Rails のテストの仕組み

y-yagi　著

2019-04-01 版　　発行

はじめに

本書の内容

本書は、Ruby on Rails(以降 Rails) が提供しているテストの仕組みについて説明した本です。

普段 Rails を使用してアプリケーションを作られている方は、一緒にテストも書かれているかと思います。Rails は Web アプリケーションを作る為のフレームワークですが、そのアプリケーションをテストする為の仕組みも合わせて提供しています。本書では、Rails がどのような仕組みを提供しているかについて説明します。

そのため、Rails を使用したアプリケーションでのテストについての話ではなく、Rails が提供しているテストについての話です。仕組みの使い方について詳細には触れていません。ご了承ください。

なお、特に注記が無い場合は、Rails は執筆時点で最新のバージョン (6.0.0.beta3, commit: 6a5c8b9) を対象にしています。

対象読者

本書は、既に Ruby、Rails を使った事がある人を対象としています。そのため、Ruby や Rails 自体の説明や、Ruby に関連するツールについての説明は端折っています。予めご了承ください。

免責事項

本書に記載された内容は、情報の提供のみを目的としています。したがって、本書を用いた開発、製作、運用は、必ずご自身の責任と判断によって行ってくだ

さい。これらの情報による開発、製作、運用の結果について、著者はいかなる責任も負いません。また、本書に記載されいてる情報は、執筆時点でのものです。時間経過によって情報が古くなっている可能性がありますのであらかじめご了承ください。

第 1 章

Rails のテスト

本章では、Rails が提供しているテストの概要について説明します。

1.1 テストフレームワーク

Ruby でテストフレームワークといえば、test-unit、minitest、RSpec の名前が挙がることが多いと思います。Rails は、かつて test-unit を使っていて、現在は minitest を使用しています。Rails 自体のテストも、Rails を使用するユーザに向けたテストの仕組みも、どちらも minitest を使用しています。日本で Rails アプリケーションを開発されている場合、RSpec をお使いの方が多いのではないかと思います。しかし実は RSpec は Rails がデフォルトでサポートしているテストフレームワークではなく、Rails 自体に RSepc の為の機能は全くありません。

1.2 minitest と Rails

先に述べた通り、Rails では minitest を使用しています。が、minitest をそのまま使用している訳ではありません。色々と機能拡張をおこなっています。

例えば、minitest では否定の assert を行うのに、`refute` というメソッド、及び、`refute` で始まる各種 assert 用のメソッドを使用します。例えば値が一致しない事を確認したい場合、`refute_equal` というメソッドを使用します。

リスト 1.2: refute_equal

```
refute_equal 5, User.count
```

しかし Rails では refute を使うことを推奨していません。代わりに、assert_not というメソッド、及び、assert_not で始まる各種 assert 用が提供されており、そちらを使用することを推奨しています。先の例だと、代わりに assert_not_equal を使用する必要があります。

リスト 1.2: refute_equal

```
assert_not_equal 5, User.count
```

元々 test-unit には assert_not があり、test-unit から minitest に移行する際に互換性の為にこれらのメソッドに追加されました。その後、refute に移行する？というような提案もあったようなのですが、それは進まず、assert_not を使う形のままで落ち着いています[*1]。

なお、refute ではなく assert_not を使用する事をチェックする為の RuboCop の設定[*2]もあります。Rails のリポジトリではこの cop が有効化されており、refute は一切使われないようになっています。

他にも、Rails でテストを書く際、test というメソッドを使用してテストを定義します。

リスト 1.3: test

```
test "should get index" do
  get users_url
  assert_response :success
end
```

[*1] 当時のちゃんとした議論が見つからなかったので推測混じりなのですが、どうも refute という名前をあまり好ましく思わない人がいたため、移行は行われなかったようです。恐らくレーサー。

[*2] https://www.rubydoc.info/gems/rubocop/RuboCop/Cop/Rails/RefuteMethods

これも Rails が提供しているメソッドで、minitest だけを使用している場合、このメソッドは使用する事は出来ません。

そのため、「Rails は mintiest を使用している」を正確ではなく、「Rails は minitest を拡張した独自の仕組みを使用している」が正確な表現になります。

1.3　テスト用のクラス

Rails というフレームワークは機能ごとにライブラリがわかれています。例えば O/R マッパーの Active Record、template の表示を行う Action View、メール送信の為の Action Mailer、という具合です。

Rails では、各ライブラリ毎に、そのライブラリの機能を提供する為のクラスを提供しています。クラス図は次のようになっています。

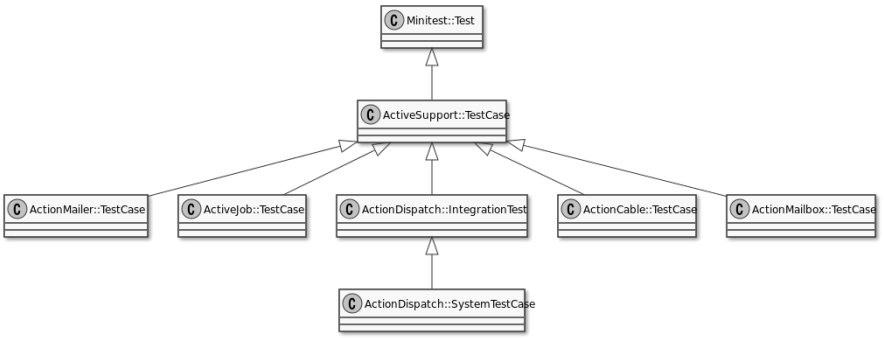

図 1.1: クラス図

紙面の都合上大分割愛していますが、minitest のテスト用のクラスである Minitest::Test を継承した ActiveSupport::TestCase というクラスがあり、各ライブラリのテストクラスはその ActiveSupport::TestCase を継承している、という点だけ覚えておいて下さい。各クラスの詳細については次章で説明します。

1.4 まとめ

本章では、Rails が提供しているテストの概要について説明しました。Rails は minitest を使用している、しかしそのまま使用しているのではなく、minitest を拡張して使用している、という点だけご理解頂けると幸いです。

第 2 章

テスト用のクラス

本章では、Rails が提供しているテスト用のクラスについて説明します。

2.1 クラスの一覧

表 2.1: Rails が提供しているテストクラス

名前 意味
PATH コマンドの存在するディレクトリ
TERM 使っている端末の種類。linux・kterm・vt100 など
LANG ユーザのデフォルトロケール。日本語なら ja_JP.eucJP や ja_JP.utf8
LOGNAME ユーザのログイン名
TEMP 一時ファイルを置くディレクトリ。/tmp など
PAGER man などで起動するテキスト閲覧プログラム。less など
EDITOR デフォルトエディタ。vi や emacs など
MANPATH man のソースを置いているディレクトリ
DISPLAY X Window System のデフォルトディスプレイ

Railsのテストの仕組み

2019年4月1日　初版第1刷　発行

著　者　y-yagi