

Redmineプラグインの テスト自動化を頑張っている話

2022/11/05(土) redmine.tokyo 第23回勉強会

Ko Nagase @sanak



自己紹介

- 名前: Ko Nagase (ながせ)
- GitHub: [@sanak](#)
- Twitter: [@geosanak](#) ※ブックマーク代わりに利用
- Redmineとの関わり
 - Redmineの利用歴: 10年
 - Redmineプラグインの開発・メンテナンス歴: 4年
 - Redmineパッチ会メンバー (第4回から参加)
 - Redmine Japan実行委員会メンバー (Vol.2から参加)
 - redmine.tokyo 第20回LT発表
 - [Redmineで地理空間情報を扱う、Redmine GTT \(Geo-Task-Tracker\) plugin の紹介](#)



きっかけ (1 / 3)

- 約2年前(2020/10/24)の第4回 Redmineパッチ会に初めて参加
 - 初めてのDiscord
 - 初めてのモブプロ
 - 初めてのテスト駆動開発
 - **GitHub上でのRedmineテスト自動実行を初めて見た**



The screenshot shows a GitHub Actions workflow run for 'Add test #84'. The workflow is titled 'Test' and is public. The summary section shows a list of jobs: 'test-with-mysql (2.4, 5.7)', 'test-with-mysql (2.5, 5.7)', 'test-with-mysql (2.6, 5.7)', 'test-with-mysql (2.7, 5.7)', 'test-with-postgres (2.4, 9.5)', 'test-with-postgres (2.5, 9.5)', 'test-with-postgres (2.6, 9.5)', 'test-with-postgres (2.7, 9.5)', and 'test-with-sqlite (2.4)'. The total duration is 9m 34s. A speech bubble points to the workflow run, stating: 'Rubyのバージョン、DB種別毎にテスト結果が分かる(リンク)'. The bottom right corner of the screenshot shows a message: 'This workflow graph will be generated'.



きっかけ (2 / 3)

- 業務では20以上のプラグインを利用するプロジェクトに参加していて、Redmineのバージョンアップに伴うプラグイン側の対応はいつも大変 (いずれも数ヶ月以上かかる作業)
 - Redmine 3.4 => 4.0 (2019年対応)
 - Redmine 4.0 => 4.2 (2021年対応、一部プラグインの整理・統合含む)
 - Redmine 4.2 => 5.0 (2022年対応、デプロイ準備中)
- 他に、開発したプラグインで、MySQLだと動かないという報告があった。(普段はPostgreSQLを利用)
 - [Deploying on Redmine 5.x did not work. May be a DB issue · Issue #19 · gtt-project/redmine custom fields groups](#)



きっかけ (3 / 3)

- OSSプラグイン開発者としての自分は、なるべく多くの環境をサポートしたいが、慣れないデータベースをローカル環境にインストールしたりするのは大変
- Dockerを利用しても手作業での環境切り替えはそれなりに手間がかかる



