Rails2's template

(respond_to method)

可以怎麼玩?

ihower@handlino.com

about me

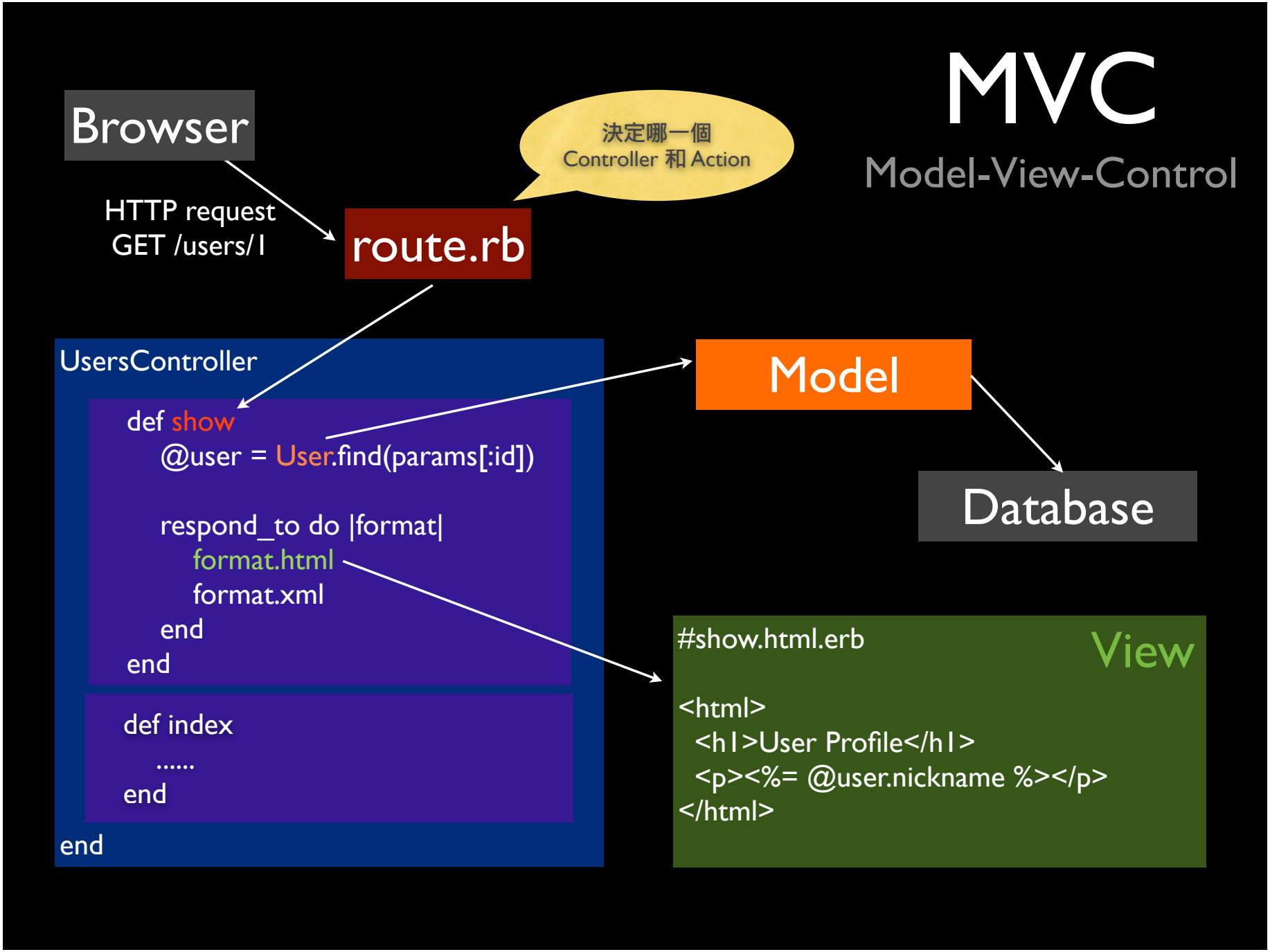
- 張文鈿 (a.k.a ihower)
  http://ihower.idv.tw/blog/
- 和多(Handlino) 鐵道員
- 主持 http://registrano.com 開發

agenda

- Ruby on Rails overview
- respond_to (what's format you want)
- template (how to generate format)

Rails Overview

- an open-source web framework for developing database-backed web applications
- Model-View-Control pattern
- a pure-Ruby development environment, from the Ajax in the view, to the request and response in the controller, to the domain model wrapping the database.


