

Rails



Nội dung

- ✎ Giới thiệu về Rails
- ✎ Với project nào nên sử dụng Rails ?
- ✎ Kiến trúc của Rails
- ✎ Thành phần của một project
- ✎ Demo

Giới thiệu về Rails

- ☞ Là một framework MVC phát triển ứng dụng web được viết bằng ngôn ngữ Ruby.
- ☞ Được tạo ra bởi David Heinemeier Hansson.
- ☞ Thiết kế để việc lập trình ứng dụng web trở nên dễ hơn.
- ☞ Sử dụng một số mẫu thiết kế như convention over configuration , don't repeat yourself , active record pattern

2 nguyên lý chính

Don't repeat yourself

- ☞ Để duy trì code sạch và giảm sai sót.
- ☞ Rails áp dụng ý tưởng DRY.
- ☞ nghĩa là viết 1 lần và tái sử dụng nó.

Convention Over Configuration:

- ☞ Rails cho rằng cách tốt nhất để làm nhiều việc trong ứng dụng web.
- ☞ Và mặc định thiết lập nhiều quy ước.
- ☞ Những quy ước này sẽ giúp tăng tốc độ phát triển , giữ code ngắn và dễ hiểu
- ☞ Cho phép bạn dễ dàng điều chỉnh bên trong ứng dụng.

Ví dụ về DRY

Trước

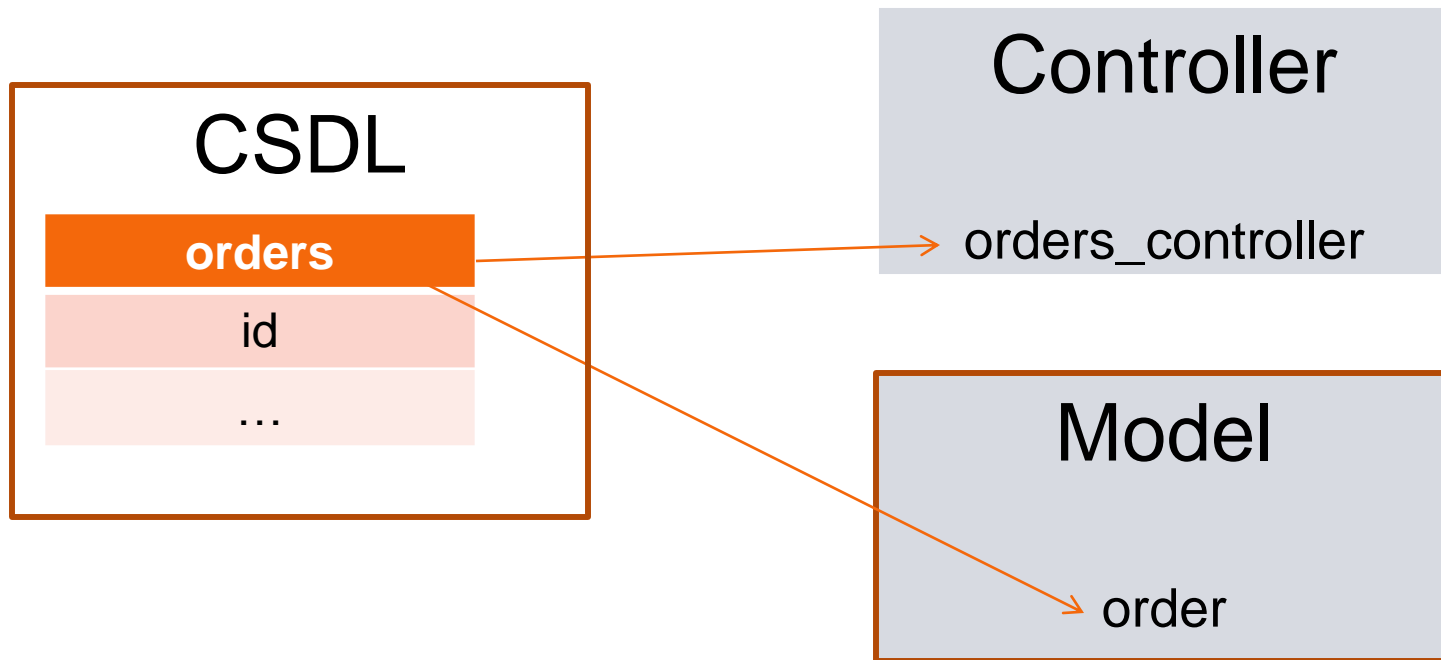
```
class Article < ActiveRecord::Base
  def self.all_published
    where("state = ?", "published")
  end
  def self.all_draft
    where("state = ?", "draft")
  end
  def self.all_spam
    where("state = ?", "spam")
  end
  def published?
    self.state == 'published'
  end
  def draft?
    self.state == 'draft'
  end
  def spam?
    self.state == 'spam'
  end
end
```

