
Subject: A few questions about Skylark sessions
Posted by [Peter](#) on Thu, 27 Dec 2012 19:42:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm trying to understand how sessions work in Skylark. I already asked a few questions some time ago in other part of the forum and the answers were helpful, but I still have some doubts.

1. When does a new session actually start? Suppose I have a basic Skylark application with just one handler which does nothing (empty definition) and go to the corresponding URL. Will this start a new session if no session is active? Or would I need an explicit call to `SessionSet()` or `NewSessionId()`?

2. What's the exact difference between `ClearSession()` and `NewSessionId()`? The former is supposed to clear current session id and variables, but according to manual it retains values in shared variable space. Does it mean that if I have a session variable `.VAR` set to 1 and call `ClearSession()` then `http[".VAR"]` (cast to int) will still be 1?

Basically: what does `ClearSession()` exactly do (what does it clear and what does it preserve)? What's the difference between:

- calling `ClearSession()` alone?
- calling `NewSessionId()` alone?
- calling `ClearSession()` and then `NewSessionId()`?

3. What's the real meaning of `SkylarkSessionConfig::expire`? When I set it to 600 (10 minutes), it should clear the current session after 10 minutes since... when? The moment a session file is saved on server? What's supposed to happen after 10 minutes? A new session is started automatically or not? Is it supposed to work the same in all browsers? In another thread someone said it shouldn't be set to less than 10 minutes. Why is that?

Subject: Re: A few questions about Skylark sessions
Posted by [dolik.rce](#) on Fri, 28 Dec 2012 08:33:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Peter

Peter wrote on Thu, 27 December 2012 20:421. When does a new session actually start? Suppose I have a basic Skylark application with just one handler which does nothing (empty definition) and go to the corresponding URL. Will this start a new session if no session is active? Or would I need an explicit call to `SessionSet()` or `NewSessionId()`? The session cookie is first sent to client (and the session record on the server is first created) when it is first needed. That is when you first use `SessionSet` or `NewSessionId` or when you use `$post_identity()` in `witz` or similar function that calls `SessionSet` internally. Empty handler doesn't require any session information, so no session is created.

Peter wrote on Thu, 27 December 2012 20:422. What's the exact difference between

ClearSession() and NewSessionId()? The former is supposed to clear current session id and variables, but according to manual it retains values in shared variable space. Does it mean that if I have a session variable .VAR set to 1 and call ClearSession() then http[".VAR"] (cast to int) will still be 1?

Basically: what does ClearSession() exactly do (what does it clear and what does it preserve)? What's the difference between:

- calling ClearSession() alone?
- calling NewSessionId() alone?
- calling ClearSession() and then NewSessionId()?

NewSessionId() keeps all the information associated with the session, only changes the key under which it is stored. So it clears nothing, only changes the id. The most common reason for this is to change the id (and cookie value on client side) after log in and log out to prevent session hijacking.

ClearSession() deletes both the session variables and the session id, only thing that is kept is values of the session variables in shared variable space. So you are right about the behavior: if you have session variable .VAR and call ClearSession(), you can still use its value from http[".VAR"], but only for the same request when you called ClearSession. Any subsequent requests will not have the session variables stored, so they won't appear in the shared variable space anymore.

Calling both, ClearSession() and then NewSessionId() is probably not useful in any way. As soon as you ClearSession(), next call to SessionSet() would create new session id anyways.

Peter wrote on Thu, 27 December 2012 20:423. What's the real meaning of SkylarkSessionConfig::expire? When I set it to 600 (10 minutes), it should clear the current session after 10 minutes since... when? The moment a session file is saved on server? What's supposed to happen after 10 minutes? A new session is started automatically or not? Is it supposed to work the same in all browsers? In another thread someone said it shouldn't be set to less than 10 minutes. Why is that?

The true meaning of SkylarkSessionConfig::expire is "The minimal time between requests (in seconds), that the session information is guaranteed to persist". The expiration time is always counted from last request that modified the session. Because expiring the sessions is costly operation, it is highly optimized so it happens in batches from time to time. The exact mechanism (in pseudo code) is:

```
Every 1000 requests do {
    if (now() - last_expiration_check > 10 min) {
        last_expiration_check = now();
        expire_time = now() - SkylarkSessionConfig::expire
        for each session {
            if (session.last_write_time < expire_time){
                delete(session);
            }
        }
    }
}
```

So as you can see the expiration is performed after every 1000th request is served (counted per thread), but only if it is longer then 10minutes after last expiration. That is why I said it is useless

to set the expire value to less than 10 minutes.

After the session is expired, the server will not recognize it, and will behave as if it is a new, unknown user coming. If you try to set any session variables for him, new session will be created.

It should work the same way across browsers, only difference might be in how the browsers are set up to handle cookies. The session cookie is sent as a session cookie, meaning that it should be deleted when the browser is closed. It is possible that some browsers misbehave a bit in this behavior...