MVC
Model-View-Control
Browser
HTTP request
GET /users/1
route.rb
決定哪一個
Controller 和 Action
UsersController
def show
@user = User.find(params[:id])
respond_to do |format|
format.html
format.xml
end
end
def index
.....
end
end
Model
Database
#show.html.erb
View
<html>
<h1>User Profile</h1>
<%= @user.nickname %>
</html>

respond_to

你要什麼格式?

One Action, Multiple Response Formats

```
def index
  @users = User.find(:all)
    respond_to do |format|
      format.html # index.html.erb
      format.xml { render :xml => @user.to_xml }
    end
end
```

format.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
    <p> ihower at 2008-01-19 </p>
</body>
</html>
```

format.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <created-at type="datetime">2008-01-19T09:55:32+08:00</created-at>
  <id type="integer">2</id>
  <name>ihower</name>
  <updated-at type="datetime">2008-01-19T09:55:32+08:00</updated-at>
</user>
```

you don't need this!

```
def show_html
  @users = User.find(:all)
end

def show_xml
  @users = User.find(:all)
  render :xml => @users.to_xml
end

def show_json
  @user = User.find(:all)
    render :json => @user.to_json
end
```

只需定義一個 action
減少重複的程式碼

Don’t repeat yourself

更多 formats

- format.html
- format.xml
- format.js
- format.json
- format.atom
- format.rss
- format.csv
- format.xls
- format.yaml
- format.txt
- more....


http://registrano.com/events/5381ae/attendees
iGoogle Twitter Blog PlanetRoR Yahoo! Yahoo!字典 ST Trac myCATCH Registrano localhost
iGoogle
Registrano | Attendees
Registrano
你的活動平台
嗨，ihower (設定). 登出？
首頁 舉辦的活動 參與的活動 我的群組 管理後台
舉辦的活動 > [Sample] 愛跨年 > 瀏覽參與者
[Sample] 愛跨年 Registrano Sample
活動主控台 瀏覽參與者 瀏覽取消報名的參與者 活動報表 整合方案
建立新的參與者 E-mail 給參與者
整批動作： Mail Batch
每頁 20 筆資料，根據 報名時間 排序 送出
all / none 報名編號 姓名 E-Mail 手機 報名時間 狀態 Action
8 test@test.com 2008/03/17 4:19 PM Confirmed 詳細顯示 編輯 移除
6 duckl@gmail.com 2008/02/24 4:44 AM Confirmed 詳細顯示 編輯 移除
3 詳細顯示 編輯 移除
2
1 idv@gmail.com 2007/12/19 3:47 PM Pending 詳細顯示 編輯 移除
http://registrano.com/events/5381ae/attendees.csv
http://registrano.com/events/5381ae/attendees.xls
下載其他格式:
CSV | Excel
關於我們 服務條款 隱私權政策 定價 討論區
Choose your language: 繁體中文 English

輕鬆在原有的程式架構上
新增不同格式的支援
(i.e. 不同 UI 介面)

Rails: how to know?

根據 URL

http://localhost:3000/users.xml

在 template 中可以這樣寫

```
<%= link_to ‘User List’, formatted_users_path(:xml) %>
```


```
<a href=”/users.xml”>User List</a>
```

根據 HTTP request Headers

```
GET /users HTTP/1.1

Host: localhost:3000
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X; zh-TW; rv:1.8.1.13)
            Gecko/20080311 Firefox/2.0.0.13
Accept: text/javascript, text/html, application/xml, text/xml, */*
Accept-Language: zh-tw,en-us;q=0.7,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: Big5,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
X-Requested-With: XMLHttpRequest
X-Prototype-Version: 1.6.0.1
```

通常透過 Javascript 發送 Ajax request 時，加以設定。

- 根據 params[:format] 參數
- 例如 GET /users/1?format=xml

- 直接在 Controller code 中設定
- 例如

```
def show
  request.format = :xml
  ......
end
```

這些 format 可以
怎麼跟 Ajax 搭配

複習一下 Ajax 簡易用法

```
<a onclick="$.ajax({async:true, beforeSend:function(xhr)
{xhr.setRequestHeader('Accept', 'text/html, */*')}, complete:function(request){
$("#content").html(request.responseText);}, dataType:'html', type:'get', url:'/terms'});
return false;" href="/terms">服務條款</a>

<div id="content">
</div>
```

把 #content 的內容換成傳回來的 HTML 內容

Browser

Ajax 請求

回應 format.html

```
<h1>ABC</j1>
<ul>
   <li>1</li>
   <li>2</li>
</ul>
```

Server

古典 Ajax 用法

```
$.ajax({ async:true,
beforeSend:function(xhr) {xhr.setRequestHeader('Accept', 'text/xml, */*')}, type: “get”,
url: “/users”, dataType: “xml”, success: function(xml){
 // js blah code
 // js blah code
 // js blah code
});
```

