



Rails4 でつくる Instagram ビュー アーアプリ

あさくらひろし 著

2013-12-31 版 もちもち堂 発行

はじめに

本を手にとって頂いてありがとうございます。

2013 年の春に、Instagram に投稿された猫の写真だけを閲覧する iOS アプリ Nyanstagram を開発したのですが、その時の経験を活かして何かコミケで本を出してみたいなと思ったのが本書を作ったきっかけです。リリースされている Nyanstagram のように、Rails+iOS でのアプリ作成チュートリアルをと考えていたのですが、ちょっとボリューミーになりすぎるので、Rails アプリとして完結する形にして、Rails アプリのチュートリアル本として楽しんでいただければと思っています。

内容について

Rails4, Ruby2.1 で、モダンな Rails アプリを作成するためのチュートリアル本です。デスクトップ、スマホ、タブレットでも閲覧できるようレスポンシブな表示になるよう、Bootstrap を使います。Heroku ヘデプロイすることで、すぐにエンドユーザーへ公開できるようになっています。RSpec の書き方をちょっとだけかじることができます。Rubocop, RailsBestPractices, CodeClimate など、コード品質チェックについて知ることができます。作成済みのサンプルアプリが github リポジトリ、Heroku にアップされているので、動いているコードを確認しながら進めることができます。

動作環境について

Mac OSX 10.9 での動作を想定しています。

目次

はじめに	i
内容について	i
動作環境について	i
第1章 Nyanstagram 風の Rails アプリを作ってみよう	1
1.1 やること	1
1.2 必要なもの	1
1.3 対象読者	2
1.4 参考リポジトリ	2
第2章 Rails アプリをセットアップしよう	3
2.1 Rails アプリを新規作成する	3
2.2 必要な Gem をインストールする	4
2.3 データベースをセットアップする	5
2.4 Rails アプリを起動する	6
2.5 この章での変更点	7
第3章 Instagram API から写真情報を取得しよう	8
3.1 instagram gem をインストールする	8
3.2 Instagram クライアント ID を取得する	8
3.3 接続設定をする	9
3.4 キーワードで写真を取得しよう	10
3.5 この章での変更点	11
第4章 写真を一覧表示しよう	12
4.1 ホームコントローラーを作成する	12
4.2 ルーティングを更新する	12
4.3 写真一覧を表示する	13
4.4 この章での変更点	14
第5章 Heroku へデプロイしてみよう	15
5.1 heroku アプリを作成する	16
5.2 デプロイする	16

目次

5.3	heroku アプリへ環境変数を設定する	16
5.4	表示を確認する	17
第 6 章	Bootstrap で見た目を良くしよう	18
6.1	bootstrap-sass をインストールする	18
6.2	view を改良する	19
6.3	assets の静的ファイルの設定を変更する	20
6.4	デプロイする	21
6.5	この章での変更点	22
第 7 章	次のステップへ	23
第 8 章	Appendix1	24
8.1	Ruby 2.1.0 のインストール	24
8.2	Rails 4.0.2 のインストール	24
8.3	Bundler 1.5 のインストール	24
第 9 章	Appendix2	25