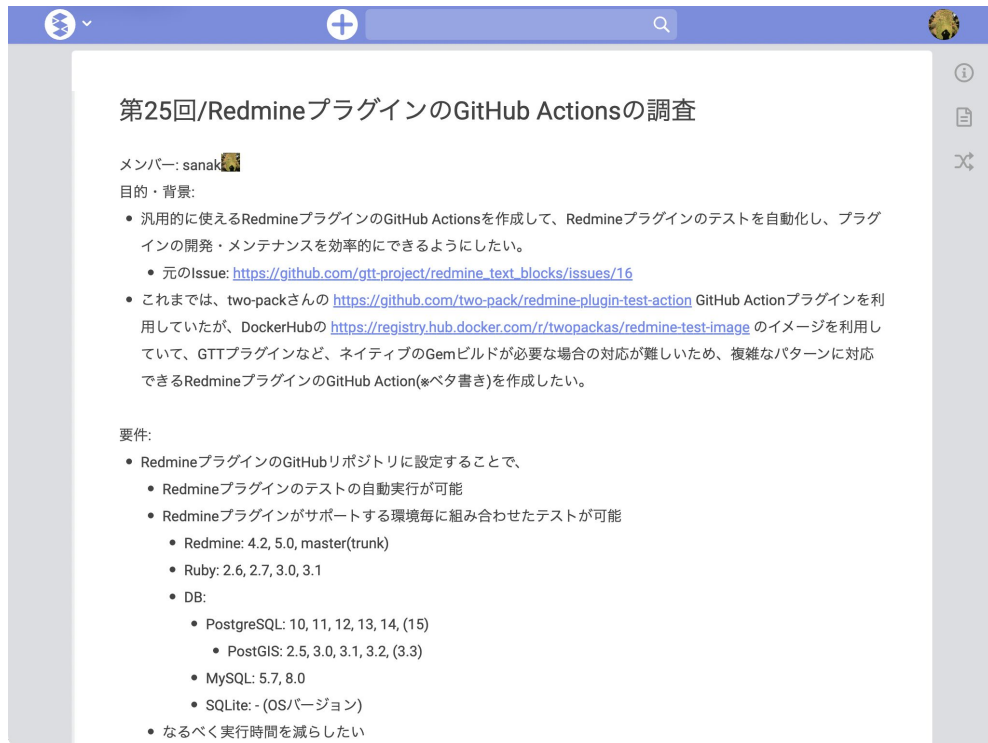
Redmineパッチ会でやっているようなテスト自動実行を

RedmineプラグインのGitHubリポジトリでもできるようにしよう！



プラグインのテスト自動化調査・対応 (1 / 3)

- 2022/07/30の第25回Redmineパッチ会の際に、調査開始
 - GitHub上のRedmineプラグインで、”.github/workflows”フォルダが存在し、テストを行うYAMLファイルが存在するものを先行事例として参考に
 - 他に@ishikawa999さんアドバイスで、@agileware-jpさんのCircle CI実装も参考に



第25回/RedmineプラグインのGitHub Actionsの調査

メンバー: sanak

目的・背景:

- 汎用的に使えるRedmineプラグインのGitHub Actionsを作成して、Redmineプラグインのテストを自動化し、プラグインの開発・メンテナンスを効率的にできるようにしたい。
 - 元のIssue: https://github.com/gtt-project/redmine_text_blocks/issues/16
- これまでは、two-packさんの <https://github.com/two-pack/redmine-plugin-test-action> GitHub Actionプラグインを利用していたが、DockerHubの <https://registry.hub.docker.com/r/twopackas/redmine-test-image> のイメージを利用して、GTTプラグインなど、ネイティブのGemビルドが必要な場合の対応が難しいため、複雑なパターンに対応できるRedmineプラグインのGitHub Action(*ベタ書き)を作成したい。

要件:

- RedmineプラグインのGitHubリポジトリに設定することで、
 - Redmineプラグインのテストの自動実行が可能
 - Redmineプラグインがサポートする環境毎に組み合わせたテストが可能
 - Redmine: 4.2, 5.0, master(trunk)
 - Ruby: 2.6, 2.7, 3.0, 3.1
 - DB:
 - PostgreSQL: 10, 11, 12, 13, 14, (15)
 - PostGIS: 2.5, 3.0, 3.1, 3.2, (3.3)
 - MySQL: 5.7, 8.0
 - SQLite: - (OSバージョン)
 - なるべく実行時間を減らしたい

プラグインのテスト自動化調査・対応 (2 / 3)

- 当初は業務(GTTプラグイン)の方での必要性から、DB毎にファイルを分けて作成
 - [Pull Request #26 · gtt-project/redmine text blocks](#)
 - DB種別をPostgreSQLに固定し、DBバージョンとその地理情報拡張機能(PostGIS)の組み合わせのみに対応したPull Request
 - [Pull Request #168 · gtt-project/redmine gtt](#)

