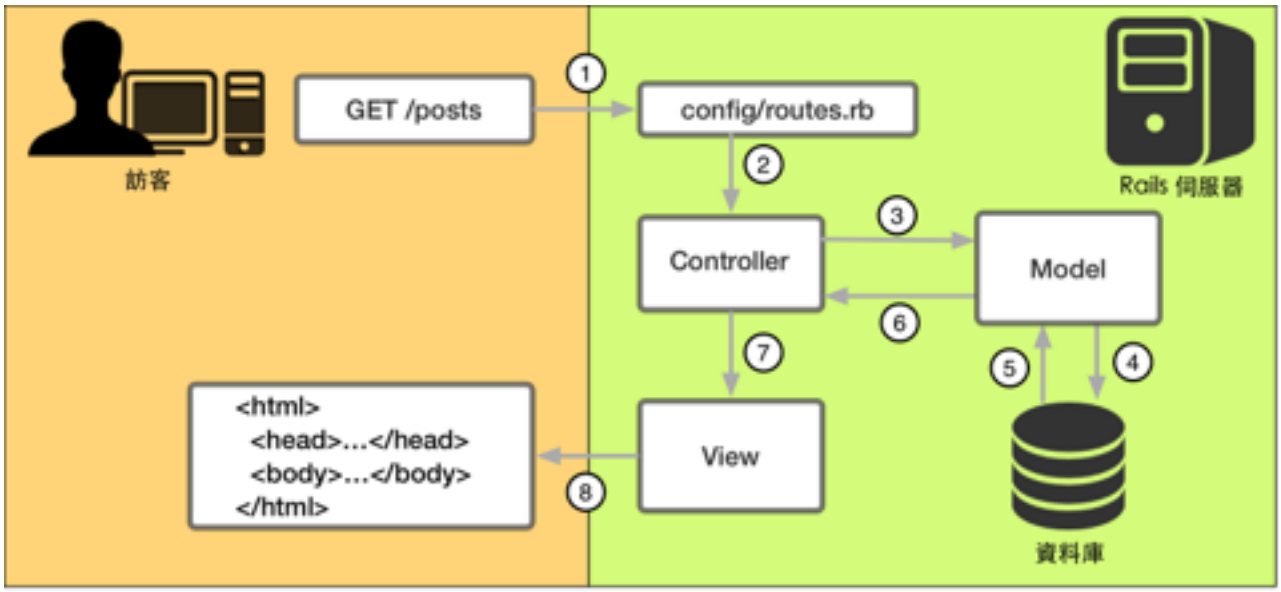
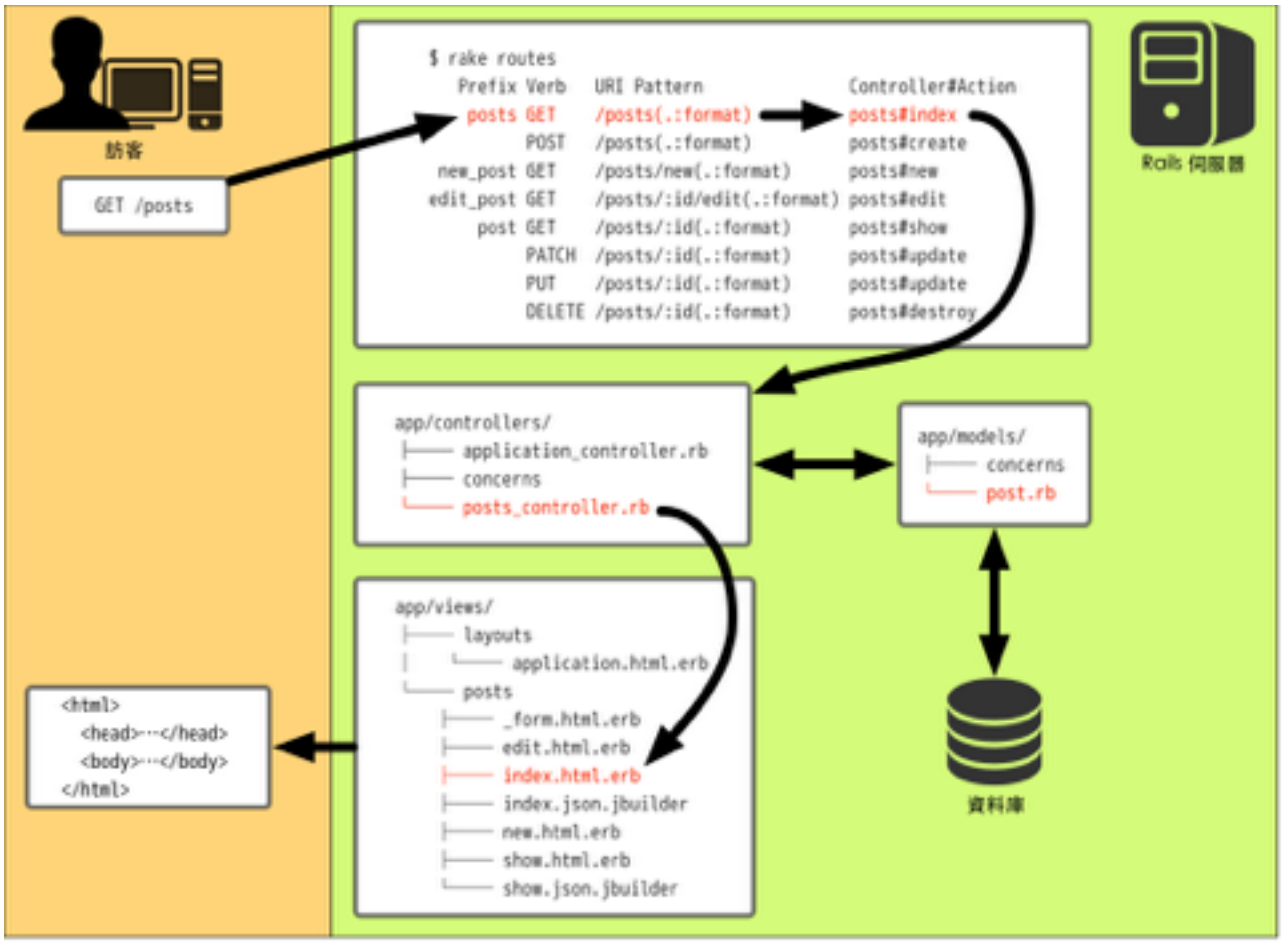
中華電信學院