第1章

Nyanstagram 風の Rails アプリを作ってみよう

1.1 やること

Instagram に投稿された「猫」タグ付きの写真が一覧表示される Rails アプリを作ることを目指します。ねこにまみれたい。

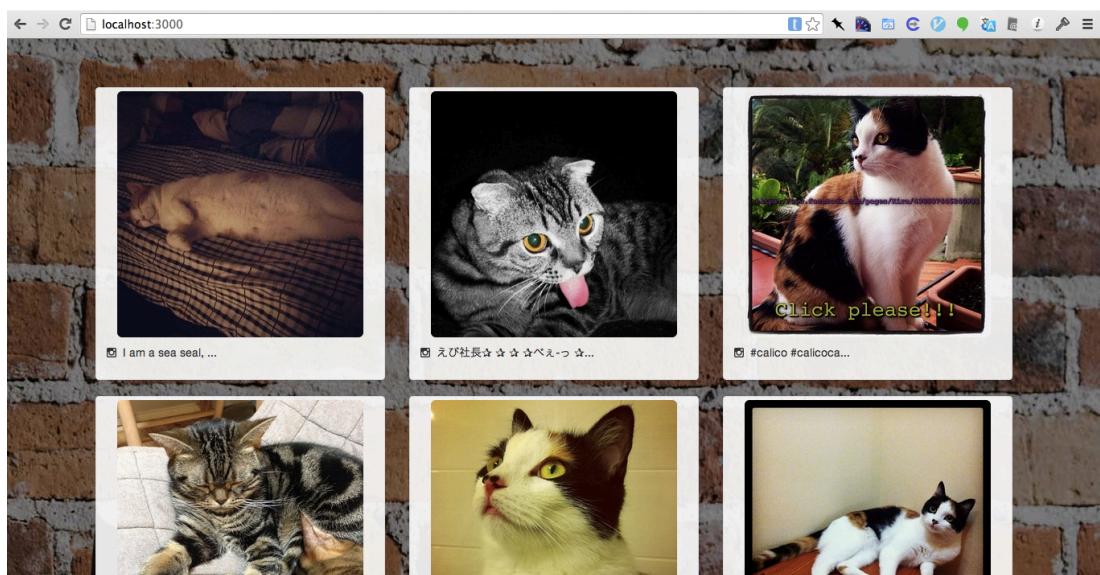


図 1.1 画像サンプル

1.2 必要なもの

- Ruby 2.1
- PostgreSQL 9.3
- Bundler 1.5
- Git 1.8

- Heroku Toolbelt
- Instagram アカウント
- Heroku アカウント

1.3 対象読者

ある程度 Rails を触ったことがある方を想定しています。

一度も触ったことのない方は、先に Ruby on Rails Tutorial を触つてみることをオススメします。

<https://github.com/kmuto/review/blob/master/doc/format.rdoc>

1.4 参考リポジトリ

参考用リポジトリを用意しておきました。コードの詳細はこちらからご確認ください。

<https://github.com/mochiz/nyansta-demo>

第2章

Rails アプリをセットアップしよう

2.1 Rails アプリを新規作成する

Rails アプリを新規作成します。Gem のインストールは別途行うので、`--skip-bundle` オプションをつけましょう。

```
$ rails new nyansta-demo --skip-bundle
  create
  create  README.rdoc
  create  Rakefile
  create  config.ru
  create  .gitignore
  create  Gemfile
  create  app
  create  app/assets/javascripts/application.js
  create  app/assets/stylesheets/application.css
  create  app/controllers/application_controller.rb
  create  app/helpers/application_helper.rb
```

都度 git リポジトリへ保存しておきましょう。

```
$ git init
Initialized empty Git repository in /Users/asakura/Sites/github/mochiz/nysta-demo/.git/
$ git add .
$ git commit -m 'initial commit'
[master (root-commit) 74a30c8] initial commit
 52 files changed, 702 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 Gemfile
 create mode 100644 README.rdoc
 create mode 100644 Rakefile
...
```

2.2 必要な Gem をインストールする

bundle コマンドで Gem をインストールしましょう。内容は以下のとおりです。

```
/Gemfile
source 'https://rubygems.org'
ruby '2.1.0'

gem 'rails', '4.0.2'
gem 'pg'

# assets
gem 'sass-rails', '~> 4.0.0'
gem 'uglifier', '>= 1.3.0'

# javascript
gem 'coffee-rails', '~> 4.0.0'
gem 'jquery-rails'
gem 'turbolinks'
gem 'jbuilder', '~> 1.2'

group :development, :test do
  gem 'rspec-rails'
end

group :development do
  gem 'better_errors'
  gem 'binding_of_caller'
  gem 'bullet'
  gem 'byebug'
  gem 'pry-rails'
  gem 'rails_best_practices'
  gem 'rubocop'
end

group :test do
  gem 'capybara'
  gem 'factory_girl_rails'
  gem 'poltergeist'
  gem 'shoulda'
  gem 'vcr'
  gem 'webmock'
end
```

bundle install コマンドを実行します。システムの Gem と切り離すため、--path オプションを追加しています。

```
$ bundle install --path vendor/bundle
Fetching gem metadata from https://rubygems.org/.....
```

```
Fetching additional metadata from https://rubygems.org/..
Resolving dependencies...
Using rake (10.1.1)
...
Your bundle is complete!
It was installed into ./vendor/bundle
```

2.3 データベースをセットアップする

今回のアプリではデータベースは必要ないのですが、heroku へデプロイするため PostgreSQL 用の設定に変更します。

```
/config/database.yml
development:
  adapter: postgresql
  encoding: unicode
  database: nyansta-demo_development
  pool: 5
  username: nyansta-demo
  password:

test:
  adapter: postgresql
  encoding: unicode
  database: nyansta-demo_test
  pool: 5
  username: nyansta-demo
  password:

production:
  adapter: postgresql
  encoding: unicode
  database: nyansta-demo_production
  pool: 5
  username: nyansta-demo
  password:
```

username に既存のユーザー名を使用するか、createuser コマンドでユーザーを作成してください。

```
$ createuser nyansta-demo -s
```

DB のマイグレーションを行います。

```
$ bundle exec rake db:create  
$ bundle exec rake db:migrate
```

2.4 Rails アプリを起動する

rails server コマンドを実行して、`http://localhost:3000` にアクセスしてみましょう。次のような画面が表示され、アプリの起動を確認できます。

```
$ bundle exec rails s
```