Sau

```
class Article < ActiveRecord::Base
  STATES = ['draft', 'published', 'spam']
  class << self
    STATES.each do |state_name|
      define_method "all_#{state_name}" do
        where("state = ?", state_name)
      end
    end
  end

  STATES.each do |state_name|
    define_method "#{state_name}?" do
      self.state == state_name
    end
  end
end
```

Ví dụ về Convention Over Configuration



Ưu và nhược điểm

Ưu điểm

- Nhiều collection lớn mã nguồn mở.
- Với các quy ước làm việc dành cho các nhà phát triển.
- Phát triển ứng dụng nhanh phù hợp với sự thay đổi yêu cầu.
- Ruby dễ học và code Ruby có thể tự làm tài liệu.
- Framework tốt về kiểm thử
- Rails và phần lớn thư viện đều là mã nguồn mở.

Nhược điểm

- Không phải tất cả các host đều hỗ trợ Rails.
- Không có nhiều nhà phát triển như Java và PHP.
- Ứng dụng không chạy nhanh như Java ...
- Nhiều quy ước nên mới học có vẻ rất phức tạp.

Với project nào nên sử dụng Rails ?

- ✎ Project cần code phần lõi và cơ sở dữ liệu mới.
- ✎ Cần tốc độ phát triển nhanh.
- ✎ Ứng dụng phức tạp.
- ✎ Khách hàng yêu cầu project mẫu.
- ✎ Quá trình phát triển đắt đỏ.
- ✎ Loại giá project là Fixed price project .

Kiến trúc của Rails

- ☞ Là framework sử dụng kiến trúc MVC.
- ☞ Gồm nhiều thành phần như Action Model, Action Record