Ruby on Rails 重點筆記

by 五倍紅寶石股份有限公司 - 11/24/14

Rails MVC 運作

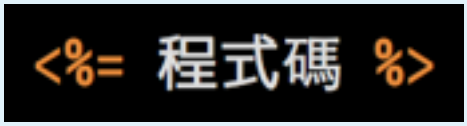
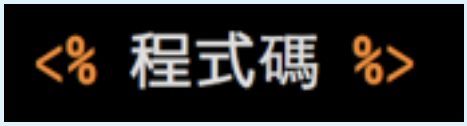
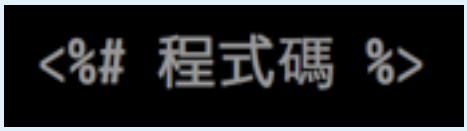
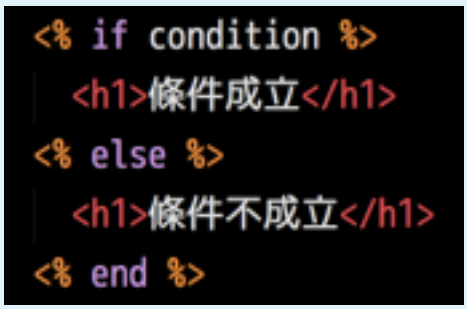
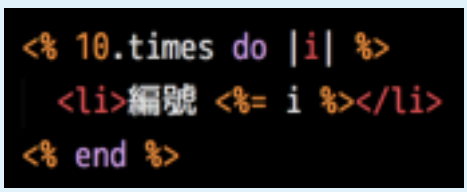

訪客
GET /posts
1
config/routes.rb
2
Rails 伺服器
Controller
3
Model
6
7
5
4
<html>
<head>...</head>
<body>...</body>
</html>
8
View
資料庫


訪客
GET /posts
$ rake routes
Prefix Verb URI Pattern Controller#Action
posts GET /posts(.:format) posts#index
POST /posts(.:format) posts#create
new_post GET /posts/new(.:format) posts#new
edit_post GET /posts/:id/edit(.:format) posts#edit
post GET /posts/:id(.:format) posts#show
PATCH /posts/:id(.:format) posts#update
PUT /posts/:id(.:format) posts#update
DELETE /posts/:id(.:format) posts#destroy
Rails 伺服器
app/controllers/
application_controller.rb
concerns
posts_controller.rb
app/models/
concerns
post.rb
app/views/
layouts
application.html.erb
posts
_form.html.erb
edit.html.erb
index.html.erb
index.json.jbuilder
new.html.erb
show.html.erb
show.json.jbuilder
<html>
<head>···</head>
<body>···</body>
</html>
資料庫

