

# **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<sup>\*1</sup>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)<sup>\*2</sup>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。

---

<sup>\*1</sup> <http://roda.jeremyevans.net/>

<sup>\*2</sup> <http://rubyonrails.org/>

## 第 1 章

# Roda について知ろう

### 1.1 Roda とは

Roda は、Jeremy Evans([@jeremyevans](https://github.com/jeremyevans))<sup>\*1</sup>氏によって作られたライブラリです。Jeremy Evans 氏は、データベースツールキットの Sequel<sup>\*2</sup>や、テンプレートエンジンの Erubi<sup>\*3</sup>の作者でもあります。特に Erubi は、Rails 5.1 から Rails 標準のテンプレートエンジンになっており、名前を聞いた事ある方も多いかもしれません。

最初のバージョン (2014/07/30 リリース) から現在まで、Roda は Jeremy Evans 氏個人によって開発が続けられています。

HP<sup>\*4</sup>やソースコードの README<sup>\*5</sup>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<sup>\*6</sup>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。

### 1.2 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に対して、Roda は木構造で定義します。

リスト 1.1: sinatra

---

<sup>\*1</sup> <https://github.com/jeremyevans>

<sup>\*2</sup> <http://sequel.jeremyevans.net/>

<sup>\*3</sup> <https://github.com/jeremyevans/erubi>

<sup>\*4</sup> <http://roda.jeremyevans.net/>

<sup>\*5</sup> <https://github.com/jeremyevans/roda/blob/master/README.rdoc>

<sup>\*6</sup> <http://sinatrarb.com/>

```
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
    @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
```

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
```

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<sup>\*7</sup>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<sup>\*8</sup>。

Roda は、Cuba と Sinatra という二つの異なるライブラリの思想や設計に影響を受けて、今のようになっています。

## 1.3 プラグインシステム

Roda の大きな特徴の一つに、プラグインシステムがあります。Roda は、各種機能をプラグインとして提供するようにして、コアの機能は最小限になるようにしています。例えばテンプレートのレンダリング処理もデフォルトでは使用出来るようになっておらず、明示的にプラグインを利用する必要があります。

プラグインは `plugin` メソッドを対象のプラグインを読む込みことで出来るようになっています。先ほどのコード例だと、`plugin :all_verbs` がそうです。all\_verbs プラグイン使用する事により、`patch` や `delete` のような HTTP リクエストメソッドと同じ名前のメソッドが使用出来るようになります。

これは Rails の原則の一つである"おまかせ"とは真逆で、使用したい料理を自分で明示的に選択する必要があります。そもそもどんな料理があるかがわからない最初のうちは、これは大変な作業かもしれません。

しかしこのシステムにより、不要な機能が読み込まれず、結果ライブラリの動作が高速になったり、何が問題があった際に確認すべきソースが少なくなったりと、少なくない恩恵があります。また、何か自分が独自に機能を追加したい場合も、プラグインとしてその機能を提供することで、簡単に機能を追加出来る、というメリットもあります。

## 1.4 Faster

---

<sup>\*7</sup> <http://roda.jeremyevans.net/>

<sup>\*8</sup> 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

## 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

---