

Re: Fundamental ASP.NET question

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2006-07/msg00779.html>

- *From:* "Maneesh" <maneeshkhare@xxxxxxxx>
 - *Date:* 7 Jul 2006 13:25:32 -0700
-

Erik, Thanks for your reply.

The reason is this. I usually try to have a backend class library providing me utility functions/DAL functionality to be used by the Web Application.

Now, I want to store the connection string in the Web.config file, and then set some static property of the backend class (where I cannot get a handle to the Session variable) with it, so that I can re-use it throughout the application, whenever I want to make a connection, instead of passing it to a backend method each time.

I'm not sure if this is a good practice, or what would be a good practice in such a case, i.e. when you want to have a class library at the back, but want to use some data from Web.config throughout the lifetime of the Web Application.

Do let me know if you can shed some light on this.

regards,
Maneesh

Erik Funkenbusch wrote:

On 7 Jul 2006 11:49:16 -0700, maneeshkhare@xxxxxxxx wrote:

I have a doubt regarding the architecture, and working of the ASP.NET framework. I haven't been able to satisfy myself with any answer.

I do understand that for each request for a resource (let's talk page), we have 1 HttpApplication object from a pool that is managed by HttpApplicationFactory. My question is this. After all the objects that are created to service 1 request, is the same dll referenced for 1 page.

Let's say 3 people are trying to access Page1.aspx. Each of their requests will be handled by 3 separate HttpApplication objects, but will all their requests be delegated to 1 dll corresponding to Page1.aspx, or 3 separate dlls.

The reason I'm asking is because if there is only 1 dll being used, I will think twice before using any static properties/fields, to avoid synchronization issues, when the properties are read/write.

Re: Fundamental ASP.NET question

Thanks in advance.
Maneesh

There is seldom any valid reason to use static data anyways. Why would you even want to? Each HTTP Request is it's own universe. It may have data saved via Session or page variables that is carried over from one request to the next, but it must be a self contained unit in and of itself because requests are stateless.

This means that you can't expect any data in a static variable to exist after a round trip. It might, or it might not. It