操作傳回來的 XML data，例如 DOM 操作

Browser

Ajax 請求

回應 format.xml

```
<data title=”title”>
   <item>1</item>
   <item>2</item>
   <item>3</item>
</data>
```

Server

新潮 Ajax 用法

```
$.ajax({ async:true,
beforeSend:function(xhr) {xhr.setRequestHeader('Accept', 'text/json, */*')}, type: “get”,
url: “/users”, dataType: “json”, success: function(json){
 // js blah code
 // js blah code
 // js blah code
});
```

操作傳回來的 json data，例如 DOM 操作

Browser

Ajax 請求

回應 format.json

[{"name": "aaaa", "updated_at": "2008/01/19 09:55:32 +0800", "id": 2, "created_at": "2008/01/19 09:55:32 +0800"}, {"name": "bbbb222", "updated_at": "2008/01/19 09:56:11 +0800", "id": 3, "created_at": "2008/01/19 09:55:40 +0800"}]

Server

值得一提的
format.js

注入腳本到瀏覽器執行

注入腳本到瀏覽器執行的 Ajax 用法

```
<a onclick="$.ajax({async:true, beforeSend:function(xhr)
{xhr.setRequestHeader('Accept', 'text/javascript, text/html, application/xml, text/xml, */
*')}, dataType:'script', type:'get', url:'/user/1'}); return false;>User</a>

<div id=”content”>
</div>
```

執行傳回來的 Javascript 腳本

Browser

Ajax 請求

回應 format.js

```
$("#content").html(‘ blah’);
$(“#sidebar”).html(‘ blah’);
$("#content").effect("highlight");
```

Server

respond_to 的另個好處:
支援 Graceful Degradation

```
def index
  @users = User.find(:all)
    respond_to do |format|
      format.js {
        render :update do |page|
          page.replace_html 'content', '<p>blah</p>'
        end
      }
      format.html #index.html.erb
    end
end
```

Browser支援 Javascript

Browser 不支援 Javascript

```
<a href="/users" onclick="$.ajax(...blah...);return false;">
```

自訂格式
custom format

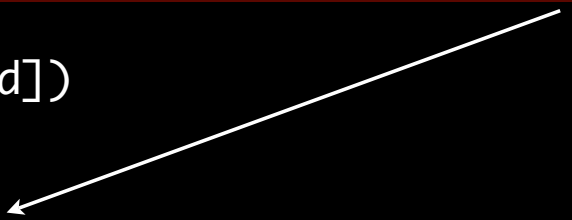
```
# config/initializers/mime_types.rb

Mime::Type.register ‘audio/mpeg’, :mp3
Mime::Type.register ‘audio/mpegurl’, :m3u
```

```
# your controller

def show
  @mp3 = Mp3.find(params[:id])
  respond_to do |format|
    format.html
    format.mp3 { redirect_to @mp3.url }
    format.m3u { render :text => @mp3.url }
  end
end
```

http://localhost:3000/mp3s/1.mp3



template

如何產生這些格式?

template

- format (minetype) 與 template generator (renderer) 是兩回事
- Rails2 的慣例是 action.minetype.renderer
  例如 filename.html.erb
- it's awesome!

template 的慣例命名

```
def index
  @users = User.find(:all)
    respond_to do |format|
      format.html # index.html.erb
      format.xml # index.xml.builder
  end
end
```

erb

- 內嵌 ruby code
- 最常用來生成 HTML（即format.html)

```
<h1><%= @group.name %></h1>
```



```
<h1>HappyDesigner</h1>
```

js.erb

- 拿成寫 Javascript 也可以
  (format.js 或 format.json)
- Write JavaScript directly
- Rails 1.x 需要 hack! (google MinusMOR)

```
$j("#foo").html(<%= (render :partial => 'bar.html').to_json %>);
$j("#foo").Highlight();
```

css.erb

- 生成 CSS (format.css)

```
body {
  background-color: <%= @bg_color %>
}
```



```
body {
  background-color: #666666
}
```

builder

- 通常用來生成 XML

```
xml.instruct!
xml.title "This is a title"
xml.person do
  xml.first_name "Ryan"
  xml.last_name "Raaum"
end
```

```
<?xml version="1.0" encoding="UTF-8"?>
<title>This is a title</title>
<person>
  <first_name>Ryan</first_name>
  <last_name>Raaum</last_name>
</person>
```

atom.builder

- 生成 Atom feed，Rails2 提供 Atom helper

```
atom_feed do |feed|
  feed.title( @feed_title )
  feed.updated((@events.first.created_at))
  for event in @events
    feed.entry(event) do |entry|
      entry.title(event.title)
      entry.content(event.description, :type => 'html')
      entry.author do |author|
        author.name( event.creator.nickname )
      end
    end
  end
end
```

Attentions for Rails 1.x developer

- filename.rhtml 變成 filename.html.erb
- filename.rxml 變成 filename.xml.builder
- 雖然變囉唆，但彈性更棒!!

