisent open connections, but when you fill a DataSet/ DataTable with an Adapter, you defitely don't need an open connection (or want an open connection) after you are done filling and/or updating your table. The period between fill and update should definitely have the connection closed barring some compelling reason to do otherwise.

Remember that ADO.NET has a Connected (cmd.Executexxx) component and a Disconnected Component (dataadapter.fill)

microsoft.public.dotnet.framework.adonet: Re: Implementing a DataReader : where to store the result from DB ?

- > *So, I see 3 alternatives here, and I'm looking for a general advise.*
- > *Which is the best or more commonly used approach to store the result*
coming
- > *from the database in an IDataReader?*

Yes, it's true that DataReaders are used to fill datatables, but that's not the same as using a DataReader in general. You can iterate a datareader and load a custom collection just as easily as a datatable, and if that's how you want to implement your class, you are free to do so. The distinction and 'commonly used approach' really depends on the architecture of the app and the requirements. However the connected mode vs. disconnected mode is generally used b/c it's the one MS implemented and works pretty well in most cases.

HTH,

Bill

"Craig Kenisston" <craigkenisston@hotmail.com> wrote in message
news:OOoetC1IEHA.520@tk2msftngp13.phx.gbl...

- >
- > *Hi,*
- >
- > *I'm working in the implementation of a little .net provider for a*
proprietary
- > *database format.*
- > *I've studied the .Net provider sample that is in the vs.net help, as well*
as
- > *other providers a bit.*
- >
- > *There is a class in the sample to store the result from the database :*
- >
- >
- > *public class MaxResultSet*
- > *{*
- > *public struct MetaData*
- > *{*
- > *public string name;*
- > *public Type type;*
- > *public int maxSize;*
- > *}*
- >
- > *public int recordsAffected;*
- > *public MetaData[] metaData;*
- > *public object[,] data;*
- > *}*
- >
- >
- >

Re: Implementing a DataReader : where to store the result from DB ?

microsoft.public.dotnet.framework.adonet: Re: Implementing a DataReader : where to store the result from DB ?

>

>

> *Thanks in advance.*

>

>