Rails 目錄結構

app	網站主要內容，在開發過程中最常訪問的資料夾，MVC 的設計也在這個資料夾內。
config	Rails 的設定檔，包括資料庫、網址路由等，第三方 Gem 的設定檔也會在這。
db	存放資料庫 schema 與遷移檔（migration file）
lib	存放用於擴展你的網站的程式碼，或是可獨立於網站外的模組化程式碼。
vendor	存放第三方程式碼。
public	存放靜態檔案，例如 404 頁面、favicon、robots.txt 與壓縮過的 JS 與 CSS 檔。
log	網站的紀錄檔案。
config.ru	Rack 的設定檔。
test	各種網站的單元測試
tmp	暫存檔，例如 pid、session、cache 等。

ERB 樣板

語法	說明
	執行並顯示
	執行但不顯示
	不執行也不顯示
	if else 用法
	block 用法

routes.rb

```
root 'pages#home'
root to: 'pages#home'
# root GET / pages#home

get 'about' => 'pages#about'
get :about, to: 'pages#about'
get :about, controller: :pages
get :about, controller: :pages, action: :about
match :about, controller: :pages, action: :about, via: :get
# about GET /about pages#about

scope controller: :pages do
  get :about, :contact
end
get :about, :contact, controller: :pages
#   about GET  /about   pages#about
# contact GET  /contact pages#contact

get 'about/me'
get 'about/me' => 'about#me'
get 'about/me', to: 'about#me'
get 'about/me', controller: :about, action: :me
# about_me GET /about/me about#me

resources :posts
#     posts GET    /posts          posts#index
#           POST   /posts          posts#create
#  new_post GET    /posts/new      posts#new
# edit_post GET    /posts/:id/edit posts#edit
#      post GET    /posts/:id      posts#show
#           PATCH  /posts/:id      posts#update
#           PUT    /posts/:id      posts#update
#           DELETE /posts/:id      posts#destroy
```

Layout 版型

- 位於 app/views/layouts/。
- 預設 layout 為 application.html.erb。
- view 的內容會在 <%= yield %> 展開。

設定方式

```
class PagesController < ApplicationController
  # pages controller 下所有的 action 都會套上, app/views/layouts/foo.html.erb
  layout 'foo'

  def home
    # 覆蓋設定, 改套用 app/views/layouts/bar.html.erb
    render layout: 'bar'
  end
end
```

Partial 局部樣板

- 位於 app/views/*/。
- 檔名皆以底線開頭，使用時並不加底線。

語法	說明
`<%= render 'shared/navbar' %>`	展開 app/views/shared/_navbar.html.erb
`<%= render 'navbar' %>`	展開 app/views/CONTROLLER/ _navbar.html.erb
`<%= render 'navbar', var: 'hello' %>`	傳入區域變數
`<!-- 令變數 post 是 Post 物件 -->` `<%= render post %>`	等同於 `<%= render 'posts/post', post: post %>`

`<!-- 令變數 posts 是 Post 物件集合 -->`
`<%= render posts %>`
	等同於
	`<% posts.each do |post| %>`
	`  <% render post %>`
	`<% end %>`

View Helper

URL Helper

- 根據 config/routes.rb 產生的路由表，有相對應的 URL helper。
- 可由 rake routes 查看 prefix。
- 方法命名為 xxx_path 與 xxx_url。
- 皆可在 controller 與 view 中使用。

以下表為例：

```
Prefix Verb URI Pattern      Controller#Action
  root GET  /                pages#home
  home GET  /home(.:format)  pages#home
 about GET  /about(.:format) pages#about
```

產生的 URL helper 有 root_path、root_url、home_path、home_url、about_path、about_url。

Tag Helper

link_to

```
<!-- 基本用法 -->
<%= link_to('關於', '/about', target: :_blank) %>
<a href="/about" target="_blank">關於</a>

<!-- 利用 URL helper -->
<%= link_to('文章', posts_path) %>
<a href="/posts">關於</a>

<!-- block 用法 -->
<%= link_to('/about') do %>
  關於
<% end %>
```

image_tag、stylesheet_link_tag、javascript_include_tag

```
<%= image_tag("icon.png") %>
<img alt="Icon" src="/assets/icon.png" />

<%= stylesheet_link_tag "style" %>
<link href="/assets/style.css" media="screen" rel="stylesheet" />

<%= javascript_include_tag "xmlhr" %>
<script src="/assets/xmlhr.js"></script>
```

