

Practical Rails2

ihower@handlino.com

about me

- 張文鈞 (a.k.a ihower)
<http://ihower.idv.tw/blog/>

The top screenshot displays a presentation slide from the OSDC.tw conference. The title is "演講: Practical Rails2 (OSDC.tw)". The content discusses RESTful development in Rails 2.0, mentioning the shift from Representational State Transfer (REST) to RESTful principles. It highlights the use of RESTful controllers and actions, and the integration of various technologies like Ajax, Metaprogramming, and useful plugins. The bottom screenshot shows the homepage of Handlino.com, featuring a navigation bar with links to Home, Blog, Service, Product, Portfolio, About, and Contact. The main content area introduces Handlino as a web design & development company based in HsinChu, Taiwan. It includes sections for services, portfolio, products, and a contact form.

Registrano | Lobby

http://registrano.com/#start

Google

iGoogle Twitter Blog Flickr GitHub PlanetRoR Yahoo! Google Translate Yahoo!字典 Trac ST myCATCH Registrano localhost Git - SVN Crash Course

Registrano Registrano History.txt at master fr... Timeline - Rails Trac - ...

Registrano

你的活動平台

嗨，Ihower (設定). 登出？

首頁 據辦的活動 參與的活動 我的群組 管理後台

Registrano 是你的線上活動平台

研討會、講座、營隊、團購、大食團、徵人、連署、出遊、進香團或各種社群活動.....你每次要舉辦活動的時候，都在煩惱怎麼製作活動網站、處理報名事宜嗎？在 Registrano 建立活動是如此簡單，一切都是可以在彈指之間輕鬆完成。（請參考“關於我們”頁面的功能列表）

這項服務是免費的嗎？

所有基本功能都是免費的，我們也不會在你的活動頁面放置任何廣告連結。如果你要舉辦超過 30 人的活動，請考慮我們的 premium plan。不過，在 2008/03/31 前，Registrano 辦任何活動都是免費的囉。

可以給我一些參考範例嗎？

當然，我們提供了一些範本。

想要更深入瞭解 Registrano 嗎？請見，請見 關於我們。

近期公開活動

- 人本森林育活動員培訓 on 2008-03-30
- 《網頁親和力》助寫贊助暨初回限定版預購 on 2008-03-31
- OSDC.TW 2008 (Open Source Developers' Conference) on 2008-04-12
- UbiWe 4/13 on 2008-04-13
- 第七屆 CocoaHeads Taipei Meeting on 2008-04-17
- 2008四月份 SA@Tainan 結構化佈線 on 2008-04-19
- 閱讀沙龍第四講--龍應台讀《秧歌》 on 2008-04-21
- 【清華思沙龍】戰火下的墮天使——童兵戰地烏干達 on 2008-06-09

Registrano your event hosting

Hi, Ibir [setting] Logout

Home Hosting Events Attending Events My Groups

Rails2 Tutorial

View Event

Dashboard View Attendees Overview

Time: 2007/12/25 2:00 AM - 12/25 5:00 AM
URL: http://beta.registrano.com/events/rails2-tutorial-2007
Status: Active
Attendees: 5 / 30 (25 MB)

Step by Step Edit Event Info Edit Event Form Options Edit Google Map Edit attendee status Edit Advanced Options

Back to Testing Mode

看瀏覽範例 建立你的活動

新群組

- Even Design
- 華娟網友會
- KaLUG 大高雄 Linux 使用者協會
- UbiWe
- Jollen's Consulting 教育訓練
- Tsing Hua Thinkers Salon
- Route 2 - 自駕旅行
- OSDC
- myCATCH
- 快樂設計師_HappyDesigner



Search:

search is working now...

Job Board[BizUnite is looking
for a UI](#)[Design/Developer.](#)See more on the [Job
Board](#).**Worth attention**[Workshops](#)[Podcast](#)**Categories**[Documentation](#)[General](#)[Horizon](#)[Jobs](#)[Launches](#)[Praise](#)[Releases](#)[Sightings](#)[Tricks](#)[Tools](#)[Syndicate](#)**Rails 2.0: It's done!**

Posted by David December 07, 2007 @ 03:03 PM

2007/12/7

Rails 2.0 is finally finished after about a year in the making. This is a fantastic release that's absolutely stuffed with great new features, loads of fixes, and an incredible amount of polish. We've even taken a fair bit of cruft out to make the whole package more coherent and lean.

What a milestone for Ruby on Rails as well. I've personally been working on this framework for about four and a half years and we have contributors who've been around for almost as long as well. It's really satisfying to see how far we've come in that period of time. That we've proven the initial hype worthy, that we've been able to stick with it and continue to push the envelope.

