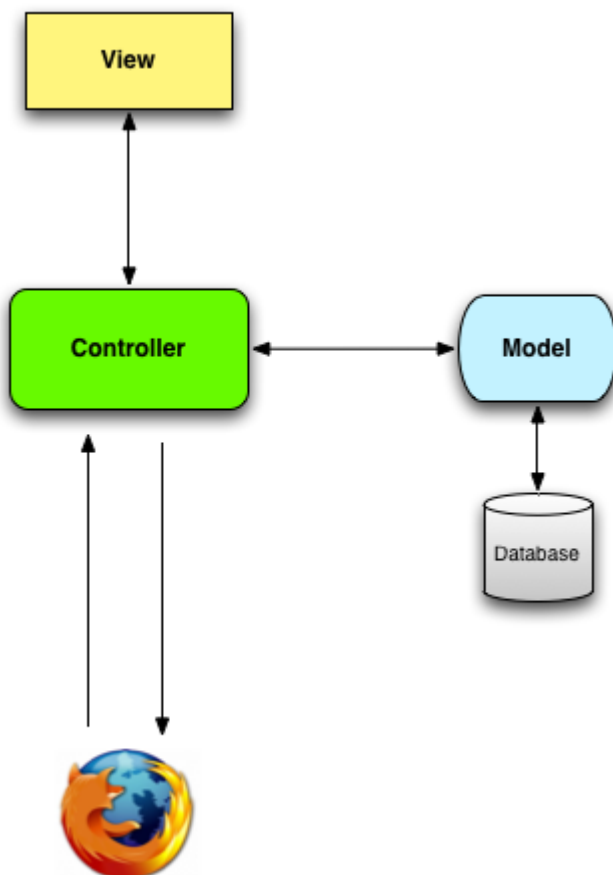


- 章別のメモ、分からない点など

chapter 1(7/20-7/23)

1. MVCの構造

Railsアプリと通信する際、ブラウザはWebサーバーにrequestを送信し、これはrequestを処理する役割を担っているRailsのcontrollerに渡されます。controllerは、場合によってはすぐにviewを生成してHTMLをブラウザに送り返します。動的なサイトでは、一般にcontrollerは(ユーザーなどの)サイトの要素を表しており、データベースとの通信を担当しているRubyのオブジェクトであるmodelと対話します。modelを呼び出した後、controllerは、viewを描画し、完成したWebページをHTMLとしてブラウザに返します。



1. git

Gitにプロジェクトのファイルをaddすると、最初はStaging)という一種の待機用リポジトリに置かれ、コミットを待ちます。安全のため、いきなりコミットしないようになっていますのです。ステージングの状態を知るにはstatusコマンドを使います。

- `git commit -a -m`について

`git add -A`を実行することもできますが、`git commit`には現存するすべてのファイルへの変更を一括でコミットする-aフラグがあります。このフラグは非常によく使われます。

- 便利なシェルコマンド -- `ll app/controllers/*_controller.rb`
- `heroku run rails console`

- `heroku run ruby -v`

(ruby-2.6.3, rails-5.2.0 bundler-2.0.2が現状ロバスト)

Chapter2(7/23,24)

1. @users=User.allで@usersには、Userテーブルのレコードが一つ一つ配列して入ることになる。
2. REST(REpresentational State Transfer)

インターネットそのものやWebアプリケーションなどの、分散・ネットワーク化されたシステムやアプリケーションを構築するためのアーキテクチャのスタイルの1つ RailsアプリケーションにおけるRESTとは、アプリケーションを構成するコンポーネント (ユーザーやマイクロポストなど) をリソースとしてモデル化することを指します。これらのリソースは、リレーショナルデータベースの作成/取得/更新/削除 (Create/Read/Update/Delete: CRUD) 操作と、4つの基本的なHTTP requestメソッド (POST/GET/PATCH/DELETE) の両方に対応しています。

3. `routes.rb`の`resources`

- 基本的な7つのアクションとそれに対するルーティングを自動で与える

4. `heroku run rails db:migrate`

Chapter3(7/24)

1. `gemfile`には必ずrubyとそのバージョンを明記する
 - `heroku run ruby -v`をすれば分かるが、デフォルトでheroku側のrubyのバージョンとこちらのrubyのバージョンが異なっている
 - エラー出たらログちゃんと読もう (gemにbootsnap入ってないよとか)
2. HTTP methodについて

HTTP (HyperText Transfer Protocol) には4つの基本的な操作があり、それぞれGET、POST、PATCH、DELETEという4つの動詞に対応づけられています。

3. `routes.rb`

- `root "con.#act."は"/" -> "con.#act."`の略記形
- `get "con./act" -> "con.#act."`のとき`get "con./act."`と略記できる。

Chapter4(7/24,25)

1. hash 次の二つは等価

メソッド呼び出しの丸カッコは省略可能。 `stylesheet_link_tag('application', { media: 'all', 'data-turbolinks-track': 'reload' })`
`stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-track': 'reload'`

Chapter5(7/25)

1. 名前付きルートパスの設定 `get 'static_pages/help' -> get '/help', to: 'static_pages#help'`とすれば

help_path -> '/help' help_url -> 'http://www.example.com/help' という結果が得られる。

Chapter6(7/25,26)

1. コントローラ名には複数形を用い、モデル名には単数形を用いる。
2. モデルのインスタンスはレコード！
3. callback method: ある特定の時点で呼び出されるメソッド
4. index: 多数のデータが存在するときの検索効率を向上させる要素。本の索引のイメージ

Chapter7(7/26)

- css mixin unknown
1. `routes.rb`に`resources :[controller名]`を付与することで 基本機能に対するURLがすべて付与される。
 2. debugが終わったらdebuggerメソッドは削除
 3. `form_for`: Userモデルのオブジェクト（レコード, @user）を取り込み、そのレコードの属性を使ってフォームを構築する。
 4. railsはnameプロパティの値を使って、初期化したハッシュを、params変数経由で構成する。
 5. `user_params`という外部メソッドを`params[:user]`の代わりに使う -> Strong Parameters
 6. `redirect_to @user = redirect_to user_url(@user)`
 7. `puma.rb`はデフォルトを全消して、チュートリアルのにやつに置き換えればok

Chapter8(7/27)