content_tag(name, content = nil, options = nil, &block)

```
<!-- 基本用法 -->
<%= content_tag(:p, "Hello world!") %>
<p>Hello world!</p>

<!-- 可巢狀使用 -->
<%= content_tag(:div, content_tag(:p, "Hello world!"), class: "strong") %>
<div class="strong"><p>Hello world!</p></div>

<!-- block 用法 -->
<%= content_tag :div, class: "strong" do %>
  Hello world!
<% end %>
<div class="strong">Hello world!</div>
```

tag(name, options = nil, open = false)

```
<!-- 基本用法 -->
<%= tag("br") %>
<br />

<!-- 添加屬性 -->
<%= tag("input", type: 'text', disabled: true) %>
<input type="text" disabled="disabled" />

<!-- 屬性字串會自動跳脫 -->
<%= tag("img", src: "open & shut.png") %>
<img src="open &amp;amp; shut.png" />

<!-- data attribute 使用技巧 -->
<%= tag("div", data: {name: 'Stephen', city_state: %w(Chicago IL)}) %>
<div data-name="Stephen"
     data-city-state="[&amp;quot;Chicago&amp;quot;,&amp;quot;IL&amp;quot;]" />
```

Form Helper

- form_tag 對於非 GET 的請求會自動產生 _method 與 authenticity_token 隱藏欄位。
- Rack 透過 _method 變數可覆寫原請求動詞。
- authenticity_token 用於抑止 CSRF 攻擊。

```
<%= form_tag('/posts/1', method: :put) do %>
  ...
<% end %>
<form action="/posts/1" method="post">
  ...
  <input name="authenticity_token" type="hidden" value="...">
  <input name="_method" type="hidden" value="put" />
  ...
</form>
```

ActiveRecord

- 繼承自 ActiveRecord::Base 的類別會得到 ORM 功能擴充。
- 類別對應的資料表是取其名稱取複數型並改為底線命名法，例如 Post 類別會對應到 posts 資料表，UserImage 類別會對應到 user_images 資料表。
- 資料表上的欄位，會成為類別的屬性。
- 類別對應的資料表若不依照慣例（CoC），可透過 table_name 設定。

```
class Article < ActiveRecord::Base
  self.table_name = 'posts'
end
```

CRUD

```
# 新增
post = Post.create title: 'Hello', content: 'World'

# 更新
post = Post.find(1)
post.update title: 'New Title', content: 'New Content'

# 刪除
post = Post.find(1)
post.destroy
```

Migration File

- 為將資料庫結構納入版本控制而生的策略
- 遷移檔位於 db/migrate/
- 遷移檔需要定義 up 與 down 方法，如果使用的遷移方法是可逆的，則定義 change 方法

```ruby
class CreatePosts < ActiveRecord::Migration
  def change
    create_table :posts do |t|
      # 預設有 id 欄位為主鍵
      t.string :title
      t.text :content
      t.date :publish_at
      t.timestamps # 新增 created_at 與 updated_at 欄位
    end
  end
end
```

Validation

驗證別	值	範例	說明
acceptance	Boolean		必須為真（常用於使用者同意條款）
confirmation	Boolean		若屬性名稱為 password，則驗證其值與 password_confirmation 一樣
exclusion	Hash	{ in: %w(admin superuser) }	黑名單
format	Hash	{ with: /foo|bar/i }	符合正規表達式，可用於驗證 Email
inclusion	Hash	{ in: 0..9 }	白名單
length	Hash	{ maximum: 30 }	限制內容長度
numericality	Boolean		必須是數字
presence	Boolean		必須填寫
uniqueness	Boolean		不能重複（例如避免不同使用者使用相同帳號）

指令整理

指令	說明
rails new PATH	新增 Rails 專案
rails server	啟動 HTTP 伺服器
rails generate controller NAME ACTION…	新增 controller 與其多個 action，同時產生相對應的 view
rails generate model NAME ATTR:TYPE…	新增 model
rails generate scaffold NAME ATTR:TYPE…	新增符合 RESTful 的資源頁面
rails generate migration NAME	新增遷移檔
rake db:migrate	資料庫遷移（執行剩下的遷移檔）
rake db:rollback	資料庫回溯（回到上個版本）
rake routes	查看 config/routes.rb 所生的網址路由表
rake assets:precompile	編譯 assets 到 public/assets