

Let's Rspec

2019-01-16
加藤洋一

目的

- 実装変更したら、rspec で確認してから PR をつくるのが習慣になるようにしたい。
 - 他人が書いた rspec ファイルコードを読めるようになる。
 - Rspec を効率的に利用できるようになる。

10 分間の話を聞いただけでできるはずはない！

ここでは その学習のきっかけが作れば良い。

- 1 人で勉強するのが苦なら勉強会を開催する？)

教材

- <https://qiita.com/jnchito/items/60ea1389d00e72b729>
「プロを目指す人のためのRuby入門」のテストコードを、筆者自らRSpecに書き換えてみます。

github: <https://github.com/Junichilto/ruby-book-codes/tree/rspec/ruby-book>

youtube: <https://www.youtube.com/watch?v=BmPD2KIZ5eY&feature=youtu.be> (30分)

自習実施例

「プロを目指す人のためのRuby入門」のrspecを書き直している。(in progress)

<https://github.com/katoy/ruby-book-codes/tree/study>

これがわかれば読書できる!

- describe / context / example / it
xを頭につけると pnding にできる
- expect
- subject / is_expected
- change.by / change.from.to / .and で結合可能
- let
- before / after / around

- `receive(...).with(...).and_return`
 - `expect(Rails.logger).to receive(:info).with(xxx)` で logger 出力をチェックできる。
 - csv 入力内容を ファイルを作らずに読込んだ動作ができる。
- `output`
stdout, stderr への出力をチェックできる。
- `double / stub`
DB アクセス, 外部 API アクセスをダミーにできる

- `shared_context / include_context`
- `shared_example / it_behaves_like`
- `aggregate_failures`
- `raise_error` message

最近のテスト作業上のネタ

- rake のタスク全体にある処理を数せていたが、引数指定に対応していなくて引数指定した際に動作していなかった。
- eagerload 設定が develop / production で違って、rake task 動作に差がでていた。

- hash からいくつかの key を削除する際に "key" と :key の指定を混同していて、hash から情報削除できてなかった。
- rapec でのテストを追加したら Fail した。調べると x年前の更新 PR 説明では rspec が pass することになっていたが、その変更部は通過していなかった。あるいは、rspec が全くそのクラスについては作成されていなかった。

テストの種類

- モデルスペック
- コントローラスペック
- システムスペック
- 結合テスト
- リクエストスペック
- ルーティングテスト
- サーバースペック (サーバ状態のテストツール)

関連ツール

- coverage

```
$ rspec  
$ open coverage/index.ht
```

All Files (97.5%)

Generated less than a minute ago

All Files (97.5% covered at 8.84 hits/line)

11 files in total. 80 relevant lines. 78 lines covered and 2 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	L
Q lib/word_synth.rb	100.0 %	17	8	8	0
Q lib/ticket.rb	100.0 %	13	6	6	0
Q lib/team.rb	100.0 %	9	4	4	0
Q lib/rgb.rb	100.0 %	15	7	7	0
Q lib/gate.rb	100.0 %	42	17	17	0
Q lib/fizz_buzz.rb	100.0 %	9	6	6	0
Q lib/deep_freezable.rb	100.0 %	16	8	8	0
Q lib/convert_length.rb	100.0 %	10	5	5	0
Q lib/convert_hash_syntax.rb	100.0 %	5	2	2	0
Q lib/bank.rb	100.0 %	11	4	4	0
Q lib/effects.rb	84.62 %	27	13	11	2

Showing 1 to 11 of 11 entries

- TimeHelper / TimeCop

https://techracho.bpsinc.jp/penguin10/2018_12_25/6

Timecopを使わなくても時間を止められた話

... テストで時間を止めたい時、
*Timecop*を使わないと止められないと
思っていました。が、*Rails4.1*以降は標
準の`ActiveSupport::Testing::TimeHelpers`
でも止められると最近知りました。 ...

- API Blueprint

https://blog.mmmcorp.co.jp/blog/2018/01/18/api_blueprint/

API Blueprintで書いたAPI仕様書をDreddを使ってCI
トしてみた

- metric_fu

静的コードチェック

- gathling

<https://qiita.com/yasuaki0206/items/c84361faef08d67>

gatlingで負荷テストを行った

- OWASP Zed <https://www.pupha.net/archives/1370/>

OWASP Zed Attack Proxy (ZAP)で脆弱性検査する方法

参考書籍/URL

- <https://railsguides.jp/testing.html#%E3%82%B7%E3%9D%A0>
Rails テスティングガイド
- <https://www.amazon.co.jp/gp/product/B076VTMTV8/> (5th Edition)

- <https://leanpub.com/everydayrailsrspec-jp>
Everyday Rails - RSpecによるRailsテスト入門
- ソースコード
<https://github.com/everydayrails/everydayrails-rspec-2017>

- <https://www.amazon.co.jp/dp/4274217884/> テスト
駆動開発 新訳版 (2017/10/14)

... テストは品質をあげない- 質を
あげるのはプログラミング... テス
トを書いても設計を改善しないの
であれば、それはただの回帰テス
トであり、現状の追従でしかあり
ません。 ...

