

First step of Roda

y-yagi 著

2018-04-22 版 発行

はじめに

本書の内容

本書は、Roda^{*1}という Web アプリケーション用のライブラリについて紹介した本です。

Roda は Ruby 製のフレームワークです。Ruby で Web アプリケーション用のライブラリといえ
ば、一般的には Ruby on Rails(以降 Rails)^{*2}が使われる事が多いと思います。

実際に、Rails を使うのが適切なケースが殆どでしょう。しかし、Rails は多機能なフレームワ
ークな為、実は使用が適切ではないケースもあると思います。

そんな時に、もし他のフレームワークの事を知っていれば、そのフレームワークを使う事も検討
出来たと思います。

残念ながら、Roda 自体の知名度はあまり高くない印象です。本書を通じて Roda というフレーム
ワークの存在自体が少しでも広まる事を願っています。

対象読者

本書の対象は、既に Ruby、Rails を使った事がある人を想定しています。そのため、Ruby 書き
方や、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。

^{*1} <http://roda.jeremyevans.net/>

^{*2} <http://rubyonrails.org/>

第1章

Roda とは

Roda は、Jeremy Evans^{*1}氏によって作られたライブラリです。

HP^{*2}やソースコードの README^{*3}では、Roda の事を "Roda is a routing tree web toolkit" と紹介しています。

この紹介の通り、Roda はルーティング機能を提供するライブラリです。Model-View-Controller(以降 MVC) パターンにおける Controller 部分の機能のみを提供しており、Rails のように MVC 全ての機能を提供しているわけではありません。

Ruby 製で同様のライブラリだと、Sinatra^{*4}があります。実際、Roda は Sinatra を参考に作られており、HP 上には Sinatra との比較についての説明が記載されています。

1.1 Sinatra と Roda

色々違いあるのですが、一番大きな違いとしてルーティングの書き方の違いがあげれます。

Sinatra はルーティングをフラットなリストで定義するのに対して、Roda は木構造で定義します。

リスト 1.1: sinatra

```
require 'sinatra/base'

class App < Sinatra::Base
  get '/' do
    # TOP ページ
  end

  get '/todos/:id' do
    @todo = Todo[params[:id].to_i]
    # TODO 表示
  end

  post '/todos/:id' do
```

^{*1} <https://github.com/jeremyevans>

^{*2} <http://roda.jeremyevans.net/>

^{*3} <https://github.com/jeremyevans/roda/blob/master/README.rdoc>

^{*4} <http://sinatrarb.com/>

```
@todo = Todo[params[:id].to_i]
# TODO 更新
end

delete '/todos/:id' do
  @todo = Todo[params[:id].to_i]
  # TODO 削除
end
end
end
```

リスト 1.2: roda

```
require 'roda'

class App < Roda
  plugin :all_verbs
  route do |r|
    r.root do
      # TOP ページ
    end

    r.is 'todos', Integer do |todo_id|
      @todo = Todo[params[:id]]

      r.get do
        # TODO 表示
      end

      r.post do
        # TODO 更新
      end

      r.delete do
        # TODO 削除
      end
    end
  end
end
end
```

一見すると Sinatra の方が URL と処理のマッピングわかりやすく良いのでは、と思われる方もいると思います。それはそれで正しい意見だと思います。しかし、木構造になっている事で、ネームスペースや変数の定義を DRY にする事が出来ます。

これはパスが一階層しかないとあまり恩恵が無いかもしれませんが、パスの階層が深くなればなるだけ、よりわかりやすさに差が出てくると思います。

なお、ルーティングに木構造を使用するライブラリは Roda がはじめてではありません。Cuba^{*5}というライブラリがこの方法をとっており、実際 Roda は Cuba を fork して開発されています^{*6}。

^{*5} <http://roda.jeremyevans.net/>

^{*6} Roda の GitHub のリポジトリをみると、"forked from sovereign/cuba"と表示されているのが確認できます

1.2 Plugin System

Roda の大きな特徴の一つに、Plugin System があります。

1.3 Faster

第2章

Roda でアプリケーションを作ってみよう

First step of Roda

2018 年 4 月 22 日 初版第 1 刷 発行
著 者 y-yagi
