

Environmental Variable Programming

環境変数プログラミング

sanemat {AT} tachikoma.io

Examples

環境変数を取得してプログラミングする必要があることがある

- リポジトリにpush したくないデータ
 - GitHub の access token など
- 環境固有のデータ
 - RAILS_ENV など
 - CI 環境など

```
# リポジトリにpush したくないデータ
token = ENV['GITHUB_ACCESS_TOKEN']

# 環境固有のデータ
if ENV['SOME_VAR'].downcase == 'true'
  # your code
end
```

今回は、環境固有のデータ、特に CI 環境の話をする。dotenv などの話はしない。

CI 環境固有のデータは、build 周りのツールや、デプロイ周りのツールで必要となる。この中で出会った興味深いこと、作ったツールの話をする。

CI での環境変数の扱い

Environment Variables - Travis CI

```
CI=true
TRAVIS=true
CONTINUOUS_INTEGRATION=true
DEBIAN_FRONTEND=noninteractive
```

```
HAS_JOSH_K_SEAL_OF_APPROVAL=true
USER=travis (do not depend on this value)
HOME=/home/travis (do not depend on this value)
LANG=en_US.UTF-8
LC_ALL=en_US.UTF-8
RAILS_ENV=test
RACK_ENV=test
MERB_ENV=test
```

Environment variables - CircleCI

```
CIRCLECI=true
CI=true
CIRCLE_PROJECT_USERNAME=foo
    The username or organization name of the project being tested,
    i.e. "foo" in circleci.com/gh/foo/bar/123
CIRCLE_PROJECT_REPONAME=bar
    The repository name of the project being tested,
    i.e. "bar" in circleci.com/gh/foo/bar/123
CIRCLE_BRANCH=master
    The name of the branch being tested, e.g. 'master'.
```

いろんな環境変数

CI 環境によって違う

特にルールはない (と思う) あるのかも? ちょっと後で聞いてみたいけど

結構なんとなく似通ったいろいろ `CI=true` など。ツールによっては、`CI=true` だとカバレッジを coveralls に送る、など。

なんとなく感じ取ったルール

`truthy` のとき

- 何か文字列が入る

falsey のとき

- 環境変数の key 自体がなくなるパターン
- 環境変数の value が空文字列のパターン

問題

あるある 1

結構こういう、このキーである、という情報はどうにか有るのだが、こういう値を取りうる、という記述が欠けていることが多い。

しかも、これが CI 環境間で統一されていない。さらに、同じ CI 環境内でも、key によって違う。

あるある 2

ruby 固有のメソッドのこととしては、CI 環境的には空文字列は falsey だけど、Ruby 的には空文字列は truethy

あるある 3

travis-ci 決め打ちで作って、circle-ci で使いたくなる よくある

あるある 4

テストでいちいち考えなくちゃいけないが増える pull request やテスト自体が CI 環境上で動くので。環境変数消し漏れたり、戻し漏れたり、で動かないはずのものが動く分岐の方に行ってしまうたり。

env__branch

branch 情報を取り出したいことがよくあって、環境変数から取り出す部分を gem に切り出した。

Usage

```
require 'env_branch'

env_branch = EnvBranch.new
env_branch.branch? #=> true
env_branch.branch_name #=> 'your-branch-name'
```

Question

branch 名って git コマンドで取れるのでは? CI 環境によって違う

Travis-CI だと、環境変数から取るのが良い

checkout

なので、git branch しても branch 名はいない

helper

```
require 'env_branch/test_helper'

class TestExample < Test::Unit::TestCase
  extend ::EnvBranch::TestHelper

  def self.startup
    stash_env_branch
  end

  def self.shutdown
    restore_env_branch
  end
end
```

各 CI 環境での branch に関する環境変数を、いったん退避して、最後書き戻す。便利。

env_pull_request

pull request idを取り出したい。 <https://github.com/sanemat/node-boolify-string/pull/16> だとしたら、'16' これ。 pull request だった場合、ここに数字が入る。 GitHub の pull request に対して hook なりで何かをしたい場合、これを使ってリクエストする必要がある。

pull_request 番号を取り出したいことがよくあって、環境変数から取り出す部分を gem に切り出した。

```
require 'env_pull_request'

env_pull = EnvPullRequest.new
env_pull.pull_request? #=> true
env_pull.pull_request_id #=> 800

require 'env_pull_request/test_helper'

class TestExample < Test::Unit::TestCase
  extend ::EnvPullRequest::TestHelper

  def self.startup
    stash_env_pull_request
  end

  def self.shutdown
    restore_env_pull_request
  end
end
```

便利なので使ってください

対応している CI 環境

- env_branch
 - Travis-ci
 - CircleCI
- env_pull_request

- Travis-ci
- CircleCI
- Jenkins GitHub pull request builder plugin

余談

pull request でない場合

falsey の場合、環境変数の key 自体がない場合と、value が空文字列の場合があるといった。

引用

TRAVIS_PULL_REQUEST: The pull request number if the current job is a pull request, “false” if it’s not a pull request.

!???

CI 環境の環境変数、基本的には

- truthy のとき
 - 何か文字列が入る
- falsey のとき
 - 環境変数の key 自体がなくなるパターン
 - 環境変数の value が空文字列のパターン
 - 環境変数の value が “false” のパターン

そういうのにも env_branch や env_pull_request は対応済みです。なのでぜひ使って。その他 ci 環境は pull request ください。drone や wercker など。使う人が対応しようってことで。

楽しい環境変数プログラミングを。