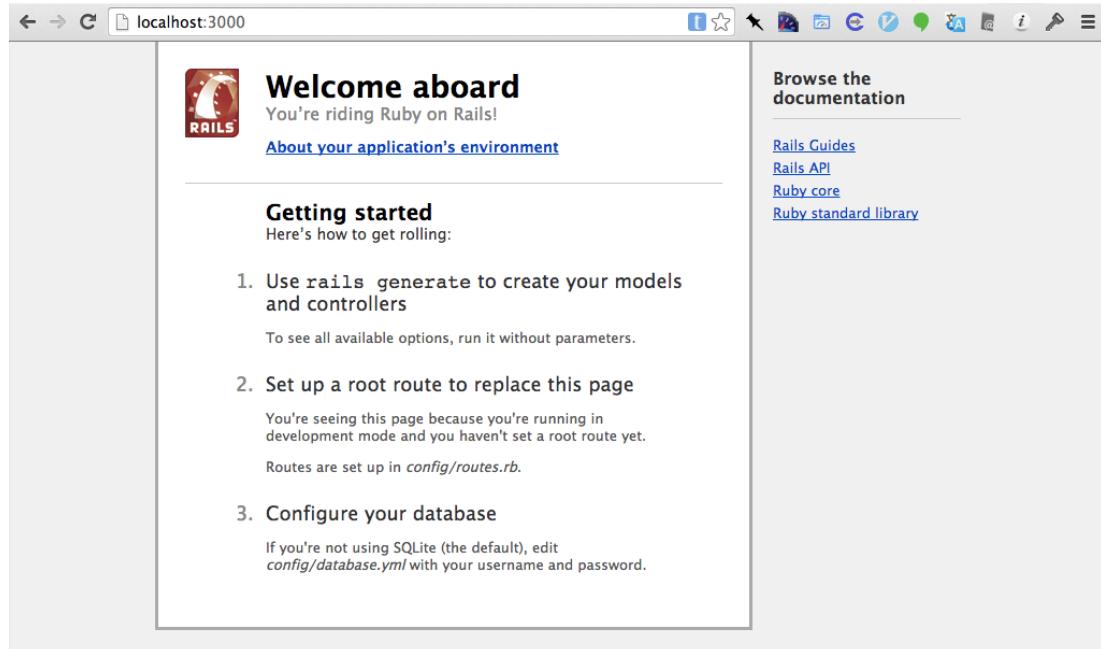


図 2.1 初期状態のホームページ

最後に変更を git リポジトリへコミットしておきましょう。

```
$ git add .  
$ git commit -m 'Instagram API から写真情報を取得できるようにした'
```

2.5 この章での変更点

ここまで変更は下記コミットログで確認できます。

<https://github.com/mochiz/nyansta-demo/commit/6a8b519>

コラム: 不要なファイルをジェネレートしないようにする

初期設定のままだと、コントローラー作成時に、不要な assets や helper が生成されて鬱陶しいので、不要なファイルはジェネレートしないよう以下のように設定しています。

```
/config/application.rb
...
module NyanstaDemo
  class Application < Rails::Application
    config.generators do |g|
      g.assets false      # アセットをジェネレートしない
      g.helper false     # ヘルパーをジェネレートしない
      g.view_specs false # ビュースペックをジェネレートしない
    end

    config.time_zone = 'Tokyo'          # タイムゾーンを Tokyo に変更
    config.i18n.default_locale = :ja # 言語を日本語に変更
  end
end
...
```

設定についてより詳しくは、公式のガイド等をご確認ください。

<http://guides.rubyonrails.org/configuring.html>

第3章

Instagram API から写真情報を取得しよう

3.1 instagram gem をインストールする

Instagram API 用に公式の Gem が提供されているのでこちらを利用します。

<https://github.com/Instagram/instagram-ruby-gem>

Gemfile に下記を追加し、bundle コマンドでインストールします。

```
/Gemfile
...
# instagram
gem 'instagram'

group :development, :test do
  gem 'dotenv-rails'
end
...
```

```
$ bundle
```

3.2 Instagram クライアント ID を取得する

API へアクセスするために、クライアント ID とクライアントシークレットが必要です。Instagram のアプリケーション管理画面から CLIENT ID, CLIENT SECRET を取得しましょう。

