Ruby - Feature #15517

Net::HTTP not recognizing valid UTF-8

01/08/2019 04:13 PM - cohen (Cohen Carlisle)

Status:	Closed
Priority:	Normal
Assignee:	naruse (Yui NARUSE)
Target version:	

Description

I created a case at https://github.com/Cohen-Carlisle/utf8app that shows Net::HTTP labeling a response body as ASCII-8BIT encoded because it contains a non-ascii character (specifically, the double prime symbol: ″), but recognizing ascii-only strings as UTF-8 encoded. The example is live on heroku but because it's a free dyno, it will go to sleep and take a while to start up the first time it is hit after a while.

As explained there, I would expect response body strings with the double prime symbol to still have an encoding of UTF-8 since they are valid UTF-8.

The README from the repo (which shows the behavior) is reproduced below:

The purpose of this app is to demonstrate unexpected behavior in Ruby's net/http library. Valid UTF-8 response bodies are encoded as ASCII-8BIT, which apparently means Ruby is treating them as pure binary data, even when Content-Type headers label the body as UTF-8.

In the example below, I would expect the response body to have UTF-8 encoding. Especially because when I copy and paste the body into a new string literal in my console, that string is UTF-8 encoded.

```
require 'net/http'
uri = URI('https://utf8app.herokuapp.com')
uri.path = '/utf8/example'
res = Net::HTTP.get_response(uri)
res['Content-Type']
# => "text/plain; charset=utf-8"
puts res.body
# The symbol for the inch unit of measurement is ″.
res.body.encoding
# => #<Encoding:ASCII-8BIT>
res.body.ascii_only?
# => false
'The symbol for the inch unit of measurement is ″.'.encoding
# => #<Encoding:UTF-8>
```

We can demonstrate that the encoding issue is due to the non-ascii inches symbol by replacing it with a double quote instead.

```
uri.path = '/ascii/example'
res = Net::HTTP.get_response(uri)
res['Content-Type']
# => "text/plain; charset=utf-8"
puts res.body
# The symbol for the inch unit of measurement is ".
res.body.encoding
# => #<Encoding:UTF-8>
res.body.ascii_only?
# => true
```

Finally, as an extra WTF, JSON.parse recognizes the non-ascii characters as valid UTF-8 in a JSON example.

```
require 'json'
uri.path = '/utf8/example_json'
res = Net::HTTP.get_response(uri)
res['Content-Type']
# => "application/json; charset=utf-8"
puts res.body
# {"feet":"′","inches":"″"}
```

```
res.body.encoding
# => #<Encoding:ASCII-8BIT>
json = JSON.parse(res.body)
# => {"feet"=>"′", "inches"=>"″"}
json.values.map { |v| [v.encoding.to_s, v] }
# => [["UTF-8", "′"], ["UTF-8", "″"]]
```

Related issues:

Is duplicate of Ruby - Feature #2567: Net::HTTP does not handle encoding corr...	**Closed**

Associated revisions

Revision ebb4378237e572ce2e888136a613c7c051439f95 - 04/11/2022 03:17 PM - jeremyevans (Jeremy Evans)

[ruby/net-http] Add HTTP#response_body_encoding for setting response body encoding

This allows for the ability to opt-in to a method to set the
encoding of response bodies. By setting the accessor to a String
or Encoding instance, it will use the specified encoding.
Setting the value of true will try to detect the encoding of the
response body, either using the Content-Type header (assuming it
specifies charset) or by scanning for a tag in the document
that specifies the encoding. The default is false in which case
no forcing of encoding will be done (same as before the patch).

Implements [Feature #2567]
Implements [Feature #15517]

https://github.com/ruby/net-http/commit/6233e6b7c1

Co-authored-by: Yui Naruse naruse@ruby-lang.org

Revision ebb4378237e572ce2e888136a613c7c051439f95 - 04/11/2022 03:17 PM - jeremyevans (Jeremy Evans)

[ruby/net-http] Add HTTP#response_body_encoding for setting response body encoding

This allows for the ability to opt-in to a method to set the
encoding of response bodies. By setting the accessor to a String
or Encoding instance, it will use the specified encoding.
Setting the value of true will try to detect the encoding of the
response body, either using the Content-Type header (assuming it
specifies charset) or by scanning for a tag in the document
that specifies the encoding. The default is false in which case
no forcing of encoding will be done (same as before the patch).

