

# Ruby(Rails)から カジュアルに Q4M を 使える何かを作ったお話

関口亮一

 ryopeko

 ryopeko

# 関口 亮一

 ryopeko

 ryopeko

- パーフェクトRuby 共著者
- デブサミ2014スピーカー
- ベストスピーカー賞 2位受賞
- 最近人間たちからは複数DBと呼ばれている



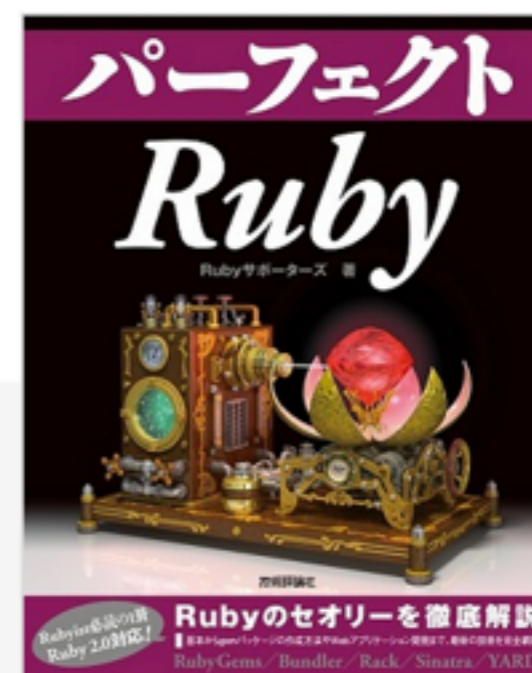
パーフェクト

## パーフェクトRuby

著者 Rubyサポーターズ 著

発売日 2014年11月1日

更新日 2014年11月1日



本書は、**2013年8月10日に発売された書籍**の電子版です。

**ご購入には会員登録・ログインが必要です**

[学校・法人等団体でのご利用について](#)

EPUB

3,200円

640ページ相当 EPUB

**11月1日発売予定**

1

0

0

0

ツイート

いいね!

+1

ブックマーク

みんな大好き

キューフォーエム!!!

ルビーで使うには  
ちょっとつらい...

ルビラーの世界では  
Redis 使ってるよ

# Sidekiq

Simple, efficient background processing for Ruby.

Upgrade to  
**Sidekiq Pro**  
for more great  
features!

What if **1** Sidekiq process could do the work of **20**  
Resque or DelayedJob processes?

 [Get the Source](#)

 [Documentation](#)

## Install It

Put sidekiq in your Gemfile:

```
gem 'sidekiq'
```

Install the gem with **Bundler**:

```
bundle install
```

## Call It

Place your worker classes in `app/workers`:

## Tweets

 Follow



**Sidekiq** @sidekiq 21 Oct  
Sidekiq 3.2.6 is released with minor  
bugfixes [github.com/mperham/sideki...](https://github.com/mperham/sidekiq)

 Show Summary



**David Stosik** @davidstosik 19 Oct  
Stuck on a tricky problem?  
Want to flip tables?  
Apparently, @sidekiq has that!  
[#RailsRumble pic.twitter.com/hgrVJCPcsL](https://pic.twitter.com/hgrVJCPcsL)

 Retweeted by Sidekiq

```
sto@Ubuntu-Carbon:~/dev/clonecall (master *  
(1754) $ rails c
```

# shinq

( ryoeko/shinq)



Worker and enqueueer for Q4M using the interface of ActiveJob — Edit

117 commits      1 branch      6 releases      1 contributor

branch: master shinq / +

rails 4.2.0.rc2

ryoeko authored 5 minutes ago      latest commit 3e00a3bb0a

bin	launch sequence	2 months ago
lib	Add worker life cycle	22 days ago
spec	Add spec for logger	19 days ago
.gitignore	first commit	2 months ago
.rspec	Remove --warnings option	2 months ago
.travis.yml	Change schema	a month ago
Gemfile	first commit	2 months ago
LICENSE.txt	first commit	2 months ago

