

もちもち テクニク

Vol. 1 2014 Summer

Ember.js+Railsで
マイ妖怪図鑑を
つくる

あさくらひろし

Docker入門
danny



もちもちテクニック Vol.1 2014 SUMMER

あさくらひろし、danny 著

2014-08-17 版 もちもち堂 発行

目次

第 I 部	Ember.js+Rails でマイ妖怪図鑑をつくる	1
第 1 章	Rails + Ember をセットアップする	2
1.1	セットアップ	2
1.2	Rails のトップページを表示する	3
1.3	Ember のトップページを表示する	4
第 II 部	Docker 入門	5
第 2 章	Docker	6
2.1	Docker とは	6
2.2	確認環境	7
2.3	シェルの操作	7

第Ⅰ部

Ember.js+Rails でマイ妖怪図鑑をつくる

第1章

Rails + Ember をセットアップする

今回は基本の CRUD と検索ができる簡単なアプリケーションを作ります。完成イメージはこんな感じです。Heroku へ完成したアプリをアップしておきました、動作はこちらでご確認ください。

<http://ember-hyakkiyakou.herokuapp.com/>

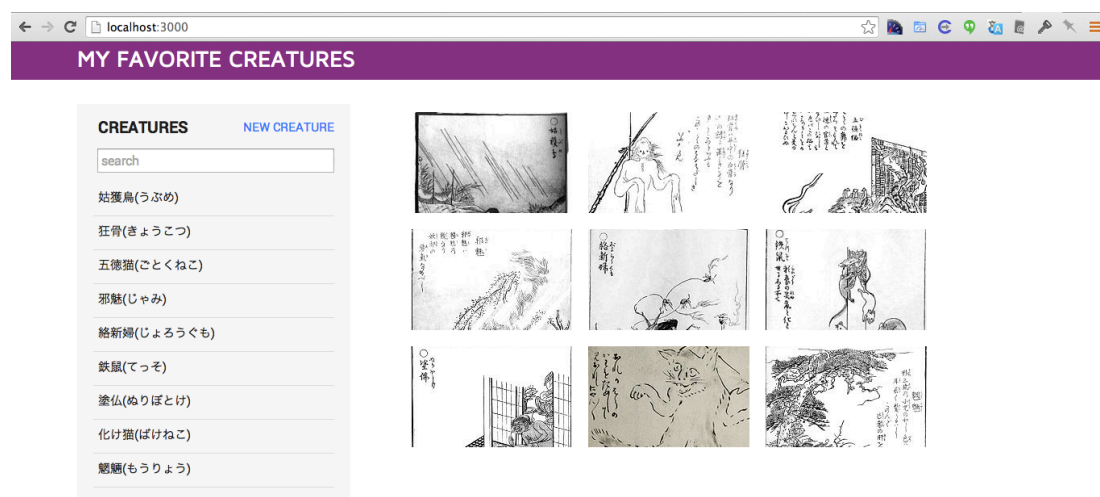


図 1.1 完成イメージ

1.1 セットアップ

まずは Rails アプリを作成します。DB はお好みでどうぞ。

```
$ rails new ember-hyakkiyakou -d postgresql --skip-bundle
$ cd ember-hyakkiyakou
$ bundle
$ bundle exec rake db:create
```

Ember と競合するので、以下のファイルから Turbolinks 関連のコードを削除します。

- Gemfile
- app/views/layouts/application.html.erb

次に ember-rails gem をインストールして、Ember ファイルをジェネレートします。今回は JavaScript エンジンに CoffeeScript を指定しています。

```
# Gemfile
gem 'ember-rails'
```

```
$ bundle
$ bundle exec rails g ember:bootstrap -n App --javascript-engine coffee
```

不要になった app/assets/javascripts/application.js を削除します。また、初期状態の application.js.coffee には jquery_ujs が足りないので追加しておきましょう。

```
# app/assets/javascripts/application.js.coffee
#= require jquery_ujs
```

1.2 Rails のトップページを表示する

Home コントローラーを作成します。Rails 側のテンプレートの中身は不要なので空にします。

```
$ bundle exec rails g controller home index --no-helper --no-assets
```

```
# app/views/home/index.html.erb
```

ルーティングを作成します。

```
# config/routes.rb
Rails.application.routes.draw do
```

```
root 'home#index'  
get '*path', to: 'home#index'  
end
```

1.3 Ember のトップページを表示する

Router ヘルパーティングを作成します。

```
// app/assets/javascripts/router.js.coffee  
App.Router.reopen  
  location: 'auto'  
  rootURL: '/'
```

Ember アプリケーションのテンプレートを作成します。

```
// app/assets/javascripts/templates/application.hbs  
<h1>Hello World</h1>  
{{outlet}}
```

動作確認してみましょう。シンプルなトップページが表示されると思います。

```
$ bundle exec rails s
```



図 1.2 トップページ

第 II 部

Docker 入門

第 2 章

Docker

2.1 Docker とは



図 2.1 Docker

<https://www.docker.com>

- LXC(Linux Containers) を使った仮想化ソフトウェアです。
- 全部を仮想化しないので、完全仮想化する仮想化ソフトウェアに比べてオーバーヘッドが少なくて済みます。
- ホスト OS のメモリーを共有するので、メモリーを効率良く使えます。
- ホスト OS からゲスト OS の起動が速いです。
- 差分で管理してるので、Dockerfile で何回もビルドする時に、すでに実行済みの部分はキャッシュされるので、ビルドが速いです。

2.2 確認環境

- Mac OS 10.9.4
- VirtualBox 4.3.14
- Vagrant 1.6.3
- Docker 1.1.1

今回はこの環境で確認してます。最新の VirtualBox が動く環境であれば、Mac OS X 以外の Windows や Linux などの OS でも動きます。

2.3 シェルの操作

説明に使われてるコマンドがどれで実行するか分かりやすいように、シェルのプロンプト表示を変えています。

VirtualBox を動かしてる OS での操作

```
$
```

VirtualBox で動いてる Docker のホスト OS での操作

```
vagrant@ubuntu:~$
```

Docker で動いてるコンテナでの操作

```
$bash-4.2#
```

もちもちテクニック Vol.1 2014 SUMMER

2014 年 08 月 17 日 v1.0.0 版発行

著 者	あさくらひろし、danny
デザイン	あさちゅん
イラスト	あさちゅん
発行所	もちもち堂

(C) 2014 Hiroshi Asakura