Implements [Feature #2567]
Implements [Feature #15517]

https://github.com/ruby/net-http/commit/6233e6b7c1

Co-authored-by: Yui Naruse naruse@ruby-lang.org

Revision ebb43782 - 04/11/2022 03:17 PM - jeremyevans (Jeremy Evans)

[ruby/net-http] Add HTTP#response_body_encoding for setting response body encoding

This allows for the ability to opt-in to a method to set the
encoding of response bodies. By setting the accessor to a String
or Encoding instance, it will use the specified encoding.
Setting the value of true will try to detect the encoding of the
response body, either using the Content-Type header (assuming it
specifies charset) or by scanning for a tag in the document
that specifies the encoding. The default is false in which case
no forcing of encoding will be done (same as before the patch).

Implements [Feature #2567]
Implements [Feature #15517]

https://github.com/ruby/net-http/commit/6233e6b7c1

Co-authored-by: Yui Naruse naruse@ruby-lang.org

History

#1 - 01/13/2019 10:33 AM - naruse (Yui NARUSE)

- Is duplicate of Feature #2567: Net::HTTP does not handle encoding correctly added

#2 - 01/19/2019 03:30 AM - cohen (Cohen Carlisle)

I'm not sure I think this is exactly the same as https://bugs.ruby-lang.org/issues/2567, as that one has focused on using the HTTP headers to guess the content type. Here I'm pointing out that ASCII-only strings are recognized as UTF8, but valid, multi-byte UTF8 strings are not recognized as UTF8 encoded. I suppose the trouble is that checking if the string is a valid UTF8 encoded string is not trivial, but other core/stdlib functions, like File.read seem to perform this.

#3 - 01/20/2019 01:15 AM - duerst (Martin Dürst)

I think this issue is not a duplicate of issue #2567, but it is clearly related.

Checking whether the string is valid UTF-8 is rather easy;

```
string.force_encoding('UTF-8').valid_encoding?
```

should do.

There are some security issues with browsers accepting certain mime types in certain encodings, and servers might mislabel stuff, but for text/plain; charset=utf-8, the text/plain part isn't a security issue, and text/plain doesn't have any way of indicating the encoding inside the document, so there should be no problems declaring the encoding of the resulting string as UTF-8.

#4 - 03/09/2021 10:41 PM - jeremyevans0 (Jeremy Evans)

- Tracker changed from Bug to Feature

- Status changed from Open to Assigned

- Assignee set to naruse (Yui NARUSE)

- ruby -v deleted (2.6.0)

- Backport deleted (2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN)

I've submitted a pull request (https://github.com/ruby/net-http/pull/17) that takes the patch provided by @naruse (Yui NARUSE) in #2567 and modifies it to be opt-in in a backwards compatible manner. It also fixes various issues with the patch and adds some basic tests. There are definitely cases that would not be handled correctly in terms of detecting content through meta tags, but since it is opt-in it should not break existing code.

The current behavior is not considered a bug, so I'm switching this to a feature request, the same as #2567.

#5 - 04/12/2022 12:56 PM - jeremyevans (Jeremy Evans)

- Status changed from Assigned to Closed

Applied in changeset git|ebb4378237e572ce2e888136a613c7c051439f95.

[ruby/net-http] Add HTTP#response_body_encoding for setting response body encoding

This allows for the ability to opt-in to a method to set the
encoding of response bodies. By setting the accessor to a String
or Encoding instance, it will use the specified encoding.
Setting the value of true will try to detect the encoding of the
response body, either using the Content-Type header (assuming it
specifies charset) or by scanning for a tag in the document
that specifies the encoding. The default is false in which case
no forcing of encoding will be done (same as before the patch).

Implements [Feature #2567]
Implements [Feature #15517]

https://github.com/ruby/net-http/commit/6233e6b7c1

Co-authored-by: Yui Naruse naruse@ruby-lang.org