

Re: Finally which ORM tool?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-10/msg01917.html>

- *From:* "Frans Bouma [C# MVP]" <perseus.usenetNOSPAM@xxxxxxxxxx>
 - *Date:* Sat, 13 Oct 2007 02:43:56 -0700
-

Jon Skeet [C# MVP] wrote:

James Crosswell <james@xxxxxxxxxxxxxxxx> wrote:

Jon Skeet [C# MVP] wrote:

The decision to be session-based or not is pretty fundamental to an ORM. If you don't want to use sessions, use an ORM which isn't session-based, such as LLBLGen – but please don't try to suggest that no ORMs should use sessions and manage aliasing and identity issues just because they happen not to be issues for you.

I think you're intentionally mashing my words up here. I never said any such thing. I said that ORMs should not REQUIRE us to use sessions in order to persist and load objects.

What I did NOT say (and which you suggested I did) is that ORMs shouldn't provide session capabilities. I think they're just fine and dandy and a nice feature. However I see no reason why they shouldn't be OPTIONAL and I've provided some pretty good examples why they should be...

Let me rephrase then:

It is easier to write an ORM system which has a rich session system if you force sessions to be used everywhere: it means that every layer of the ORM framework code can rely on sessions being available.

Using a session makes an o/r mapper easier to write, as you can just pick Scott Ambler's design and start typing code. The thing is that as

Re: Finally which ORM tool?

soon as you create a distributed system, you run into problems which have to be solved by the user of the framework and not the framework itself. Re-attaching entity objects from a previous session is 'ok' but you won't know which fields have changed.

Also, if you use a distributed system, and you want on the _client_ to have unique objects, you're in for a lot of pain, as you don't have that option, you have to write your own uniquing code on the client (as the session lives on the server!).

With in-memory context objects you solve this, and with a session-less design in the o/r mapper you solve the re-attach crap.

So why people still find it necessary to design an o/r mapper around a session is beyond me. Sure, it's more work to do it session-less, but the point of a framework isn't the joy it gives to the author of it, but the joy it gives to the user of it.

If you don't want to use sessions, I would suggest you use an ORM which makes them optional, understanding that there may well be session-based features which aren't available because the sessions aren't pervasive in that system.

I can't name one feature which isn't possible in these kind of frameworks to be honest.

You need to distinguish between the idea of a per-request session and a "per user transaction" session. Again, Hibernate in Action deals with this very well.

Unless by per user transaction you mean a one thread per client model for the server threads then I don't see how this makes a difference.

I mean a session which is potentially split over multiple client/server requests.

that means the service is stateful. Isn't that a problem waiting to happen, sooner or later?>

never mind services, let's use a simple example of asp.net and postbacks. Exact the same problem.

Re: Finally which ORM tool?

Re: Finally which ORM tool?

When one framework forces you into a particular and irritating mindset, it can be seen as a mistake. When several (including the most commonly used ones) do so, it's worth taking a step back and looking for the underlying reasons. The experts are best placed to do that, and I don't claim to be an expert in ORM in general.

(As another couple of data points, LINQ to SQL has the same session bias, and I believe that the ADO.NET Entity Framework does too.)

That's also why they have teh add/detach hell already looming on the horizon (Linq to sql will throw exceptions in these cases when attaching graphs) and together with deferred execution of linq queries, it will give a lot of problems for users who expect A but run into snag B.

FB

--

Lead developer of LLBLGen Pro, the productive O/R mapper for .NET
LLBLGen Pro website: <http://www.llblgen.com>
My .NET blog: <http://weblogs.asp.net/fbouma>
Microsoft MVP (C#)

.