

Ruby trunk - Feature #3848

Using http basic authentication for FTP with Open URI

09/18/2010 04:02 AM - jlecour (Jérémy Lecour)

Status:	Assigned	
Priority:	Normal	
Assignee:	akr (Akira Tanaka)	
Target version:		
Description		
=begin Hi, I needed to download a file from a FTP server, with OpenURI and I've found that it is not possible to do it with authentication. There is an option in the OpenURI open method to use a username/password but it is not used in FTP connection, the username/password for FTP is always set to anonymous/ <u>nil</u> . I've written a very basic patch to use the option. Is it something that can be included in future releases of Ruby ? Is there a "clean" way to patch OpenURI in the meantime ? Thanks. =end		
Related issues:		
Related to Ruby 1.8 - Bug #435: open-uri.rb 407 Proxy Authentication Required...	Rejected	08/13/2008
Related to Ruby trunk - Feature #4071: support basic auth for Net::HTTP.get r...	Closed	11/19/2010

History

#1 - 09/20/2010 04:47 PM - jlecour (Jérémy Lecour)

=begin
As James Gray pointed out (<http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/32470>), it is already possible to use FTP authentication, by passing the username/password *in* the URL.

It's a very good thing because it solves the original issue, but I guess it's a little confusing.

You can also do HTTP basic authentication in the Net::HTTP wrapper of OpenURI, by passing an option.
And if we look for this particular option in the Net::FTP wrapper and use it, we can also authenticate without encapsulation the username/password in the URL.

In some situations, it also may not be as straightforward as it seems to encapsulates the username/password in the URL. Let's say that the URL is coming from a configuration and the need for authentication is determined from other factors.

Maybe the name of the option (`http_basic_authentication`) is not the best if we intend to use it for HTTP and FTP, but I think it would be good to be able to use the options hash for authentication.
=end

#2 - 09/24/2010 04:09 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

=begin
=end

#3 - 09/27/2010 10:45 AM - meta (mathew murphy)

=begin
On Sat, Sep 18, 2010 at 13:19, James Edward Gray II
james@graysoftinc.com wrote:

OpenURI is just a wrapper around some clever URI parsing and URI seems to support it just fine:

```
require "uri"  
=> true  
u = URI.parse("http://user:pass@server.com/")
```

```
=> #
u.userinfo
=> "user:pass"
u.host
=> "server.com"
```

I feel obliged to point out that this is deprecated by the current generic URI RFC.

RFC3986 section 3.2.1: 'Use of the format "user:password" in the userinfo field is deprecated.'

RFC2396 stated that it was "NOT RECOMMENDED".

RFC1738 didn't allow it at all.

So building systems which store and process username and password in this way is a bad idea.

mathew

--

[URL:http://www.pobox.com/~meta/](http://www.pobox.com/~meta/)

=end

#4 - 09/27/2010 11:01 AM - meta (mathew murphy)

=begin

On Sun, Sep 26, 2010 at 20:57, James Edward Gray II james@graysoftinc.com wrote:

On Sep 26, 2010, at 8:44 PM, mathew wrote:

So building systems which store and process username and password in this way is a bad idea.

I'm not sure that really applies to open-uri's usage of this strategy. It's just a familiar interface, not some attempt to correctly mirror how FTP servers manage logins. Does that make sense?

I agree that open-uri shouldn't care what is passed to it, and should just pass the URI to URI for resolution.

I'm not sure URI should be supporting username and password for http, given the history, but there's also an argument for making the code general and being permissive.

I was mostly pointing out to the person who wanted to user user:pass in URIs that he really shouldn't, even if Ruby lets him.

mathew

--

[URL:http://www.pobox.com/~meta/](http://www.pobox.com/~meta/)

=end

#5 - 09/27/2010 11:23 AM - meta (mathew murphy)

=begin

On Sun, Sep 26, 2010 at 21:07, James Edward Gray II james@graysoftinc.com wrote:

On Sep 26, 2010, at 9:01 PM, mathew wrote:

I was mostly pointing out to the person who wanted to user user:pass in URIs that he really shouldn't, even if Ruby lets him.

OK, but, especially in the case of FTP, they aren't really. They are building a URL, just because that's open-uri's supported interface. Under the hood it picks this apart and does a proper FTP login. So this is just smoke and mirrors and not really any different than doing a normal FTP

login. Do you see what I mean?

Sure. I wouldn't do it that way myself in something I expected to need to keep running long term, though, because you have to worry about correctly URI-escaping the username and password. I'd much rather learn to use the actual FTP or HTTP APIs, it's not like they are particularly difficult.

But yes, if you're writing a quick throwaway 10-line script to solve some problem and don't care too much about people with : or @ in their username or password, fair enough. :-)

mathew

--

[URL:http://www.pobox.com/~meta/](http://www.pobox.com/~meta/)

=end

#6 - 11/09/2012 06:00 PM - mame (Yusuke Endoh)

- *Description updated*

- *Target version set to 2.6*

#7 - 12/25/2017 06:14 PM - naruse (Yui NARUSE)

- *Target version deleted (2.6)*

Files

openuri-ftp-auth.patch	573 Bytes	09/18/2010	jlecour (Jérémy Lecour)
------------------------	-----------	------------	-------------------------