I hope I explained it better this time. If there is still something unclear, don't hesitate to ask more. You can also try to reach me on irc/gtalk/icq to ask me in real time.

Best regards,
Honza

Subject: Re: A few questions about Skylark sessions
Posted by [Peter](#) on Mon, 31 Dec 2012 11:33:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

What if I call `NewSessionId()` and want to read the new value of session id in the same request? Since the old value is preserved in the shared variable space until next request, I can't access the new one by `http["@__skylark_session_cookie__"]`.

Subject: Re: A few questions about Skylark sessions
Posted by [dolik.rce](#) on Tue, 01 Jan 2013 14:16:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Peter wrote on Mon, 31 December 2012 12:33: What if I call `NewSessionId()` and want to read the new value of session id in the same request? Since the old value is preserved in the shared variable space until next request, I can't access the new one by `http["@__skylark_session_cookie__"]`.

First of all, `@__skylark_session_cookie__` is not something really related to this. The '@' prefix means it is variable read from cookie, in this case it is the session cookie. So no matter what you do with the session id, it is logical that `@__skylark_session_cookie__` still has the same value, because it shows what came from the cookie. The preservation of variables in shared space as written about in docs actually concerns only the '.' prefixed session variables. There is no other way to read the session id and the fact that you can read it this way is more or less just a coincidence.

In your code you should never need to know the session id, it is just a behind the scenes mechanism. If you need to identify the user or something, just use a session variable or separate cookie (using `SetCookie()` method). It will give you better control without interfering with the Http internals.

Subject: Re: A few questions about Skylark sessions

Posted by [Peter](#) on Mon, 07 Jan 2013 21:17:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

This time I have a general problem and would like to know the best way to solve it.

As before, suppose I have a web portal. All users must log in to gain access to it. There are two types of users: admins and non-admins. Admin is able to delete any user from database or change his/her data. Let's consider the following scenario. We have two users: admin (A) and other user (B, admin or non-admin).

Both are logged in to my portal. At some point A deletes B from database while B is still logged in. What should happen next? The obvious idea is that B should get logged out as soon as he gets deleted from database, otherwise we'd have an inconsistency such that B would be still "logged in" (his session variable ".LOGGED" set to 1/true/etc.) while he'd be physically removed from database.

How can this problem be solved? My only idea is to check at the beginning of every request handler if the user is still in the database. The test would be based on a unique id value - users table primary key. Session variable would be initialized to this unique id as soon as the user logs in. The pseudocode for one of the handlers would be something like this:

```
SKYLARK(Handler, "handler")
{
    if(http[".LOGGED"]) // if user is logged
        if table doesn't contain user with id=http[".ID"] anymore
            ClearSession();
}
```

Is that a correct approach?

What about even trickier case - while B is logged in, A alters some of B's data (login, password or even unique id)? What's the expected behaviour then? Should B remain logged in or not?

Subject: Re: A few questions about Skylark sessions

Posted by [dolik.rce](#) on Mon, 07 Jan 2013 22:24:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi again Peter

Peter wrote on Mon, 07 January 2013 22:17 At some point A deletes B from database while B is

still logged in. What should happen next? The obvious idea is that B should get logged out as soon as he gets deleted from database, otherwise we'd have an inconsistency such that B would be still "logged in" (his session variable ".LOGGED" set to 1/true/etc.) while he'd be physically removed from database.

How can this problem be solved? My only idea is to check at the beginning of every request handler if the user is still in the database. The test would be based on a unique id value - users table primary key. Session variable would be initialized to this unique id as soon as the user logs in. I think it should log the user out as soon as possible just as you say. The trick about his session variables is that you should delete that too. For that you must know his session id, which is the tricky part. If you use a database in your project, I would definitely advise you to use it to store sessions as well (you just have to set up `SkylarkSessionConfig::table` to a name of the table) and then you can easily access all the sessions and find&delete the one corresponding to the deleted user (especially when session format is set to JSON). If you use the default setting for sessions, they are stored in files and those are much harder to cope with, but in theory, if you know the correct session id, you can delete the corresponding file in `SkylarkSessionConfig::dir` and achieve the same result - user being effectively erased. Perhaps you could just store the session id of each user when they log in, so you can retrieve it later in case you need to tamper with his session.

Peter wrote on Mon, 07 January 2013 22:17: What about even trickier case - while B is logged in, A alters some of B's data (login, password or even unique id)? What's the expected behaviour then? Should B remain logged in or not? As a user, I would expect that nothing happens and that I can still use the web as any other logged in user. Changing the unique id is probably not very good idea in any scenario I can imagine, but changing anything else (even the login) should just work as long as you use the unique id to reference the user everywhere or, in other words, if you don't store the changed data anywhere.

Anyway, I think that this might be quite common situation. Perhaps there should be some functions to allow modifications of other users cookies. At least `DeleteCookie(String id)`, to allow logging user off in certain situations. It should be fairly easy to implement, do you want to try?