Code

- Issues 0
- Pull Requests 0
- Wiki

- Pulse
- Graphs

Settings

SSH clone URL  
git@github.com:ryoeko/shinq

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

**shinq は ActiveJob の  
インターフェースを使ったライブラリ**

**ActiveJob ?**

branch: master rails / activejob / +

Added queue\_name\_delimiter attribute. ...

zv1n authored on 24 Sep

latest commit 11ab04b111

..		
lib	Added queue_name_delimiter attribute.	3 days ago
test	Added queue_name_delimiter attribute.	3 days ago
.gitignore	ActiveJob Integration Tests	2 months ago
CHANGELOG.md	Add 'activejob/' from commit '14f74a8331f94150dfce653224de8fc837797709'	3 months ago
MIT-LICENSE	Add 'activejob/' from commit '14f74a8331f94150dfce653224de8fc837797709'	3 months ago
README.md	Added RDoc for each Active Job adapter	a month ago
Rakefile	Enable emitting of warnings from ActiveJob tests.	11 days ago
activejob.gemspec	Bump globalid version	2 months ago

README.md

# Active Job -- Make work happen later

Active Job is a framework for declaring jobs and making them run on a variety of queueing backends. These jobs can be everything from regularly scheduled clean-ups, to billing charges, to mailings. Anything that can be chopped up into small units of work and run in parallel, really.

# ActiveJob

- Rails4.2 から入る予定のキューアダプター
- キューライブラリを差し替えしやすいようにするもの
- daemon 実行は各自頑張って

# shinq の使い方

**queue table を作る**

```
class CreateMyQueues < ActiveRecord::Migration
  def change
    create_table :my_queues, {id: false, options: "ENGINE=QUEUE"} do |t|
      t.string :job_id, null: false
      t.string :title
      t.text :desc

      t.datetime :enqueued_at, null: false
    end
  end
end
```



**Worker Class を作る**

```
1 class MyQueue < ActiveJob::Base
2   queue_as :my_queues
3
4   def perform(args)
5     p args
6   end
7 end
```

**generator もあるよ！**

```
% r g shinq:worker foo_queue title:string  
create db/migrate/20141212082227_create_foo_queues.rb  
create app/workers/foo_queue.rb
```

```
% rake db:migrate
== 20141212082227 CreateFooQueues: migrating =====
-- create_table(:foo_queues, {:id=>false, :options=>"ENGINE=QUEUE"})
   -> 0.0233s
== 20141212082227 CreateFooQueues: migrated (0.0234s) =====
```

enqueue する

```
[2] pry(main)> MyQueue.perform_later(title: 'foo', desc: 'barfoo')  
Enqueued MyQueue (Job ID: 69e96b31-ef52-4d87-bbb4-52a73366bb9c) to  
Shing(my_queues) with arguments: {:title=>"foo", :desc=>"barfoo"}  
=> #<MyQueue:0x007fccb9aaf568  
@arguments=[{:title=>"foo", :desc=>"barfoo"}],  
@job_id="69e96b31-ef52-4d87-bbb4-52a73366bb9c",  
@queue_name="my_queues">
```

dequeue する



```
% be shing --worker my_queue  
{}  
{ "job_id" => "6f860e1f-742e-4639-a68c-e9b2d856e19c",  
  "desc" => "barfoo", "enqueued_at" => 2014-10-30 21:1  
{ "job_id" => "d2f1156c-9ed1-40fe-8a4e-1fee819d1b33",  
  "desc" => "barfoo", "enqueued_at" => 2014-10-30 21:1  
{ "job_id" => "1e637f31-42f1-472e-89fd-f614dae5d0dc",  
  "desc" => "barfoo", "enqueued_at" => 2014-10-30 21:1
```


コマンドラインオプション

# コマンドラインオプション

- `-worker` 実行する worker name
- `-process` fork するプロセス数
- `-queue-timeout` `queue_wait` でタイムアウトする秒数
- `-d, -daemon` daemon として実行
- etc...

まとめ

# まとめ

- Ruby(Rails) で Q4M をカジュアルに扱う何かを作ったよ
- モダンなインターフェースだよ
- モダンすぎて4.2じゃないと使えないよ(今RC2)
-  ryopeko/shinq だよ