

Ruby master - Feature #4142

multipart/form-data for net/http

12/10/2010 09:17 AM - naruse (Yui NARUSE)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
<pre>=begin multipart/form-data net/http API Net::HTTPRequest#set_form akr multipart/form-data API chunked encoding diff --git a/lib/net/http.rb b/lib/net/http.rb index 4d475b1..2751f77 100644 --- a/lib/net/http.rb +++ b/lib/net/http.rb @@ -22,6 +22,7 @@ require 'net/protocol' autoload :OpenSSL, 'openssl' require 'uri' +autoload :SecureRandom, 'securerandom' module Net #:nodoc: @@ -1772,7 +1773,8 @@ module Net #:nodoc: alias content_type= set_content_type # Set header fields and a body from HTML form data. • # +params+ should be a Hash containing HTML form data. • # +params+ should be an Array of Arrays or • # a Hash containing HTML form data. # Optional argument +sep+ means data record separator. # # Values are URL encoded as necessary and the content-type is set to @@ -1792,6 +1794,48 @@ module Net #:nodoc: alias form_data= set_form_data • # Set a HTML form data set. • # +params+ is the form data set; it is an Array of Arrays or a Hash • # +enctype+ is the type to encode the form data set. • # It is application/x-www-form-urlencoded or multipart/form-data. • # +formpot+ is an optional hash to specify the detail. • # • # boundary:: the boundary of the multipart message • # charset:: the charset of the message. All names and the values of • # non-file fields are encoded as the charset.</pre>	

- #
- # Each item of params is an array and contains following items:
- # +name+:: the name of the field
- # +value+:: the value of the field, it should be a String or a File
- # +opt+:: an optional hash to specify additional information
- #
- # Each item is a file field or a normal field.
- # If +value+ is a File object or the +opt+ have a filename key,
- # the item is treated as a file field.
- #
- # If Transfer-Encoding is set as chunked, this send the request in
- # chunked encoding. Because chunked encoding is HTTP/1.1 feature,
- # you must confirm the server to support HTTP/1.1 before sending it.
- #
- # Example:
- # `http.set_form("q", "ruby").{"lang" => "en"}`
- #
- # See also RFC 2388, RFC 2616, HTML 4.01, and HTML5
- #
- `def set_form(params, enctype='application/x-www-form-urlencoded', formopt={})`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `[REDACTED]`
- `end`
- `+`
- `# Set the Authorization: header for "Basic" authorization.`
- `def basic_auth(account, password)`
- `@header['authorization'] = [basic_encode(account, password)]`

```
@@ -1849,6 +1893,7 @@ module Net #:nodoc:
self['User-Agent'] ||= 'Ruby'
@body = nil
@body_stream = nil
```

- [REDACTED]

```
end
```

```
attr_reader :method
@@ -1876,6 +1921,7 @@ module Net #:nodoc:
def body=(str)
@body = str
@body_stream = nil
```

- [REDACTED]

```
str
```

```
end
```

```
@@ -1884,6 +1930,7 @@ module Net #:nodoc:
def body_stream=(input)
@body = nil
@body_stream = input
```

- @body_data = nil input end

```
@@ -1901,6 +1948,8 @@ module Net #:nodoc:
send_request_with_body sock, ver, path, @body
elsif @body_stream
send_request_with_body_stream sock, ver, path, @body_stream
```

- elsif @body_data

- [REDACTED]

```
else
write_header sock, ver, path
end
```

```
@@ -1935,6 +1984,92 @@ module Net #:nodoc:
end
end
```

- def send_request_with_body_data(sock, ver, path, params)

- [REDACTED]

- [REDACTED]

- [REDACTED]

- [REDACTED]

```
+
```

- [REDACTED]

- [REDACTED]

- [REDACTED]

- [REDACTED]

- [REDACTED]

- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]
- end
 - +
 - def supply_default_content_type
 - return if content_type()
 - warn 'net/http: warning: Content-Type did not set; using application/x-www-form-urlencoded' if \$VERBOSE
 - diff --git a/lib/net/protocol.rb b/lib/net/protocol.rb
 - index 2a6cfb4..a3ffa71 100644
 - a/lib/net/protocol.rb
 - +++ b/lib/net/protocol.rb
 - @@ -168,6 +168,8 @@ module Net # :nodoc:
 - }
- alias << write
 - +
 - def writeline(str)
 - writing {
 - write0 str + "\r\n"
 - diff --git a/test/net/http/test_http.rb b/test/net/http/test_http.rb
 - index 76280ad..12c03a4 100644
 - a/test/net/http/test_http.rb
 - +++ b/test/net/http/test_http.rb
 - @@ -303,6 +303,102 @@ module TestNetHTTP_version_1_2_methods
 - assert_equal data.size, res.body.size
 - assert_equal data, res.body
 - end
- def test_set_form
- require 'tempfile'
- file = Tempfile.new('ruby-test')
- file << "\u{30c7}\u{30fc}\u{30bf}"
- data = [
 - [REDACTED]
 - [REDACTED]
 - [REDACTED]
 - [REDACTED]
-]
- expected = <<"EOM".gsub(/\n/, "\r\n")
 - +--
 - +Content-Disposition: form-data; name="name"
 - +
 - +Gonbei Nanashi
 - +--
 - +Content-Disposition: form-data; name="name"
 - +
 - +\xE5\x90\x8D\xE7\x84\xA1\xE3\x81\x97\xE3\x81\xAE\xE6\xA8\xA9\xE5\x85\xB5\xE8\xA1\x9B
 - +--
 - +Content-Disposition: form-data; name="s\i\o"

```
+
+\xE3\x81\x82\xE3\x81\x84\xE4\xBA\x9C\xE9\x89\x9B
+--
+Content-Disposition: form-data; name="file"; filename="ruby-test"
+Content-Type: application/octet-stream
+
+\xE3\x83\x87\xE3\x83\xBC\xE3\x82\xBF
+----
+EOM
```

- start {|http|
- [REDACTED]
- [REDACTED]
- [REDACTED]
- }
- end
- +
- def _test_set_form_urlencoded(http, data)
- req = Net::HTTP::Post.new('/')
- req.set_form(data)
- res = http.request req
- assert_equal
"name=Gonbei+Nanashi&name=%E5%90%8D%E7%84%A1%E3%81%97%E3%81%AE%E6%A8%A9%E5%85%B5%E8%A1%9B", res.body
- end
- +
- def _test_set_form_multipart(http, chunked_p, data, expected)
- data.each{|k,v|v.rewind rescue nil}
- req = Net::HTTP::Post.new('/')
- req.set_form(data, 'multipart/form-data')
- req['Transfer-Encoding'] = 'chunked' if chunked_p
- res = http.request req
- body = res.body
- assert_match(/^A--(?:\S+)/, body)
- /^A--(?:\S+)/ =~ body
- expected = expected.gsub(/,/, boundary)
- assert_equal(expected, body)
- end
- +
- def test_set_form_with_file
- require 'tempfile'


```
class TestNetHTTP_version_1_1 < Test::Unit::TestCase
=end
```

History

#1 - 12/13/2010 10:07 AM - naruse (Yui NARUSE)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin

This issue was solved with changeset r30188.

Yui, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

=end

#2 - 06/20/2011 02:27 PM - nahi (Hiroshi Nakamura)

- Status changed from Closed to Open

Naruse-san, would you please add an explanation of this feature to NEWS file?

#3 - 06/20/2011 02:28 PM - nahi (Hiroshi Nakamura)

ruby-dev orz

NEWS

#4 - 06/27/2011 12:55 AM - naruse (Yui NARUSE)

- Status changed from Open to Closed

r32241