プラグインのテスト自動化調査・対応 (3 / 3)

- その後、10月末から再開して、DB毎に分かれて重複していた記述を1つのファイルにし、RedMica、システムテスト(E2E、ブラウザテスト)にも対応
 - [Unifying each db tests to one by sanak · Pull Request #2 · sanak/redmine text blocks](#)
 - [Support system test by sanak · Pull Request #4 · sanak/redmine text blocks](#)
 - 対応中にRedmine本体側で最近のGem更新に由来する問題に遭遇して、Redmine本家の方にパッチを送ったりも ([#37867](#)、[#37883](#)、[#37884](#))

次のページからは具体的な設定手順を説明します。



GitHub Actionsでのテスト自動化設定 (1 / 8)

- 前提条件:
 - RedmineプラグインのリポジトリがGitHub上にあること
- 推奨:
 - GitHub ActionsのベースOSにUbuntu 22.04LTSを利用しているので、Linux環境のコマンドに慣れていると、カスタマイズが容易に
- 設定手順:
 - https://raw.githubusercontent.com/sanak/redmine_text_blocks/main/.github/workflows/test.yml から“test.yml”をダウンロード
 - ローカル環境でRedmineプラグインの作業用のブランチに切り替え後、“.github/workflows”フォルダを作成して、“test.yml”を追加



GitHub Actionsでのテスト自動化設定 (2 / 8)

- 設定手順(続き): “test.yml”をエディタで開き、適宜編集 (3~14行目)

```
env:  
  PLUGIN_NAME: ${{ github.event.repository.name }}  
  
on:  
  push:  
    branches:  
      - main  
      - master  
  pull_request:  
    branches:  
      - main  
      - master
```

プラグイン名がGitHubリポジトリ名と異なる場合は、固定値に変更

プッシュ・プルリクエスト時のターゲットブランチを指定



GitHub Actionsでのテスト自動化設定 (3 / 8)

- 設定手順(続き): “test.yml”の編集 (26～36行目)

```
matrix:  
  redmine: [  
    { repo: redmine, version: 4.2-stable },  
    { repo: redmine, version: 5.0-stable },  
    { repo: redmine, version: master },  
    { repo: redmica, version: stable-2.0 },  
    { repo: redmica, version: stable-2.1 },  
    { repo: redmica, version: master },  
  ]  
  ruby_version: ['2.6', '2.7', '3.0', '3.1']  
  db: [mysql, postgres, sqlite]
```

テストに利用するRedmineのリポジトリ(redmine/redmica)、バージョン(ブランチ名/バージョンタグ名)の組み合わせを指定

テストするRubyのバージョン、DB種別(mysql/postgres/sqlite)を指定



GitHub Actionsでのテスト自動化設定 (4 / 8)

- 設定手順(続き): “test.yml”の編集 (44～52行目)

```
exclude:  
- redmine: { repo: redmine, version: 4.2-stable }  
  ruby_version: '3.0'  
- redmine: { repo: redmine, version: 4.2-stable }  
  ruby_version: '3.1'  
- redmine: { repo: redmica, version: stable-2.0 }  
  ruby_version: '3.0'  
- redmine: { repo: redmica, version: stable-2.0 }  
  ruby_version: '3.1'
```

Redmineのバージョンによっては、Rubyバージョンをサポートしていない場合があるので、組み合わせから除外

<https://www.redmine.org/projects/redmine/wiki/RedmineInstall> を参照



GitHub Actionsでのテスト自動化設定 (5 / 8)

- 設定手順(続き):
 - “test.yml”の以降の行は必要に応じて編集し、終わったらGitに追加・コミットして、GitHubにプッシュ
 - GitHubのリポジトリ上でPull Requestを作成 (※フォークリポジトリで試す場合は、Pull Request作成時に自分のリポジトリのターゲットブランチ(master or main)を指定)