<http://instagram.com/developer/clients/manage/>

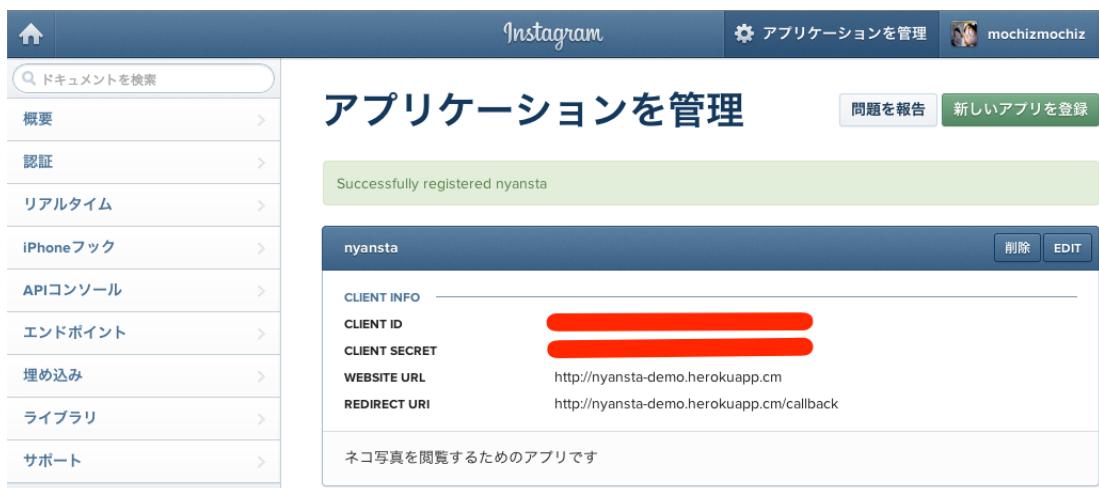


図 3.1 instagram アプリケーション管理画面

Instagram API への接続確認をしてみましょう。成功している場合は、images 等を含んだハッシュが返ります。

```
$ bundle exec rails c
[1] pry(main)> Instagram.configure do |config|
[1] pry(main)*   config.client_id = 'f46bbe97d5d44c5e8d1432919f4b86e5',
[1] pry(main)*   config.client_secret = '23f75449716b45e6a5055b816ac37d42',
[1] pry(main)* end
[2] pry(main)> Instagram.media_popular
=> [{"attribution"=>nil,
      "videos"=>
        ...
      "images"=>
        {"low_resolution"=>
          {"url"=>
            "http://distilleryimage10.s3.amazonaws.com/ecee78bc6f1d11e3aa990e4f40f44b5e_6.jpg",
            "width"=>306,
            "height"=>306},
          ...
        }
      }
    ]
[3] pry(main)>
```

3.3 接続設定をする

毎回 client_id, client_secret を指定しなくて良いよう、設定ファイルに保存しておきましょう。

```
/config/initializers/instagram.rb
require "instagram"
Instagram.configure do |config|
  config.client_id = ENV["INSTAGRAM_CLIENT_ID"]
```

```
config.client_secret = ENV["INSTAGRAM_CLIENT_SECRET"]
end
```

ここで ID をベタ書きしてしまうと、git リポジトリにクライアント ID, クライアントシークレットが保存されてしまってよろしくないので、環境変数から読み込むようにしています。

ここで、.bash や.zsh に環境変数を設定すると、他の環境へ影響が出てしまうので、このプロジェクトでのみ適用されるよう、dotenv を利用します。

前章で dotenv-rails gem はインストール済みなので、プロジェクトルートに.env ファイルを作成することでプロジェクトでのみ有効な環境変数を設定することができます。ここにクライアント ID, クライアントシークレットを設定しましょう。このファイルはリポジトリにプッシュしないよう、.ignore に追記しておきます。

```
/.env
INSTAGRAM_CLIENT_ID=[your client id]
INSTAGRAM_CLIENT_SECRET=[you client secret]
```

dotenv についてより詳しくは、<https://github.com/bkeepers/dotenv> を御覧ください。

3.4 キーワードで写真を取得しよう

tag_recent_media メソッドで、任意のタグがついた写真を取得することができます。日本語でタグを検索する場合は URI エンコードを忘れずにしましょう。

リスト 3.1: ネコ写真の取得

```
$ bundle exec rails c
[1] pry(main)> medias = Instagram.tag_recent_media(URI.encode('猫'))
[2] pry(main)> medias.first.images.standard_resolution
=> {"url"=>
  "http://distilleryimage4.s3.amazonaws.com/dfeed81a6ef411e3b5bb12dcee79a560_8.jpg",
  "width"=>640,
  "height"=>640}
```

instagram-ruby-gem についてより詳しくは、<https://github.com/Instagram/instagram-ruby-gem> を御覧ください。

最後に変更を git リポジトリへコミットしておきましょう。

```
$ git add .
```

```
$ git commit -m 'Instagram API から写真情報を取得できるようにした'
```

3.5 この章での変更点

ここまで変更は下記コミットログで確認できます。

<https://github.com/mochiz/nyansta-demo/commit/391cfa38>

第4章

写真を一覧表示しよう

4.1 ホームコントローラーを作成する

rails generate コマンドで、写真を表示するためのコントローラーとビューを作成しましょう。

```
$ bundle exec rails g controller home index
  create  app/controllers/home_controller.rb
  route   get "home/index"
  invoke  erb
  create    app/views/home
  create    app/views/home/index.html.erb
  invoke  rspec
  create    spec/controllers/home_controller_spec.rb
```

4.2 ルーティングを更新する

作成したコントローラーがトップ画面に表示されるよう、ルーティングを次のように更新します。

```
/routes.rb
NyanstaDemo::Application.routes.draw do
  root 'home#index'
end
```

