

Re: session advice

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005-07/msg00018.html>

- *From:* "Robbe Morris [C# MVP]" <info@xxxxxxxxxxxxxxxx>
 - *Date:* Thu, 30 Jun 2005 20:17:25 -0400
-

The view state issue surprises me. As long as your machine key is valid across the farm, you should be in good shape.

In Windows 2003 Server IIS 6, your app is being recycled periodically. InProc session runs "in process". If you want to avoid this altogether, you'll need to implement StateServer on your local server or go with SQL Server Session State management.

Read more:

<http://www.eggheadcafe.com/articles/20021016.asp>

--

2004 and 2005 Microsoft MVP C#
Robbe Morris
<http://www.masterado.net>

Earn \$\$\$ money answering .NET Framework messageboard posts at EggHeadCafe.com.
<http://www.eggheadcafe.com/forums/merit.asp>

"No One" <aintnoway@xxxxxxxxxxxxxxxx> wrote in message news:pevvo2-mpl.ln1@xxxxxxxxxxxxxxxx
> Then why does Win2003 Server and IIS 6 suddenly recycle Sessions,
> invalidate view state and totally screw up my application? It didn't do
> this before. Some service pack seems to have broken it. Reporting
> Services is even hosed, even after an uninstall/reinstall round. And, of
> course, the client is all pissed as hell.
>
> Kevin Spencer wrote:
>
>>>Sessions seem to be a mystery and very easy to break.
>>
>>

Re: session advice

>> A mystery is only a mystery until it is understood. There is nothing
>> really mysterious about Session at all. Perhaps I can clear up the
>> mystery for you.

>>

>> ASP.Net runs in the context of HTTP messaging, generally between a
>> browser client and a web server. HTTP is stateless. This means that there
>> is no state between requests. The server receives a request for a
>> resource (an ASP.Net page), processes the request, and sends the
>> response. It immediately forgets everything it has just done. The browser
>> loads one HTML document into memory at a time, and forgets the last
>> document that was loaded upon reloading. The only memory on the browser
>> side is the history of the URLs it has visited, the cached documents it
>> stores on the machine, and text cookies, which are stored by domain.

>>

>> ASP.net is an ISAPI (Internet Server Application Programming Interface)
>> application that runs on the web server. An application has memory for
>> the lifetime of the application. This is very important. The ASP.net
>> application is started with the first request for an ASP.Net page in that
>> application, and lives until either the application is restarted, or
>> until 20 minutes (by default) after the last request it receives. So, for
>> this length of time, a memory space exists on the server in which an
>> individual ASP.Net application can store and retrieve data. Hence,
>> ASP.Net has an Application collection and an Application cache that can
>> be used to store Application-wide scoped data.

>>

>> When a Request for a resource is recieved on the server, the browser
>> sends headers with the Request. These headers contain helpful information
>> for the server, including the IP address (the "return address") of the
>> client that made the Request, as well as any Cookies that the browser may
>> hold for that domain or application. So, the ASP.Net ISAPI has event
>> handlers that fire, for example, the first time it receives a request
>> from a browser it doesn't have a Session yet for. If the browser doesn't
>> send a Session Cookie, the ASP.net ISAPI creates a memory cache called
>> "Session" that belongs to that client browser. It also sends a Session
>> Cookie ("Session Cookie" is a term that means a Cookie that is not stored
>> in the file system of the client machine, but lasts for the time that the
>> browser is requesting documents from the same domain) to the browser. The
>> Session Cookie is a unique identifier that identifies which server
>> Session belongs to that client.

>>

>> By this mechanism, the ASP.Net application creates its own form of
>> Session-specific state, a memory space that can be used across separate
>> Page Requests, and lasts for the lifetime of the client Session.

>>

>> Now, how does the ASP.Net ISAPI know when the Session has ended? This is
>> a good question, since every Request happens "in a vacuum." The browser
>> has no way of knowing which page the user browses to will be the last
>> page visited. It can't tell the server therefore. So, the Session has a
>> timer, which by default is set to 20 minutes. Every time the browser
>> sends another Request, the timer is reset to 0. If it reaches 20 minutes,
>> the Session memory for that client is cleared up, ending the Session on

>> the server.

>>

.

-
- Prev by Date: ***Re: DataBind Problem***
 - Next by Date: ***Re: COM object with CLSID {...} is either not valid or not registered***
 - Previous by thread: ***Re: DataBind Problem***
 - Next by thread: ***Re: Invalid Index in paging***
 - Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***