Rails and MVC

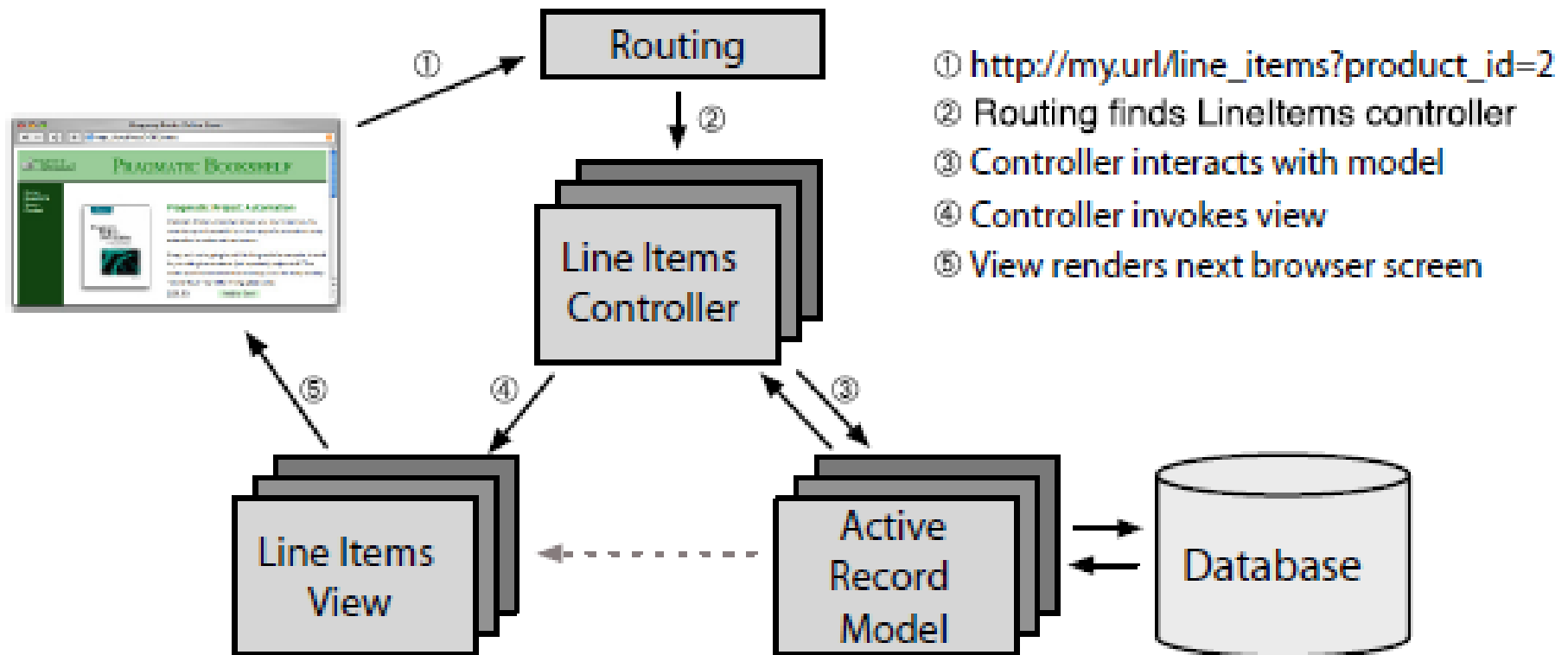
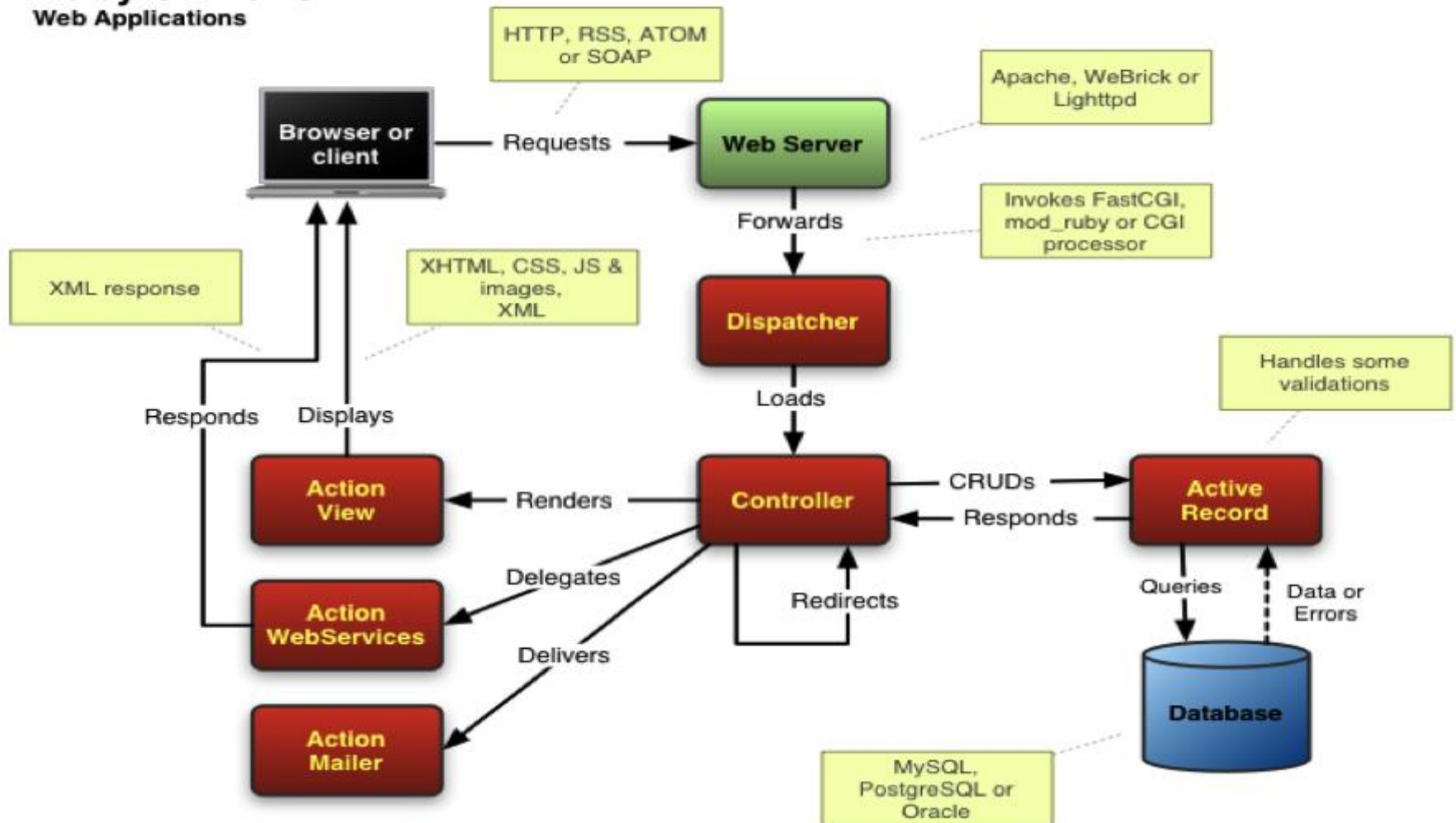