Before jumping into the breakdown of features, I'd just like to extend my deep gratitude towards everyone who helped make this release possible. From the stable of merry men in the Rails core to the hundreds of contributors who got a patch applied to everyone who participated in the community over the year. This release is a triumph for large-scale open source development and you can all be mighty proud of the role you played. Cheers!

With the touchy-feely stuff out of the way, let's dig into the feast and look at just a sliver of what's new:

Action Pack: Resources

This is where the bulk of the action for 2.0 has gone. We've got a slew of improvements to the RESTful lifestyle. First, we've dropped the semicolon for custom methods instead of the regular slash. So `/people/1;edit` is now `/people/1/edit`. We've also added the namespace feature to routing resources that makes it really easy to confine things like admin interfaces:

agenda

- RESTful Rails
- Bonus

not include these:

- Testing
- Database & ActiveRecord
- Security
- Cache & Scale
- i18n and L10n

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

View

```
#show.html.erb
<html>
  <h1>User Profile</h1>
  <p><%= @user.nickname %></p>
</html>
```

進入主題

RESTful Rails

REST

Representational State Transfer

表象化狀態轉變

a lot of theory and
abstract stuff

RESTful – Google 搜尋

http://www.google.com/search?hl=zh-TW&q=RESTful+&btnG=搜尋&lr=lang_zh-TW

iGoogle Twitter Blog Flickr GitHub PlanetRoR Yahoo! Google Translate Yahoo!字典 Trac ST myCATCH Registrano localhost RESTful - Go... {ihower.idv.t... Yahoo!奇摩字典 Twitter Rest 101: Par... 4 - Routing 5 - Respond! O'Reilly ihower@gmail.com

所有網頁 圖片 新聞 網上論壇 書籍 Gmail 更多 ▾

所有網頁 個人化 關於RESTful 大約有11,300 頁繁體中文搜尋結果，這是第 1 頁

Google RESTful 搜尋 | 進階搜尋 | 使用偏好

搜尋所有網站 搜尋所有中文網頁 搜尋繁體中文網頁

[{ihower.idv.tw} blog » Blog Archive » 什麼是REST跟RESTful?](#) - 上午 1:35
REST 是一種分散式超媒體系統(如WWW)的軟體架構風格，你可以想像它是一個良好設計的
Web 應用程式規則：一組網路Web頁面(虛擬的狀態機器)，其中Client 透過點選超 ...
[ihower.idv.tw/blog/archives/1542 - 25k - 頁庫存檔 - 類似網頁](#)

[REST - {ihower.idv.tw} blog](#)
在上一篇Rails RESTful 相關工具 中有提到，ActiveResource 是一個針對Rails RESTful APIs 所用的client-side XML consumer，我們花了好幾篇學習為何RESTful 跟如何 ...
[ihower.idv.tw/blog/archives/category/rest - 21k - 頁庫存檔 - 類似網頁](#)
[ihower.idv.tw 的其它相關資訊 »](#)

[ZDNet Taiwan - 名家專欄- 蕭百齡- REST 的口水正在洩洪](#)
蕭百齡 2007/09/07 11:17 最近幾個月，網路上 REST vs. SOAP 的口角甚囂塵上，實在看不下去了。就連在SOA 和Web services 界的重量級人物— Burton Group 的分析總 ...
[www.zdnet.com.tw/enterprise/ column/tiger/0,2000085681,20123321,00.htm - 35k - 頁庫存檔 - 類似網頁](#)

[ZDNet Taiwan - 名家專欄- 蕭百齡- REST 的口水正在洩洪](#)
ZDNet Taiwan, <http://www.zdnet.com.tw>。結合了美國科技資訊類的第一大網站ZDNet豐富的資訊與亞洲及台灣的即時新聞，ZDNet 台灣中文網站包括國際新聞、亞洲新聞及台灣 ...
[www.zdnet.com.tw/enterprise/ column/tiger/0,2000085681,20123321-1,00.htm - 34k - 頁庫存檔 - 類似網頁](#)

[勞虎跑得快» Blog Archive » 擔心未來的REST 怪物正在形成](#)

wait!
we are not REST expert or devotee.

