

Ruby master - Bug #12212

Parsing a URI of "mailto:?subject=hi" with `URI.parse` raises an exception

03/24/2016 08:59 AM - madeofcode (Mark Dodwell)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: trunk	Backport: 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN
Description Parsing a URI of "mailto:?subject=hi" with URI.parse raises an exception. I believe this is a valid "mailto" URI, as defined in RFC6068. Behavior observed: <pre>>> URI.parse("mailto:?subject=hi") NoMethodError: undefined method `split' for nil:NilClass</pre> Behavior expected: <pre>>> uri = URI.parse("mailto:?subject=hi") => #<URI::MailTo URL:mailto:?subject=hi> >> uri.to => "" >> uri.headers => [{"subject", "hi"}]</pre> Reference from RFC6068 that defines the "to" element as optional: mailtoURI = "mailto:" [to] [hfields] https://tools.ietf.org/html/rfc6068#section-2 Patch attached.	

History

#1 - 03/24/2016 09:42 AM - duerst (Martin Dürst)

Mark Dodwell wrote:

Parsing a URI of "mailto:?subject=hi" with URI.parse raises an exception. I believe this is a valid "mailto" URI, as defined in RFC6068.

As one of the co-authors of RFC 6068, I have to agree with you. See the first syntax production in <https://tools.ietf.org/html/rfc6068#section-2>, where the square brackets around 'to' in
mailtoURI = "mailto:" [to] [hfields]
indicate that the 'to' part can be omitted. This rule is exactly the same as in RFC 2368 (see <https://tools.ietf.org/html/rfc2368#section-2>), the predecessor of RFC 6068.

Although most 'mailto' URIs will have a 'to' part, it can clearly be useful to have a 'mailto' URI without a 'to' part, e.g. if only cc: and bcc: addressees are present, or for some sample emails where e.g. 'subject' and body are prepared, but each user has to enter the addressees themselves.

#2 - 06/26/2019 10:50 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

I can't replicate the NoMethodError you are receiving on any ruby version:

```
$ ruby18 -ruri -e 'p URI.parse("mailto:?subject=hi")'
#<URI::MailTo:0xab543e6f5c0 URL:mailto:?subject=hi>
$ ruby19 -ruri -e 'p URI.parse("mailto:?subject=hi")'
#<URI::MailTo:0x000ccbdl17f98 URL:mailto:?subject=hi>
```

```
$ ruby20 -ruri -e 'p URI.parse("mailto:?subject=hi")'  
#<URI::MailTo:0x000af32a635038 URL:mailto:?subject=hi>  
$ ruby21 -ruri -e 'p URI.parse("mailto:?subject=hi")'  
#<URI::MailTo:0x000054409bb308 URL:mailto:?subject=hi>  
$ ruby22 -ruri -e 'p URI.parse("mailto:?subject=hi")'  
#<URI::MailTo mailto:?subject=hi>  
$ ruby23 -ruri -e 'p URI.parse("mailto:?subject=hi")'  
#<URI::MailTo mailto:?subject=hi>  
$ ruby24 -ruri -e 'p URI.parse("mailto:?subject=hi")'  
#<URI::MailTo mailto:?subject=hi>  
$ ruby25 -ruri -e 'p URI.parse("mailto:?subject=hi")'  
#<URI::MailTo mailto:?subject=hi>  
$ ruby26 -ruri -e 'p URI.parse("mailto:?subject=hi")'  
#<URI::MailTo mailto:?subject=hi>
```

In the patch, the regexp you are trying to skip (`^(A(?:[^\@,;]+@[^\@,;]+(?:\z[;:]))*\z/)`) already matches the empty string, and skipping the conditional would prevent a `InvalidComponentError`, it wouldn't affect the `NoMethodError` you received.

Files

fix-uri-mailto-parsing-blank-to-with-query.patch	1.52 KB	03/24/2016	madeofcode (Mark Dodwell)
------------------------------------------------------------------	---------	------------	---------------------------