RJS

- 用 Ruby 來寫 Javascript (format.js)

```
page.insert_html :bottom, 'notes', :partial =>'note'
page.visual_effect :highlight, 'note'
```

```
def show
@note = Note.find(params[:id])
 respond_to |format|
   format.js # show.js.rjs
end
  end
```

```
try {
    new Insertion.Bottom("notes", "blah");
    new Effect.Highlight("note",{});
} catch (e) { alert('RJS error:\n\n' + e.toString()); alert('new
Insertion.Bottom(\"notes\", \"blah\");\nnew Effect.Highlight(\"note\",{});');
throw e }
```

以上是內建的
template generator

更多在 ruby gem/rails plugins

markby

- 用 Ruby 寫 HTML

```
h2 { h @entity.content }
p { h @entity.category.name }
p { h @entity.created_at }
ul {
  @entity.children.each do |c|
    li {
      cn = c.category ? c.category.name : ""
        text "%s: %s" % [cn, c.content]
    }
  end
}
```

Yes, it's Ruby code!

haml (marku haiku)

```
#content
    .left.column
      %h2 Welcome to our site!
      %p= print_information
    .right.column= render :partial => "sidebar"
```

產生 XHTML

```
<div id='content'>
  <div class='left column'>
    <h2>Welcome to our site!</h2>
    <p>
      <%= print_information %>
    </p>
  </div>
  <div class="right column">
    <%= render :partial => "sidebar" %>
  </div>
</div>
```

Sass

(Syntactically Awesome StyleSheets)

```
// Constant Definitions

!sidebar_width = 200px
!page_width    = 800px
!primary_color = #eeeeee

body
  :font
    :family fantasy
    :size 30em
    :weight bold

// Scoped Styles

#contents
  :width= !page_width
  #sidebar
    :float  right
    :width= !sidebar_width
  #main
    :width= !page_width - !sidebar_width
    :background= !primary_color
    h2
      :color blue
#footer
  :height 200px
```


```
body {
  font-family: fantasy;
  font-size: 30em;
  font-weight: bold; }

#contents {
  width: 800px; }

#contents #sidebar {
    float: right;
    width: 200px; }

#contents #main {
    width: 600px;
    background: #eeeeee; }

#contents #main h2 {
      color: blue; }

#footer {
  height: 200px; }
```

族繁不及備載

Textile, Markdown, liquid, erubis...etc

Bonus

iPhone UI

到底什麼時候會出 3G iPhone?

```
# config/initializers/mime_types.rb
Mime::Type.register_alias "text/html" , :iphone
```

```
class ApplicationController < ActionController::Base
  before_filter :adjust_format_for_iphone
  helper_method :iphone_user_agent?

protected

  def adjust_format_for_iphone
    request.format = :iphone if iphone_user_agent? || iphone_subdomain?
  end
    # Request from an iPhone or iPod touch?
    # (Mobile Safari user agent)
  def iphone_user_agent?
    request.env["HTTP_USER_AGENT" ] &&
      request.env["HTTP_USER_AGENT" ][/(MobileV.+Safari)/]
  end

  def iphone_subdomain?
    return request.subdomains.first == "iphone"
  end
end
```

```
class ListsController < ApplicationController
  def index
    @lists = List.find(:all)

    respond_to do |format|
      format.html # index.html.erb
      format.iphone # index.iphone.erb
      format.xml { render :xml => @lists }
    end
end

def show
  @list = List.find(params[:id])
  respond_to do |format|
    format.html # show.html.erb
    format.iphone { render :layout => false } # show.iphone.erb
    format.xml { render :xml => @list }
  end
end
# other CRUD actions

end
```