Rails アプリを再起動して、<http://localhost:3000> にアクセスして、Home#index が表示されることを確認しましょう。

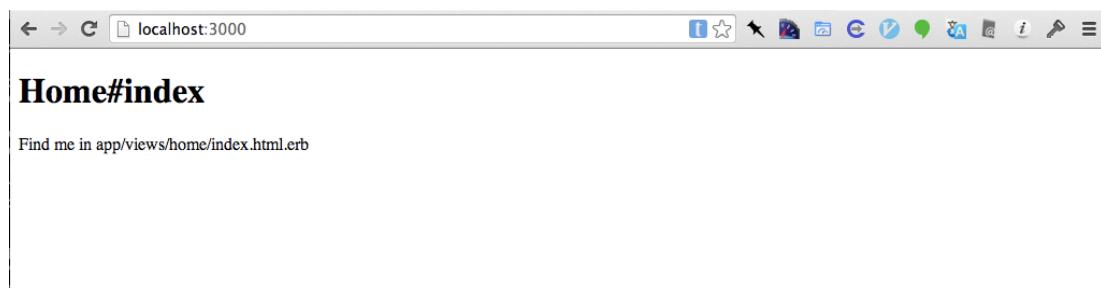


図 4.1 ホーム画面

4.3 写真一覧を表示する

画像を表示するため、コントローラーとビューを次のように更新します。

```
/app/controllers/home_controller.rb
class HomeController < ApplicationController
  def index
    @medias = Instagram.tag_recent_media(URI.encode('猫'))
  end
end
```

```
/app/views/home/index.html.erb
<%= render partial: 'media', collection: @medias %>
```

```
/app/views/home/_media.html.erb
<%= image_tag(media.images.low_resolution.url) %>
```

ホーム画面を更新してみましょう。可愛い猫の写真でいっぱいになっているはずです。

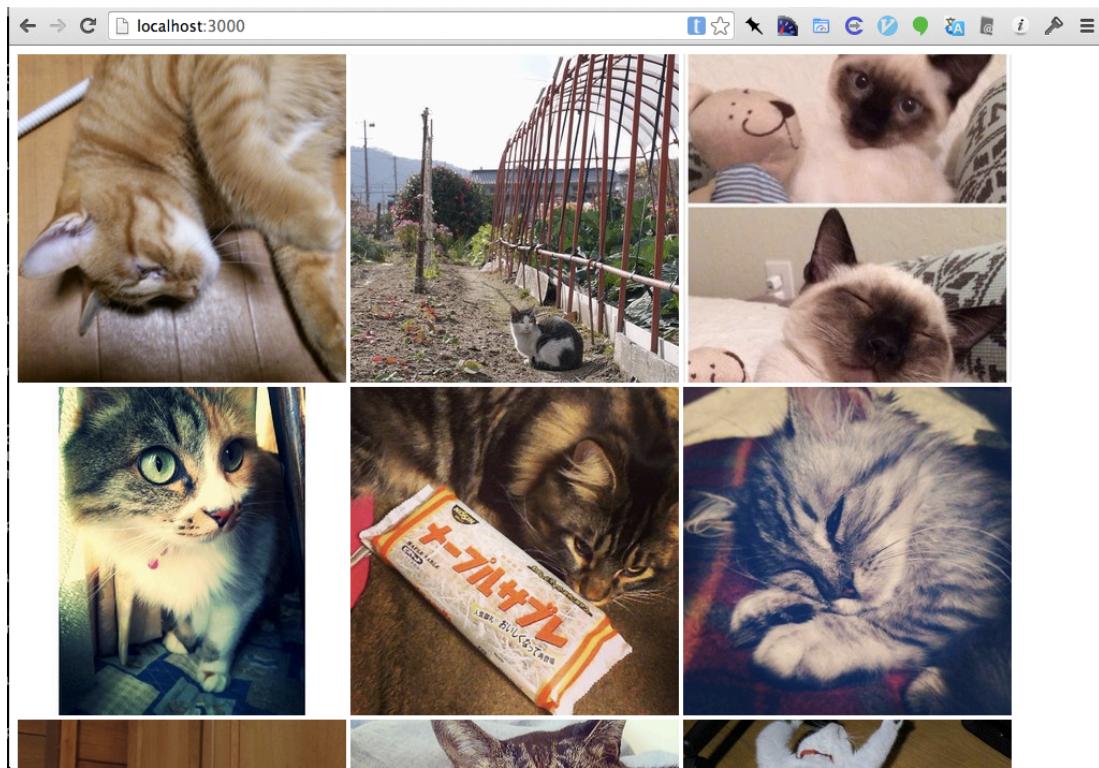


図 4.2 ホーム画面

最後に変更を git リポジトリへコミットしておきましょう。

```
$ git add .  
$ git commit -m '写真を一覧表示できるようにした'
```

