

# Rubyにpatchを 送ってみよう！

2018-07-29 at 株式会社VOYAGE GROUP  
TokyuRuby会議12

うなすけ

# 自己紹介

---

- 名前 : うなすけ
- 所属 : 株式会社バンク (エンジニア)
  - インフラ寄りサーバーサイドエンジニア
- GitHub [@unasuke](#)
- Twitter [@yu\\_suke1994](#)
- Mastodon [@unasuke@mstdn.maud.io](#)



# 最近

---



寝転がったままパソコンを使う | うなすけとあれこれ

# tqrk12



うなすけ

@yu\_suke1994



TokyuRuby会議12 LT、送信後に気づいたんだけど発表順序が最後のほう希望になってて死(どこでもいいのに……)

17:17 - 2018年6月13日

1件のいいね



# tqrk12



うなすけ

@yu\_suke1994



トリじゃん!!!! #tqrk12

TokyuRuby会議12 - Regional RubyKaigi  
[regional.rubykaigi.org/tokyu12/](https://regional.rubykaigi.org/tokyu12/)

0:47 - 2018年7月23日

12件のいいね



# はい

---

- はい

# Rubyにpatchを送る

---

- どうやって？
  - SVNわかんない……
  - C言語わかんない……
- そもそも送るネタがない
  - 私に直せるバグなんてある？
  - 新機能なんて思いつかないし……

# Rubyにpatchを送る

---

- どうやって？
  - SVNわかんない…… ←それな
  - C言語わかんない…… ←それな
- そもそも送るネタがない
  - 私に直せるバグなんてある？ ←それな
  - 新機能なんて思いつかないし…… ←それな



# Rubyにpatchを送った

---

ぼくのばあい

# bank/karakuri\_ai-ruby

The screenshot shows the GitHub interface for the repository `bank / karakuri_ai-ruby`. At the top, there is a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below the repository name, there is an 'Unwatch' button and a row of tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. The URL `https://rubygems.org/gems/karakuri_ai` is displayed, along with an 'Add topics' link. A summary bar shows 23 commits, 1 branch, 1 release, and 1 contributor. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', and 'Upload file'. A commit by `unasuke` with the message 'Fix typo' is highlighted. At the bottom, the file `bin` is shown with its associated command: `bundle gem --mit --test=minitest --no-coc karakuri_ai`.

Search or jump to... / Pull requests Issues Marketplace Explore

bank / karakuri\_ai-ruby Unwatch

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

[https://rubygems.org/gems/karakuri\\_ai](https://rubygems.org/gems/karakuri_ai)  
Add topics

23 commits 1 branch 1 release 1 contributor

Branch: master New pull request Create new file Upload file

unasuke Fix typo

bin bundle gem --mit --test=minitest --no-coc karakuri\_ai

[https://github.com/bank/karakuri\\_ai-ruby](https://github.com/bank/karakuri_ai-ruby)

# bank/karakuri\_ai-ruby's gemspec

---

```
spec.add_development_dependency "bundler", "~> 1.16"  
spec.add_development_dependency "rake", "~> 10.0"  
spec.add_development_dependency "minitest", "~> 5.0"  
spec.add_development_dependency "webmock"
```

[karakuri\\_ai-ruby/karakuri\\_ai.gemspec at master · bank/karakuri\\_ai-ruby · GitHub](#)

- runtime\_dependencyがない！
  - 標準ライブラリのみを使用している

# net/http のみで実装するAPI Client Gem

---

## ■ 動機

- 無駄に依存を増やしたくない
  - Faraday
  - httparty
  - etc...
  - <https://postd.cc/kill-your-dependencies/>
- 標準ライブラリのお勉強

# net/http のみで実装するAPI Client Gem

---

- 機能

- 認証

- JSONをPOROにmapping

- エラーが返ってきたら例外 ← ここ

# Net::HTTPResponse#value

---

[class Net::HTTPResponse \(Ruby 2.5.0\)](#)

レスポンスが 2xx(成功)でなかった場合に、対応する例外を発生させます。

オッ、便利か～～？？？

# Net::HTTPResponse#value

---

[class Net::HTTPResponse \(Ruby 2.5.0\)](#)

*[EXCEPTION] HTTPServerException:*

レスポンスが 4xx である場合に発生します。



# HTTP status code

---

*The 4xx class of status code is intended for cases in which the client seems to have erred. The 4xx class of status code is intended for cases in which the client seems to have erred.*

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

400系はクライアントが悪い場合……だよね？



# Net::HTTPResponse#value

---

[class Net::HTTPResponse \(Ruby 2.5.0\)](#)

*[EXCEPTION] HTTPServerException:*

レスポンスが 4xx である場合に発生します。

- 間違ってるのはどっち？
  - Ruby Net::HTTPResponse#value
  - 俺

困ったときの

# Asakusa.rb



<http://www.flickr.com/photos/takkanm/3978417669>

# 困ったときの Asakusa.rb

---

usa-san says...