iui
User Interface (UI) Library for Safari development on iPhone

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta id="viewport" name="viewport"
content="width=320; initial-scale=1.0; maximum-scale=1.0;
user-scalable=0;" />
<title>TODO - iPhone version</title>
<%= stylesheet_link_tag 'iui' %>
<%= javascript_include_tag 'iui', 'prototype', 'application' %>
</head>
<body>
<div class="toolbar">
<h1 id="pageTitle"></h1>
<a id="backButton" class="button" href="#"></a>
</div>
<%= yield %>
</body>
</html>
```

http://code.google.com/p/iui/

http://registrano.com/events.iphone

iPhoney: A iPhone web simulator
http://www.marketcircle.com/iphoney/

Flex on Rails

Adobe Photoshop Express 好厲害!

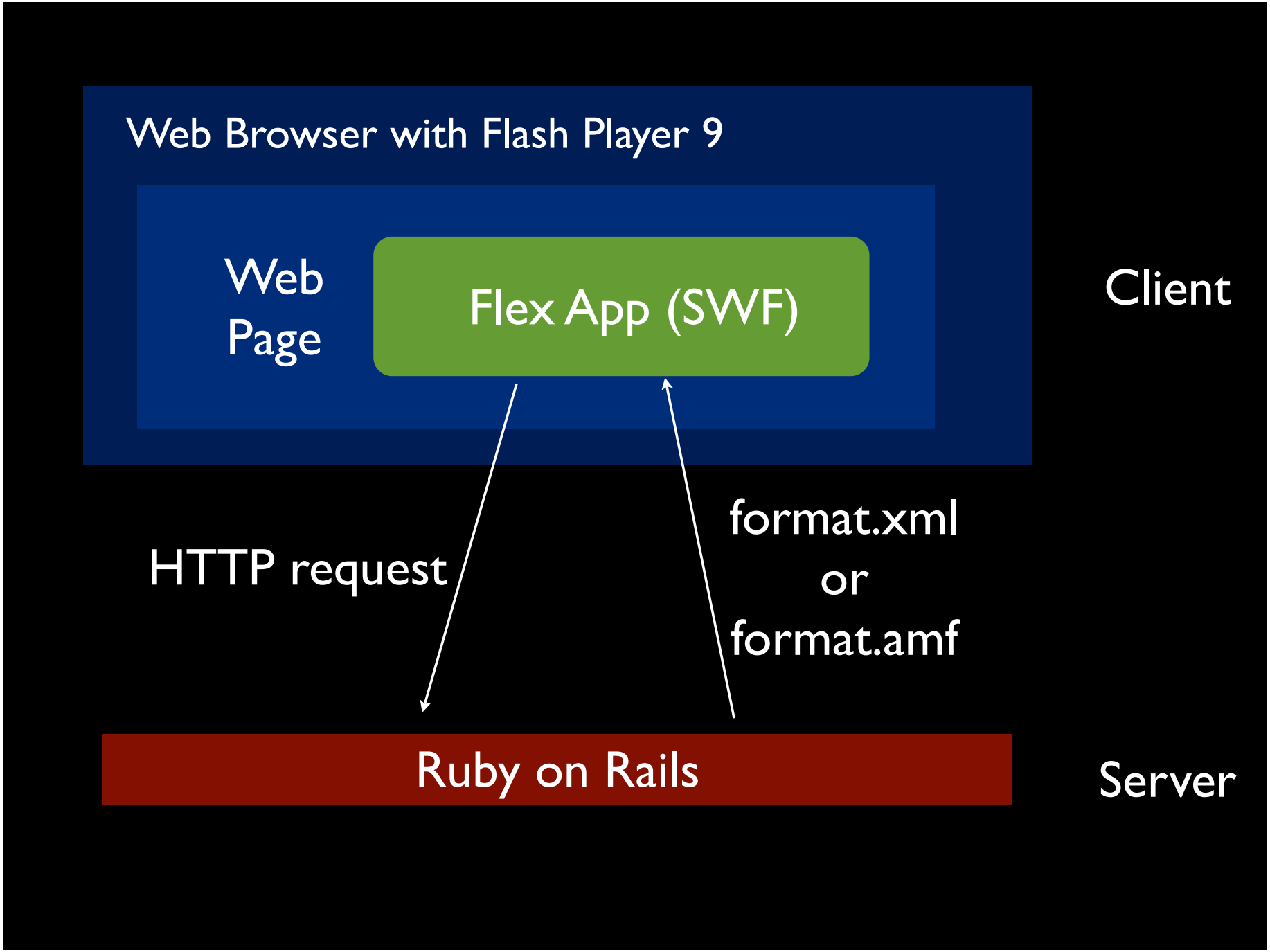
Flex 3

- Adobe's Open Source framework

MXML(XML files with .mxml)
ActionScript (text files with .as)

有免錢 Flex SDK
或 Flex Builder

a SWF file
runs in Flash player


Web Browser with Flash Player 9
Web Page
Flex App (SWF)
Client
HTTP request
format.xml or format.amf
Ruby on Rails
Server

- see you -

OSDC 2008-4-13
Practical Rails2