#tap with missing block

09/11/2013 07:05 PM - prijutme4ty (Ilya Vorontsov)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

In case when no block provided to tap, it fails. So if you want method which can be called with or without block - you can't do it like this:

```ruby
def self.create(args={}, &block)
  new.set_attributes(args).tap(&block)
end
```

but should use `block_defined?` to prevent `#tap`'s failure.

I suggest to make tap with no block given just to return self and raise no errors.

**History**

#1 - 09/11/2013 08:23 PM - charliesome (Charlie Somerville)

I would like to see `tap` return an Enumerator if a block is not passed.

#2 - 09/12/2013 01:54 AM - prijutme4ty (Ilya Vorontsov)

I would like to see `tap` return an Enumerator if a block is not passed.

Can you explain your suggestion a bit more? I can't imagine realization and use-case yet.

#3 - 09/12/2013 01:59 AM - fuadksd (Fuad Saud)

#tap without a block returning self makes sense

--

Fuad Saud
Sent with Sparrow ([http://www.sparrowmailapp.com/?sig](http://www.sparrowmailapp.com/?sig))

On Wednesday, September 11, 2013 at 1:54 PM, prijutme4ty (Ilya Vorontsov) wrote:

Issue [#8896](https://bugs.ruby-lang.org/issues/8896#change-41756) has been updated by prijutme4ty (Ilya Vorontsov).

I would like to see `tap` return an Enumerator if a block is not passed.

**Can you explain your suggestion a bit more? I can't imagine realization and use-case yet.**

Feature #8896: #tap with missing block

[https://bugs.ruby-lang.org/issues/8896#change-41756](https://bugs.ruby-lang.org/issues/8896#change-41756)

Author: prijutme4ty (Ilya Vorontsov)

Status: Open

Priority: Normal

Assignee:

Category:

Target version:

In case when no block provided to tap, it fails. So if you want method which can be called with or without block - you can't do it like this:

```ruby
def self.create(args={}, &block)
  new.set_attributes(args).tap(&block)
end
```
but should use `block_defined?` to prevent `#tap`'s failure.

I suggest to make `tap` with no block given just to return `self` and raise no errors.

---

http://bugs.ruby-lang.org/

#4 - 12/15/2018 09:48 PM - shuber (Sean Huber)
Kernel#tap related: Allow Kernel#tap to be invoked with arguments like Kernel#send - https://bugs.ruby-lang.org/issues/15419

If the LocalJumpError breaking change from this ticket is not an issue then it'd be pretty easy to add logic to that PR^ supporting it.

#5 - 12/23/2021 11:43 PM - hsbt (Hiroshi SHIBATA)
- Project changed from 14 to Ruby master