4.4 この章での変更点

ここまで変更は下記コミットログで確認できます。

<https://github.com/mochiz/nyansta-demo/commit/7f88713e>

第5章

Heroku ヘデプロイしてみよう

heroku コマンドには、Heroku Toolbelt が必要なので、まだの人はインストールしておきましょう。

<https://toolbelt.heroku.com/>

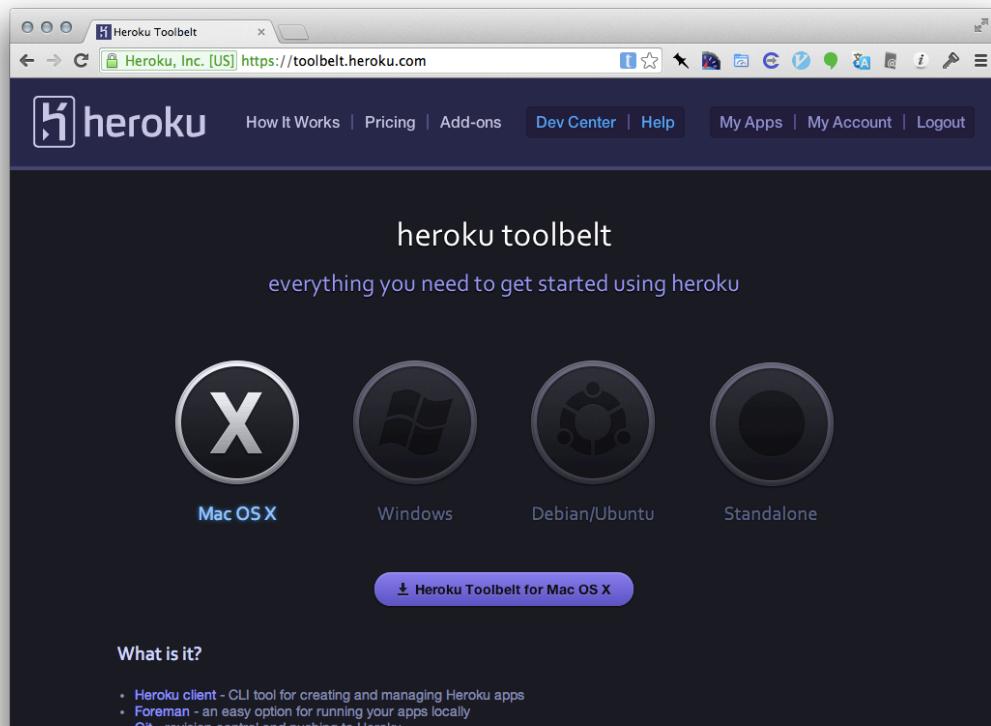


図 5.1 heroku toolbelt

5.1 heroku アプリを作成する

heroku create コマンドでアプリを作成します。

```
$ heroku create
Creating dry-everglades-1554... done, stack is cedar
http://dry-everglades-1554.herokuapp.com/ | git@heroku.com:dry-everglades-1554.git
Git remote heroku added
```

5.2 デプロイする

git push で heroku ヘデプロイします。

```
$ git push heroku master
Initializing repository, done.
Counting objects: 100, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (87/87), done.
Writing objects: 100% (100/100), 19.49 KiB | 0 bytes/s, done.
Total 100 (delta 21), reused 0 (delta 0)

-----> Ruby app detected
-----> Compiling Ruby/Rails
-----> Using Ruby version: ruby-2.1.0
...
```

5.3 heroku アプリへ環境変数を設定する

heroku config:add コマンドで、InstagramAPI 用の環境変数を設定します。

```
$ heroku config:add INSTAGRAM_CLIENT_ID=[YOUR_CLIENT_ID]
$ heroku config:add INSTAGRAM_CLIENT_SECRET=[YOUR_CLIENT_SECRET]
$ mochiz-mac% heroku config | grep INSTAGRAM
INSTAGRAM_CLIENT_ID:     xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
INSTAGRAM_CLIENT_SECRET:  xxxxxxxxxxxxxxxxxxxxxxxxx
```