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: gtt-project/redmine_text_blo... base: main ← head repository: sanak/redmine_text_blocks compare: ftr/github-actions-test3-2

Choose a Base Repository

Filter repos

gtt-project/redmine_text_blocks

sanak/redmine_text_blocks

※デフォルトでアップストリーム(フォーク元)が指定されているので、自分のリポジトリの方に変更



GitHub Actionsでのテスト自動化設定 (6 / 8)

- 確認手順:
 - GitHubリポジトリでPull Requestを作成後、Actionsタブでテストが自動実行されていることを確認 (それなりに時間(10分程度)がかかります)

sanak/redmine_text_blocks Public

forked from gtt-project/redmine_text_blocks

Code Issues Pull requests **Actions** Projects Wiki Security

Actions New workflow

All workflows

Test



Management

Caches

All workflows

Showing runs from all workflows

42 workflow runs

	Event	
 Merge pull request #5 from sanak/f...	main	2 hours ago ... 8m 46s
 Update README.md CI badge	fix/update-readme-badge	2 hours ago ... 10m 42s

テスト実行中は黄色のローディングアイコン、テストがパス(正常終了)すれば緑の✓アイコン、失敗(異常終了)すれば赤の×アイコンが表示されます



GitHub Actionsでのテスト自動化設定 (7 / 8)

- 確認手順(続き):
 - テストが失敗(異常終了)した場合は、失敗したWorkflowのタイトルをクリックして、さらに失敗したJobを選択すると、どこでどんなエラーが発生したかが分かる。

The screenshot illustrates the process of troubleshooting a failed GitHub Action workflow. On the left, a workflow run is shown with a red 'X' icon and the title "Unifying each db tests to one". A red box highlights this title, and a red arrow points from it to the "Jobs" section. In the "Jobs" section, the first job, "redmine:4.2-stable ruby:2....", is also highlighted with a red box and a red arrow pointing to the log viewer on the right. The log viewer shows the execution of "Run tests" which failed with the error: "Error: Process completed with exit code 1." The log content is as follows:

```
redmine:4.2-stable ruby:2.6
db:mysql
failed yesterday in 1m 32s

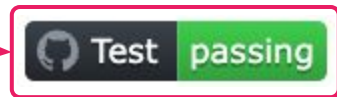
Run tests 4s
  `block in load_tests'
58 /usr/local/bundle/gems/railties-5.2.8.1/lib/rails/test_unit/runner.rb:50:in
  `each'
59 /usr/local/bundle/gems/railties-5.2.8.1/lib/rails/test_unit/runner.rb:50:in
  `load_tests'
60 /usr/local/bundle/gems/railties-5.2.8.1/lib/rails/test_unit/runner.rb:39:in
  `run'
61 /usr/local/bundle/gems/railties-5.2.8.1/lib/rails/test_unit/runner.rb:35:in
  `rake_run'
62 /_w/redmine_text_blocks/redmine_text_blocks/redmine/lib/tasks/redmine.rake
:171:in `block (4 levels) in <top (required)>'
63 /_w/redmine_text_blocks/redmine_text_blocks/redmine/lib/tasks/redmine.rake
:161:in `block (3 levels) in <top (required)>'
64 /usr/local/bundle/gems/rake-13.0.6/exe/rake:27:in `<top (required)>'
65 /usr/local/bin/bundle:23:in `load'
66 /usr/local/bin/bundle:23:in `<main>'
67 Tasks: TOP => redmine:plugins:test:units
68 (See full trace by running task with --trace)
69 Error: Process completed with exit code 1.
```



GitHub Actionsでのテスト自動化設定 (8 / 8)