到底 Rails RESTful
可以解決什麼問題？

Before RESTful we have a big problem:
**designing controller
and action is chaos**

a simple controller

```
class EventsController < ApplicationController  
  
  index  
  show  
  new  
  create  
  edit  
  update  
  destroy
```

a bloated controller

```
class EventsController < ApplicationController

  index
  show
  new
  create
  edit
  update
  destroy
  feeds
  add_comment
  show_comment
  destroy_comment
  edit_comment
  approve_comment
  mark_comment_as_spam

  watch_list
  add_favorite
  invite
  join
  leave
  white_member_list
  black_member_list
  deny_user
  allow_user
  edit_managers
  set_user_as_manager
  set_user_as_member
  ....
  etc.
```

a bloated controller

```
class EventsController < ApplicationController
```

```
index
```

```
watch_list
```

```
show
```

```
add_favorite
```

```
new
```

```
invite
```

```
create
```

```
join
```

```
update
```

```
leave
```

```
white_member_list
```

```
black_member_list
```

```
ban_member
```

什麼時候該新寫 controller 呢???
這些那些個 actions 又該怎麼放????
又該怎麼命名這些 controller 跟 actions 呢???

```
feeds
```

```
deny_user
```

```
add_comment
```

```
allow_user
```

```
show_comment
```

```
edit_managers
```

```
destroy_comment
```

```
set_user_as_manager
```

```
edit_comment
```

```
set_user_as_member
```

```
approve_comment
```

```
.....
```

```
mark_comment_as_spam
```

```
etc.
```

named routes

```
# routes.rb
map.connect '/:controller/:action/:id'
```

```
<%= link_to 'text', :controller => 'events',
            :action => 'show',
            :id => event.id %>
```

為求簡化，設定 named route

```
# routes.rb
map.event '/events/:id', :controller => 'event', :action => 'show'
```

```
<%= link_to 'text', event_path(event) %>
```

event.to_param == event.id

named routes

```
# routes.rb  
map.connect '/:controller/:action/:id'
```

<%= link_to 'text', 'events' %>
hmm.... 越來越多的 named routes

event_delete_path?

event_create_path?

events_path?

```
# routes.rb  
map.event '/events/:id', :controller => 'event', :action => 'show'
```

```
<%= link_to 'text', event_path(event) %>
```

event.to_param == event.id

we need a paradigm help us :
decide how to name and organize
the controllers and actions!!

The ideas from
CRUD...

HTTP methods

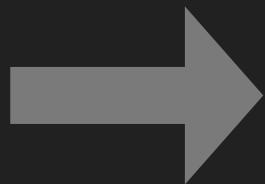
(RFC 2616)

POST	GET	PUT	DELETE
Create	Read	Update	Delete

GET is defined as
a safe method

/events/create
/events/show/ \mid
/events/update/ \mid
/events/destroy/ \mid

Add
HTTP method



POST /events
GET /events/ \mid
PUT /events/ \mid
DELETE /events/ \mid

Remove actions from URL, and
we have simple named route.

CRUD-based action names get things simpler

create	show	update	delete
POST	GET	PUT	DELETE

盡量讓每個 controller 只負責一組 CRUD

routes.rb

```
ActionController::Routing::Routes.draw do |map|  
  map.resources :events  
end
```

a resource is
something with URL

自動建立一組 named routes 對應到 actions

4

7

4 HTTP method

index

The default
request method is
GET

```
<%= link_to 'event list', events_path %>
```

```
class EventsController < ApplicationController

  def index
    @events = Event.find(:all)
  end

  ...
end
```

show

The default
request method is
GET

```
<%= link_to event.name, event_path(event) %>
```

```
class EventsController < ApplicationController

  def show
    @event = Event.find(params[:id])
  end

  ...
end
```

new/create

```
<%= link_to 'new event', new_event_path %>
```

```
class EventsController < ApplicationController
  def new
    @event = Event.new
  end
end
```

In a form, the default request method is POST

```
<% form_for @event, :url => events_path do |f| %>
  <%= f.text_field :name %>
  <%= f.submit "Create" %>
<% end %>
```

```
class EventsController < ApplicationController
  def create
    Event.create(params[:id])
  end
end
```

edit/update

```
<%= link_to event.name, edit_event_path(event) %>
```

```
class EventsController < ApplicationController
  def edit
    @event = Event.find(params[:id])
  end
end

<% form_for @event, :url => event_path(@event),
   :method => :put do |f| %>
  <%= f.text_field :name %>
  <%= f.submit "Create" %>
<% end %>
```

the request
method is
PUT

```
class EventsController < ApplicationController
  def create
    Event.create(params[:id])
  end
end
```

destroy

the request
method is
DELETE

```
<%= link_to @event, event_path(@event), :method => :delete %>
```

```
class EventsController < ApplicationController
```

```
  def destroy
    Event.find(params[:id]).destroy
  end
```

```
  ...
```

```
end
```

4 HTTP methods, 4 URL helper, 7 actions

Helper	GET	POST	PUT	DELETE
<code>event_path(@event)</code>	<code>/events/ show</code>		<code>/events/ update</code>	<code>/events/ destroy</code>
<code>events_path</code>	<code>/events index</code>	<code>/events create</code>		
<code>edit_event_path(@event)</code>	<code>/events/ edit</code>			
<code>new_events_path</code>	<code>/events/new new</code>			

Singular and Plural RESTful Routes

- show, new, edit, destroy 是單數，對特定元素操作
- index, new, create 是複數，對群集操作

event_path(@event)

需要參數，根據 HTTP verb 決定 show, update, destroy

events_path

毋需參數，根據 HTTP verb 決定 index, create

new, edit

單數?複
數?

