

## Re: Session State vs. What?

**Source:**

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2005-07/0230.html>

---

**From:** Joe Kaplan \((MVP - ADSI)\) ([joseph.e.kaplan\\_at\\_remove\\_this.accenture.com](mailto:joseph.e.kaplan_at_remove_this.accenture.com))

**Date:** 07/21/05

Date: Thu, 21 Jul 2005 13:42:41 -0500

In practice, it looks a lot like session:

```
object o = Context.Cache["key"];  
Context.Cache.Add....
```

etc.

You can get to it from the HttpContext or via the Page class which exposes it as a property.

Just remember that the semantics are different. Just because you put something in the cache doesn't mean it will be there when you go to get it out. The cache can eject things if they get too old or memory runs low. Thus, if it is not there, you may need to recreate the object. That is why it is a cache. The data is stored in process in memory only.

Joe K.

"David Lozzi" <dlozzi@(remove\_this)delphi-ts.com> wrote in message news:Oj39JFijFHA.2484@TK2MSFTNGP15.phx.gbl...

> *Interesting.... If I were to use the cache scenario, what would that look like? How do I simply load it and read from it? I've never used cache before.*

>

> *Thanks!*

>

> *David Lozzi*

>

>

> *"Joe Kaplan (MVP - ADSI)" <joseph.e.kaplan@remove\_this.accenture.com> wrote in message news:e%23EowqhfFHA.3540@TK2MSFTNGP14.phx.gbl...*

>> *Yep, this is a good point. That is another good reason to use cache in this instance (which was what I was trying to suggest as his first choice approach).*

>>

>> *Sometimes session is the appropriate way to solve a state problem, but oftentimes it is not, especially given the host of other approaches*

>> available in ASP.NET.  
>>  
>> Thanks,  
>>  
>> Joe K.  
>>  
>> "Dominick Baier [DevelopMentor]" <[dbaier@pleasepleasenosspamdevelop.com](mailto:dbaier@pleasepleasenosspamdevelop.com)>  
>> wrote in message news:628050632575659638047280@news.microsoft.com...  
>>> Hello Joe,  
>>>  
>>> the thing is that you don't have session state in AuthenticateRequest –  
>>> the SessionStateModule runs after FormsAuthentication...so you could use  
>>> Session but have to find a way to work around that.  
>>>  
>>> I personally don't like Session because it is slow (2–3 round trips per  
>>> page to the session store and you should only enable it on pages where  
>>> you need it, or set it to read only) and there is often a lot of  
>>> confusion about timeouts, especially when combined with FormsAuth.  
>>>  
>>> I would use the cache, and program defensively and just check everytime  
>>> for an invalidated cache.  
>>>  
>>> -----  
>>> Dominick Baier – DevelopMentor  
>>> <http://www.leastprivilege.com>  
>>>  
>>>> In this case, you can use Session state (still works in .NET and has  
>>>> more options such as out of process) or the Cache.  
>>>>  
>>>> The cache has the advantage of being very fast. The idea with the  
>>>> cache is that you only use it for cacheable data that can be  
>>>> regenerated if it is not in the cache. You would not use it for  
>>>> critical data that the user has input as part of an ongoing process or  
>>>> something.  
>>>>  
>>>> So in your case, you might generate an IPrincipal for the user on  
>>>> their initial login and then put it in the cache. On subsequent  
>>>> logins, you would determine their identity and try to retrieve the  
>>>> IPrincipal from the cache based on the user name. If it is in the  
>>>> cache, then you just use it, if not, you regenerate it from your data  
>>>> source, add it back to the cache and use it.  
>>>>  
>>>> You could also do this with session state just as well, but you  
>>>> probably don't have to in this case unless there is data in this  
>>>> object that you really need for the whole session that can only be  
>>>> generated on the initial login or something.  
>>>>  
>>>> Those are my thoughts,  
>>>>  
>>>> Joe K.  
>>>>

microsoft.public.dotnet.framework.aspnet.security: Re: Session State vs. What?

>>>> "David Lozzi" <dlozzi@(removethis)delphi-ts.com> wrote in message  
>>>> news:%23Z1RGPgjFHA.1464@TK2MSFTNGP14.phx.gbl...

>>>>

>>>>> Howdy,

>>>>>

>>>>> I just learned how to use Forms Authentication. yeah me! However, it  
>>>>> is quite limited as to how much information you can store and  
>>>>> retrieve from it. For example, in my past ASP and ASP.NET  
>>>>> applications I have used the session state to store a user's IS,  
>>>>> Fullname, security level, email and some other minor items. How is  
>>>>> that to be done without Session state?

>>>>>

>>>>> Here is one possible scenario I thought of: with the Forms Auth.  
>>>>> having the username, query the SQL database everytime a page is  
>>>>> loaded (or whenever necessary) to pull the user's information as  
>>>>> necessary. The bad side to this is that there will be 'unnecessary'  
>>>>> traffic on almost every page. Currently, for example, I do something  
>>>>> like so: If session("SecurityLevel") < 8 then  
>>>>> response.redirect("home.aspx"). Simple enough. With this scenario, I  
>>>>> would have to query the database first then return the security  
>>>>> level. More work for the server, no?

>>>>>

>>>>> Is there another way? Or is session state the best solution? I  
>>>>> remember reading a few posts that stated using the session state was  
>>>>> not a desired function due to overhead on the server, or something  
>>>>> like that. I can imagine the server would work harder querying the  
>>>>> database for a single number over storing a number in the session  
>>>>> state.

>>>>>

>>>>> I am trying to find the best solution moving forward, assuming  
>>>>> Session State is not it. I appreciate all of your input!

>>>>>

>>>>> David Lozzi

>>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>

>>>>