Ruby - Feature #16231

Add #location to Net::HTTPResponse

10/03/2019 03:53 AM - RichSeviora (Richard Seviora)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

Abstract

Add a location convenience method to the Net::HTTPRedirection class.

Background

When developers receive 3xx responses, we tend to do one of two things: follow the redirect or pass the redirect location onto the consumer. In both cases, we need to get the Location header value. This is a common enough use case that I had expected there'd be a convenience method, and there isn't one. I ended up googling how to do this and discovered I had to call the following:

```
response['Location']
```

Proposal

My proposal is to return the Location header value as-is (String if present, nil if header is not present) in a #location method added to Net::HTTPResponse. This will permit consumers to access the Location header in all of the response classes.

Per RFC 7231 section 7.1.2 the Location header is not limited to 3xx responses; it can also be returned in 201 Created responses. Augmenting the Net:HTTPResponse class makes more sense than augmenting (only) the Net::HTTPRedirection class.

Implementation

```
class Net::HTTPResponse
  # ...
  # Returns the location value if present, nil otherwise.
  def location
    self['Location']
  end
  # ...
end
```

Evaluation

This has nil performance impact when unused, it adds negligible costs over directly executing the same (guesstimate, I have not validated this)

Discussion

Instead of returning a String, we could instead instantiate and return a URI object when the method is called (with the option to memoize). The implementation would need to handle both absolute and relative URIs, and decide whether to return the absolute resolved URI, or just the relative URI.

I'm leaning towards returning the string as-is, and letting the consumer decide whether they need to instantiate a URI or just pass the string along.

History

#1 - 10/03/2019 11:51 AM - shevegen (Robert A. Heiler)

Seems ok to me.

#2 - 10/06/2019 11:14 PM - prajjwal (Prajjwal Singh)

Seems like a good first issue, I'd like to take this up if there's any interest in having it.