5.4 表示を確認する

heroku open コマンドでサイトの表示を確認してみましょう。ねこまみれなはずです。

```
$ heroku open
```

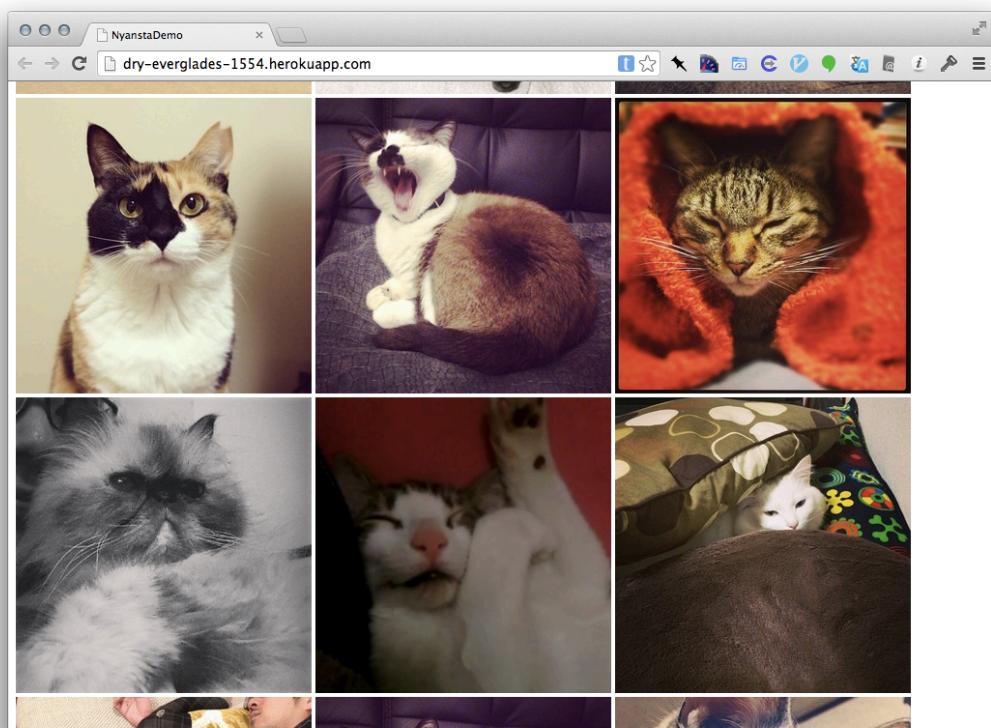


図 5.2 トップ画面

第6章

Bootstrap で見た目を良くしよう

見た目を良くするため、Bootstrap を組み込みましょう。

<http://getbootstrap.com/>

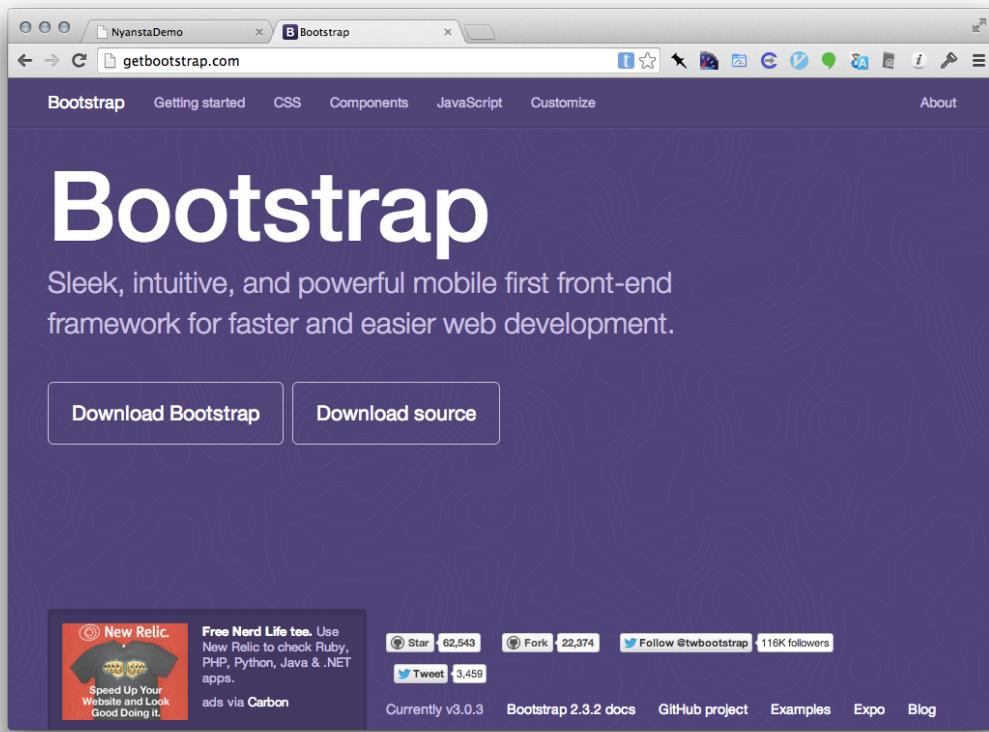


図 6.1 Bootstrap

6.1 bootstrap-sass をインストールする

Rails + Bootstrap の Gem はいくつかありますが、今回は bootstrap-sass を使用します。

Gemfile に bootstrap-sass と、ついでに font-awesome-rails を追加しましょう。
bootstrap3 を使いたいので、バージョンを指定します。

```
/Gemfile
...
# css
gem 'bootstrap-sass', '~> 3.0.3.0'
gem 'font-awesome-rails', '~> 4.0.3.0'
...
```

bundle コマンドでインストールし、スタイルシートとビューを変更します。

```
/app/assets/stylesheets/application.css.scss
@import "bootstrap";
@import "font-awesome";
```

```
/app/views/layouts/application.html.erb
...
<body>
<div class="container">
  <%= yield %>
</div>
</body>
...
```

6.2 view を改良する

ちょっとオシャレっぽくするために、背景画像やサムネイル表示を追加してみましょう。

```
/app/assets/stylesheets/nyansta.css.sass
@import "bootstrap";
@import "font-awesome";
@import "nyansta";
```