- マージ:
 - テストがパスすれば、Pull Requestをマージ
- 仕上げ:
 - README.mdに以下を含めると、バッジが表示されて見栄えが良くなります。

```
[![Test] (https://github.com/ (GitHubユーザー名) / (GitHubリポジトリ名) /workflows/Test/badge.svg?branch=(ブランチ名:mainもしくはmaster)) ] (https://github.com/ (GitHubユーザー名) / (GitHubリポジトリ名) /actions/workflows/test.yml?query=branch%3A(ブランチ名:mainもしくはmaster))
```



補足

- Redmineプラグインでテストを記述しなくても、DBマイグレーションやZeitwerkチェックとしてだけでも有効
- システムテスト関連のコメントアウト箇所を外せば、システムテスト(E2E、ブラウザテスト)の実行も可能(ただし、2~3倍時間がかかります)
- Redmine本体のコメントアウト箇所を外せば、プラグインがRedmine本体の挙動に影響を与えないことの確認も可能(こちらはさらに時間がかかります)
- 応用例として、他のプラグインと組み合わせると動かなくなるといった現象の調査も可能



今後の課題

- ジョブが並列実行されるGitHub Actionsでも、初期状態のtest.yml(組み合わせ数:60)で7~11分ほどかかるので、テスト実行時間の短縮化を図る
 - GitHub Actions自体の機能が年々向上してきているので、そちらに期待したいというものあたり...
- システムテスト(E2E、ブラウザテスト)はChromeのインストールなどでさらに時間がかかるので、良い方法を検討する



まとめ

- Redmineパッチ会が、GitHub上でのRedmineテスト自動化を知るきっかけに
- Redmineプラグインでもテストは大切だけど、複数環境のテストを手作業でやるのは大変
- GitHub Actionsで、RedmineプラグインでもGitHub上のRedmineプラグインのテスト自動化が簡単にできる



謝辞

- Redmineパッチ会の方々:
 - [redmine-patch-meetup/redmine-dev-mirror](#)
- Redmineプラグインや関連プロジェクト開発者の方々:
 - @two-pack: [two-pack/redmine xlsx format issue exporter](#)
 - @nanego: [nanego/redmine datetime custom field](#)
 - @clear-code: [clear-code/redmine full text search](#)
 - @haru: [github.com/haru/redmine theme changer](#)
 - @cat-in-136: [cat-in-136/redmine scheduling poll](#)
 - @alexandermeindl: [AlphaNodes/redmine sudo](#)
 - @ishikawa999: [redmica/redmica ui extension](#)
 - @agileware-jp: [agileware-jp/redmine-plugin-orb](#)
 - @hanachin: [Redmineプラグイン開発とCIのmatrix - Speaker Deck](#)
 - @marius-balteanu: <https://gitlab.com/redmine-org/redmine>



Redmineパッチ会 第27回

- 次回は2022/11/19(土)の13:00～17:00に開催予定
- connpass URL: <https://redmine-patch.connpass.com/event/265634/>
- Redmineの機能改善や、テストに興味のある方、モブプロやってみいたい方や、もくもく作業したい方も、是非、ご参加ください！



The screenshot shows a Connpass event page for the 27th Redmine Patch Meeting. The main content area features a large red graphic of a person juggling blocks, with the text "PATCH REDMINE" below it. The event title is "Redmineパッチ会オンライン #27" and the description is "Redmine本体を改善していく会". The event date and time are "2022/11/19(土) 13:00～17:00". The page also includes a "このイベントに申し込む" (Sign up for this event) button and a "グループ" (Group) section with "Redmineパッチ会" and "メンバーです" (Member).

11月 19 Redmineパッチ会オンライン #27
Redmine本体を改善していく会

グループ

Redmineパッチ会

イベント数 27回
メンバー数 48人

開催前

2022/11/19(土)
13:00～17:00

 Googleカレンダー  icsファイル

ハッシュタグ : #RedminePatch



ご清聴ありがとうございました

- GitHubアカウント: [@sanak](#)
- GitHubリポジトリ:
[https://github.com/sanak/redmine text blocks](https://github.com/sanak/redmine_text_blocks)