Figure 6—Rails and MVC

Ruby on Rails

Web Applications



Rails Model

- ✎ Giúp giảm bớt sự phức tạp khi giao tiếp với cơ sở dữ liệu.
- ✎ Là nền tảng của kiến trúc Rails

Action Model

- ☞ Định nghĩa giao tiếp giữa Action Pack và Action Record.

Action Record

- ✎ Sử dụng Action Record để kết hợp giữa ngôn ngữ Ruby với cơ sở dữ liệu quan hệ.
- ✎ Object-Relational Mapping(ORM) dùng để ánh xạ từ các bảng cơ sở dữ liệu vào các lớp.
- ✎ Các dòng ánh xạ tới các đối tượng, cột ánh xạ tới thuộc tính của đối tượng.

Action Pack

- ✎ Những module này chụp lấy yêu cầu từ trình duyệt và ánh xạ những yêu cầu này tới các action.
- ✎ Được định nghĩa trong tầng controller và sau khi action vẽ 1 view mà đã hiện thị trên trình duyệt.
- ✎ Chia ra là 3 modules nhỏ : Action Dispatch, Action Controller và Action view.

Action Dispatch

- ✎ Xử lý định tuyến các yêu cầu của trình duyệt web.
- ✎ Phân tích yêu cầu và xử lý cao quanh HTTP như xử lý cookies, sessions ...

Action Controller

- ☞ Cung cấp một base controller để tất cả các controller khác kế thừa.
- ☞ Bao gồm các action để điều khiển model và view.
- ☞ Làm dữ liệu có sẵn khi cần, điều khiển vẽ view và đổi hướng.
- ☞ Quản lý session, luồng ứng dụng, tính năng caching, module helper và thực hiện lọc

Thành phần của một project

app	Code lõi của ứng dụng, bao gồm models, views, controller và helpers. Tài nguyên tĩnh như CSS, JavaScript, ảnh.	test/	Sử dụng cho việc testing
bin/	Gồm các rails script để bắt đầu, setup, khởi khai, chạy ứng dụng	tmp/	File tạm thời như cache, session..
config/	Cấu hình về định tuyến, cơ sở dữ liệu	vendor/	Nơi cho tất cả code của bên thứ 3
db/	file về giản đồ cơ sở dữ liệu	README.rdoc	Hướng dẫn ngắn gọn cho ứng dụng của bạn
lib/	Mở rộng modules cho ứng dụng của bạn	Rakefile	Định vị và tải có thể chạy từ dòng lệnh. Định nghĩa nhiệm vụ được xác định trong thành phần của Rails.
log/	Ứng dụng các file log	Gemfile Gemfile.lock	Xác định những gem cần thiết cho ứng dụng.
public/	Gồm các file tĩnh và biên dịch tài nguyên	config.ru	cấu hình Rack cho rack máy chủ sử dụng để bắt đầu ứng dụng.

Tổng kết

- ☞ Nên sử dụng Rails khi nào ?
- ☞ 2 nguyên lý chính của Rails ?
- ☞ Kiến trúc của Rails.
- ☞ Các folder quan trọng .