これは仕様のバグですね

- 間違ってるのはどっち？
  - `Ruby Net::HTTPResponse#value` ← こっち
  - 俺

# なおす

---

- なおす
  - SVNわかんない……
  - C言語わかんない……

# なおす

---

- なおす
  - SVNわかんない……
    - 使うのはgitだけ
  - C言語わかんない……
    - (今回)書くのはRuby

やっていきましょう

# なおし手順

---

1. cloneする
2. grepする
3. みつける
4. 書き換える
5. testを回す
6. [bugs.ruby-lang.org](https://bugs.ruby-lang.org)に投稿
7. 待つ

# なおし手順 - cloneする

---

- git cloneでダイジョウブ
  - <https://github.com/ruby/ruby.git> ← オススメ
  - <https://git.ruby-lang.org/ruby.git>

# なおし手順 - grepする

---

```
$ git grep HTTPServerException
```

lib/net/http/exceptions.rb が怪しい！

```
# lib/net/http/exceptions.rb
module Net::HTTPExceptions
  def initialize(msg, res)    #:nodoc:
    super msg
    @response = res
  end
  attr_reader :response
  alias data response      #:nodoc: obsolete
end
### snip ###
```



# なおし手順 - みつける

---

```
class Net::HTTPServerException < Net::ProtoServerError
  # We cannot use the name "HTTPServerError", it is the name of the response.
  include Net::HTTPExceptions
end
```

[https://github.com/ruby/ruby/blob/v2\\_5\\_1/lib/net/http/exceptions.rb#l19-l22](https://github.com/ruby/ruby/blob/v2_5_1/lib/net/http/exceptions.rb#l19-l22)

ここだ~~~~~

# なおし手順 - 書き換える

---

- 単純に名前を換えるのヤバそう
  - この例外の名前を使ってるコードがぶっ壊れ
  - `alias` という便利機能があるっぽい
  - <https://docs.ruby-lang.org/ja/latest/doc/spec=2fdef.html#alias>
  - Rubyってすごいな～～

# なおし手順 - 書き換える

---

しかしこういうことはできない！

```
alias Net::NiceExceptionName Net::HTTPServerException
```

これならできる

```
module Net
  alias NiceExceptionName HTTPServerException
end
```

# なおし手順 - 書き換える

---

結局こうした

```
# for compatibility  
Net::HttpClientException = Net::HTTPServerException
```

usaさんが代入すればいい的なことをおっしゃってたので……

# なおし手順 - testを回す

---

- 変更してtestが通るか確認
  - make check
  - なんならテストも書き換える

```
# test/net/http/test_httpresponse.rb
res = Net::HTTPResponse.read_new(io)
assert_equal(nil, res.message)
-  assert_raise Net::HTTPServerException do
+  assert_raise Net::HTTPClientException do
    res.error!
  end
end
```

# なおし手順 - bugs.ruby-lang.orgに投稿

---

1. ユーザー登録する
2. チケット → 「新しいチケット」をクリック
3. 思いの丈をぶつける

[Feature #14688: Net::HTTPResponse#value raises “Net::HTTPServerException” in 4xx response - Ruby trunk - Ruby Issue Tracking System](#)

# なおし手順 - 待つ

---

- open 04/15 → close 06/06
  - 約2ヶ月
  - きっかけはnaruseさんに直に相談
    - 「会える」は強い
  - みんな暇じゃない

# my patch approved!



うなすけ

@yu\_suke1994



Rubyに送ったpatchがtrunkにmergeされていて僕は泣きそうです

[bugs.ruby-lang.org/issues/14688](https://bugs.ruby-lang.org/issues/14688)

18:12 - 2018年6月6日

7件のリツイート 77件のいいね



6



7



77





# Rubyにpatchを送る

---

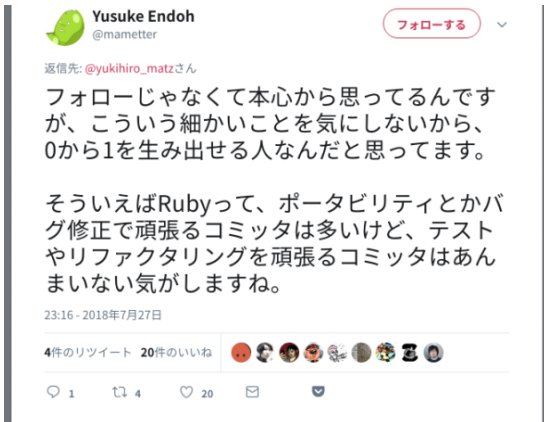
- どうやって？
  - SVNわかんない……
    - 使うのはgitだけ
  - C言語わかんない……
    - 領域による

# Rubyにpatchを送る

---

- そもそも送るネタがない
  - 私に直せるバグなんてある？
  - 新機能なんて思いつかないし……

# Rubyにpatchを送る





<https://twitter.com/mametter/status/1022848181590052864>

# Rubyにpatchを送る


---

テストやリファクタリングを頑張るコミッタは  
あんまない


# Rubyにpatchを送る







 [ruby](#) / [ruby](#)  Wa

[Code](#) [Pull requests 157](#) [Insights](#)

[Filters](#)  is:pr author:igaiga is:closed

[Labels](#) [Milestones](#)

 Clear current search query, filters, and sorts

 0 Open	 11 Closed	Author	Labels	Mil
 <b>Add lacking test for File#lstat.</b> ✓	#1231 by igaiga was closed on 31 Jan 2016			
 <b>Add test cases for ENV#reject!, select!</b> ✓	#1201 by igaiga was closed on 17 Jan 2016			
 <b>Add test for Hash.try_convert</b> ✓	#1190 by igaiga was closed on 9 Jan 2016			
 <b>Add test for Name#initialize</b> ✓				

# Rubyにpatchを送る

---

- そもそも送るネタがない
  - 私に直せるバグなんてある？
    - 不足しているテストを書くによさそう
- 新機能なんて思いつかないし……
  - (今)思いつかなくてもいいのでは
  - あせらない、あせらない

# Ruby communityにpatchを送る

---

- ruby/rubyだけがRubyじゃない

「*Rubyist*」という言葉は造語で、*Ruby*に対して単なるお客さん以上の気持ちを持っている人が*Rubyist*です。

- rurema/doctree
- rubima/magazine.rubyist.net
- many rubygems...

# まとめ

---

- Ruby 2.6には僕のpatchが含まれてます
- Rubyにpatchを送るには
  - gitの知識で十分
  - GitHubにpull requestを送るのもOK
- We are Rubyist!