

Ruby trunk - Feature #15624

Allow net/http Response to close before reading entire body

02/27/2019 03:25 AM - sam.saffron (Sam Saffron)

Status:	Rejected
Priority:	Normal
Assignee:	
Target version:	
Description	
Currently net/http has:	
<pre>def reading_body(sock, reqmethodallowbody) #:nodoc: internal use only @socket = sock @body_exist = reqmethodallowbody && self.class.body_permitted? begin yield self.body # ensure to read body ensure @socket = nil end end</pre>	
The call to self.body ensures that unconditionally if you GET you must read the entire body.	
For certain use cases a "partial" GET is useful, you may only be interested in reading the first 10000 bytes of a page and can act on that.	
Trouble is this API dictates that unconditionally the entire GET request must be consume.	
Proposal:	
Add #close on Response to allow for early closing of stream.	
So, instead of:	
<pre>def get_status_code(headers) status_code = nil Net::HTTP.start(@uri.host, @uri.port, use_ssl: @uri.is_a?(URI::HTTPS)) do http http.open_timeout = timeout http.read_timeout = timeout http.request_get(@uri.request_uri, headers) do resp status_code = resp.code.to_i resp.instance_variable_set(:@body_exist, false) resp.instance_variable_set(:@body, "") end end status_code end</pre>	
We could have:	
<pre>def get_status_code(headers) status_code = nil Net::HTTP.start(@uri.host, @uri.port, use_ssl: @uri.is_a?(URI::HTTPS)) do http http.open_timeout = timeout http.read_timeout = timeout http.request_get(@uri.request_uri, headers) do resp status_code = resp.code.to_i resp.close end end status_code end</pre>	

```
end
```

```
status_code  
end
```

Happy to submit the patch

History

#1 - 02/27/2019 01:55 PM - naruse (Yui NARUSE)

- *Status changed from Open to Rejected*

As far as I understand HTTP doesn't provide such close without reading.

You know HTTP protocol itself doesn't provide such early close mechanism itself.

And though TCP, HTTP's underlying protocol, provides its error handling packet flag, RST, it may cause problems because such error handling need to keep resources longer than normal connection close.

Therefore if you really want to fetch initial bytes and you can send Range Request if the server support it or just hack net/http by your own risk.