Best regards,
Honza

Subject: Re: A few questions about Skylark sessions
Posted by [Peter](#) on Wed, 09 Jan 2013 10:57:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

I think currently I'm not advanced enough to add anything to U++ myself. I have another question though. Is it possible to pass Http parameter of Skylark handler by reference to another function called from that handler and use it there?

I'd like to write a wrapper function inside which I'd be able to access shared variable space and session variables:

```
SKYLARK(SomeHandler, "somepath")
```

```
{  
    wrapper(http);  
}
```

```
void wrapper(Http& http)  
{  
    // can I access http passed from SomeHandler here?  
}
```

Is the above safe/guaranteed to work? Can I access variables associated with http object from within wrapper?

Subject: Re: A few questions about Skylark sessions
Posted by [dolik.rce](#) on Wed, 09 Jan 2013 12:43:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Peter wrote on Wed, 09 January 2013 11:57I think currently I'm not advanced enough to add anything to U++ myself That doesn't mean you can't try to do it Challenging tasks are the best learning experience...

Peter wrote on Wed, 09 January 2013 11:57I have another question though. Is it possible to pass Http parameter of Skylark handler by reference to another function called from that handler and use it there?

...

Is the above safe/guaranteed to work? Can I access variables associated with http object from within wrapper?

Yes, it works, there is nothing special about the Http class. I have done this and never had any problem with it. The only thing you need to keep in mind is that you should not store this reference anywhere (e.g. no asynchronous calls that might take too long). It's content is reset after the handler function exits (or it might be destroyed completely, I'm not sure from the top of my head), so it would have quite fatal consequences

Best regards,
Honza

Subject: Re: A few questions about Skylark sessions
Posted by [Peter](#) on Thu, 10 Jan 2013 14:58:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

I feel a bit embarrassed to ask for help again, but once again I encountered a problem. I thought I

finally understood how Skylark session functions worked, but another tiny code snippet proves that I don't.

I have these 3 handlers:

```
SKYLARK(AAA, "aaa")
{
  http.SessionSet(".MY_SESSION_VARIABLE", Value(1));
  http.RenderResult("ChairSupervisor/aaa");
}
```

```
SKYLARK(BBB, "bbb")
{
  http.ClearSession();
  http.RenderResult("ChairSupervisor/bbb");
}
```

```
SKYLARK(CCC, "ccc")
{
  http.RenderResult("ChairSupervisor/ccc");
}
```

and corresponding Witz templates:

aaa.witz:

```
<html><body>$set(<a href=$BBB>BBB</a></body></html>
```

bbb.witz:

```
<html><body>$set(<a href=$CCC>CCC</a></body></html>
```

ccc.witz:

```
<html><body>$set(</body></html>
```

I start from AAA handler. Link on "aaa" leads to "bbb" and link on "bbb" leads to "ccc". Each page displays a list of currently set session variables. As you can see, I define a session variable called ".MY_SESSION_VARIABLE". Since I call ClearSession() in "bbb" I assumed it would be cleared there and not visible on session variables list in "ccc". However, it's still there and still set to 1.

Why didn't ClearSession() in "bbb" destroy the session and .MY_SESSION_VARIABLE is still listed in "ccc"? What did I do wrong this time?

Subject: Re: A few questions about Skylark sessions
Posted by [dolik.rce](#) on Thu, 10 Jan 2013 16:50:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Peter wrote on Thu, 10 January 2013 15:58 Why didn't ClearSession() in "bbb" destroy the session and .MY_SESSION_VARIABLE is still listed in "ccc"? What did I do wrong this time?
You did it right, this is a bug In Http::ClearSession(), there should be session_dirty=true, otherwise the change in the variables is not stored and previous state is loaded in next request. I created an issue in redmine, so it'll be fixed soon.

Best regards,
Honza

Subject: Re: A few questions about Skylark sessions
Posted by [Peter](#) on Tue, 15 Jan 2013 12:51:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

But why doesn't this bug appear in the following simpler case?

```
SKYLARK(AAA, "aaa")
{
  http.SessionSet(".MY_SESSION_VARIABLE", Value(1));
  http.RenderResult("MyApp/aaa");
  http.ClearSession();
}
```

```
SKYLARK(BBB, "bbb")
{
  http.RenderResult("MyApp/bbb");
}
```

aaa.witz:

```
<html><body>$set(<a href=$BBB>BBB</a></body></html>
```

bbb.witz:


```
<html><body>$set()</body></html>
```

If I start from "aaa" and use link to go to "bbb", .MY_SESSION_VARIABLE is not listed on "bbb" anymore as intended.

What's the difference between this and previous case where moving from "bbb" to "ccc" didn't clear session?

Anyway, I hope the bug will be fixed soon (I can see its status is "approved", but it's still not fixed).

Best regards

Piotr