/app/assets/images/bg.jpg に適当な画像を設置してください。

```
/app/assets/stylesheets/nyansta.css.sass
body
  background: image-url('bg.jpg') no-repeat center center fixed
  padding: 60px 60px
  -webkit-background-size: cover
  -moz-background-size: cover
  -o-background-size: cover
  background-size: cover

.thumbnail
  background: #fff
  background: rgba(255,255,255,0.9)
```

```
/app/views/home/_media.html.erb
<div class="col-xs-12 col-sm-6 col-md-4">
  <div class="thumbnail">
    <%= link_to(media.link) do %>
      <%= image_tag(media.images.low_resolution.url, class: 'img-rounded') %>
    <div class="caption">
      <p>
        <i class="fa fa-instagram">&nbsp;</i>
        <%= truncate(media.caption.text, length: 20) if media.caption %>
      </p>
    </div>
    <% end %>
  </div>
</div>
```

6.3 assets の静的ファイルの設定を変更する

heroku 環境 (production 環境) で assets ファイルが有効になるよう、静的ファイルの設定を true に変更します。

```
MyansigðemvirAppmata/produonfigurb do
  ...
  config.serve_static_assets = true
  ...
end
```

変更を git リポジトリへコミットしておきましょう。

```
$ git add .  
$ git commit -m 'Bootstrap で見た目を良くした'
```

6.4 デプロイする

heroku へデプロイしてみましょう。いい感じにねこまみれてるはず。

```
$ git push heroku master  
$ heroku open
```

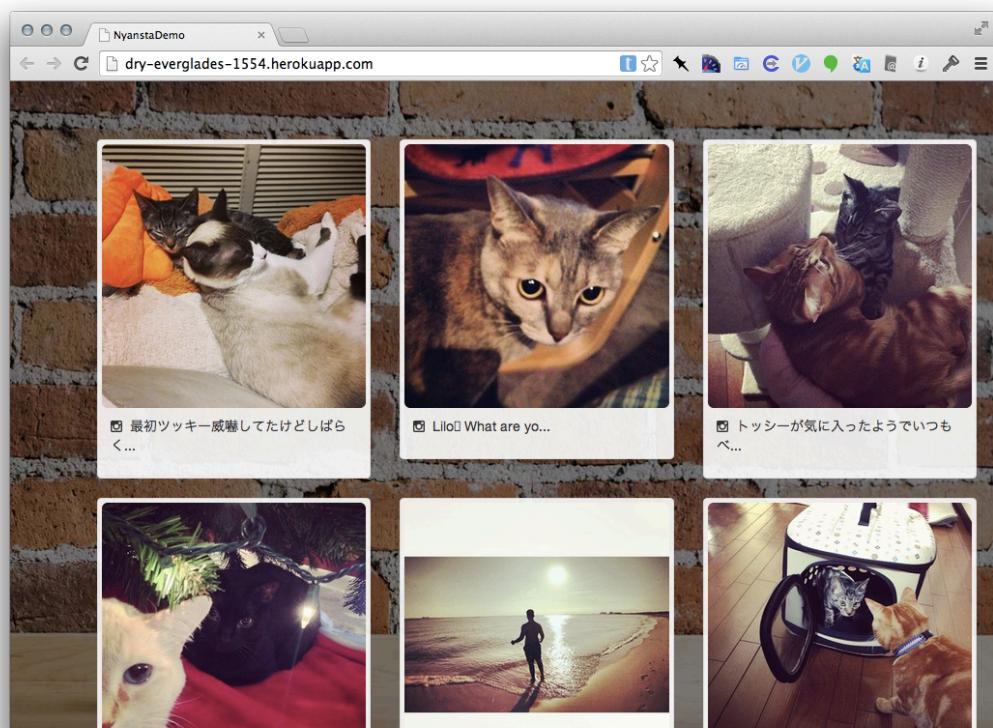


図 6.2 ねこにまみれたトップ画面

6.5 この章での変更点

ここまで変更は下記コミットログで確認できます。

<https://github.com/mochiz/nyansta-demo/commit/8edec1a3>

第7章

次のステップへ

だいぶ駆け足でしたが、ページと時間の都合で本書がカバーする内容はここまでとなります。
まだまだ足りないところがあるので、興味がある方は次の内容も実装してみましょう。

- Bootstrap
 - ヘッダー、フッターを追加する
 - ナビゲーションバーを追加する
- Instagram API
 - 任意のタグで検索できるようにする
 - 写真を次々ロードするようにする
 - いいねできるようにする

第8章

Appendix1

- 8.1 Ruby 2.1.0 のインストール
- 8.2 Rails 4.0.2 のインストール
- 8.3 Bundler 1.5 のインストール

第9章

Appendix2

Instagram API のアクセス ID の取得手順

Rails4 でつくる Instagram ビューアーアプリ

2013 年 12 月 31 日 v1.0.0 版発行

著 者 あさくらひろし

発行所 もちもち堂

(C) 2013 Hiroshi Asakura