除了_path 結尾，
_url 結尾則加上 http://domain/

[custom route]_event[s]_path(event)

:method => GET | POST | PUT | DELETE

~~map.connect ':controller/:action/:id'~~

~~link_to event.name, :controller => 'events',
:action => :show , :id => event.id~~

link_to event.name, event_path(event)

只需記得 resources 就可以寫出 URL Helper

PUT? DELETE?

The PUT&DELETE Cheat

- 瀏覽器完全不支援 PUT&DELETE method
- Rails 偷藏 _method 參數

```
<form id="edit_events_1" method="post" action="/events/1">
  <input type="hidden" value="put" name="_method"/>
  ....
</form>
```

```
<a onclick="var f = document.createElement('form'); f.style.display = 'none';
this.parentNode.appendChild(f); f.method = 'POST'; f.action = this.href;var m =
document.createElement('input'); m.setAttribute('type', 'hidden');
m.setAttribute('name', '_method'); m.setAttribute('value', 'delete');
f.appendChild(m);f.submit();return false;" href="/events/1">Destroy</a>
```

The Problem

- HTML spec only define GET/POST,
so HTML forms don't support PUT/DELETE
- XMLHttpRequest spec (i.e. Ajax request) define
at least support GET/POST/PUT/DELETE/HEAD/OPTIONS
- Firefox/Safari support
- Opera have no respond body if PUT/DELETE
- IE support except DELETE !!!!

prototype
JavaScript framework

Download API Docs Tips and Tutorials Blog Discuss Contribute

Learn more about Prototype »

● Ajax

- Ajax Options
- Ajax.PeriodicalUpdater
- Ajax.Request
- Ajax.Responders
- Ajax.Response
- Ajax.Updater

API Docs

- Utility Methods
- Ajax
- Array
- Class
- Date
- Element
- Element.Methods
- Element.Methods.Simulated
- Enumerable
- Event
- Form
- Form.Element

Ajax Options

This details all core options (shared by all AJAX requesters) and callbacks.

All requester object in the `Ajax` namespace share a common set of **options** and **callbacks**. Callbacks are called at various points in the life-cycle of a request, and always feature the same list of arguments. They are passed to requesters right along with their other options.

As a Ruby On Rails special, Prototype also reacts to other verbs (such as 'put' and 'delete' by actually using 'post' and putting an extra '`_method`' parameter with the originally requested method in there.)

encoding issues arise, you may have to tweak it in accordance with other encoding-related parts of your page code and server side.

method	'post'	The HTTP method to use for the request. The other widespread possibility is 'get'. As a Ruby On Rails special, Prototype also reacts to other verbs (such as 'put' and 'delete' by actually using 'post' and putting an extra ' <code>_method</code> ' parameter with the originally requested method in there.)
parameters	''	The parameters for the request, which will be encoded into the URL for a 'get' method or into the request body for the other methods. This can be provided

1 match delete Done

jQuery Download Documentation Plugins Tutorials Discuss Blog Log in / create account article discussion edit history

Ajax/jQuery.ajax

« Back to Ajax

jQuery.ajax(options)

[edit]

Overview Options Examples

Options:

async Boolean Default: true

By default, all requests are sent asynchronous (e.g. this is set to true by default). If you need synchronous requests, set this option to false. Note that synchronous requests may temporarily lock the browser, disabling any actions while

The type of request to make ("POST" or "GET"), default is "GET". Note: Other HTTP request methods, such as PUT and DELETE, can also be used here, but they are **not supported by all browsers**.

cache Boolean Default: true

Added in jQuery 1.2, if set to false it will force the pages that you request to not be cached by the browser.

complete Function

A function to be called when the request finishes (after success and error callbacks are executed). The function gets passed two arguments: The XMLHttpRequest object and a string describing the type of success of the request. This is an [Ajax Event](#).

```
function (XMLHttpRequest, textStatus) {  
    this; // the options for this ajax request  
}
```

contentType String Default: "application/x-www-form-urlencoded"

When sending data to the server, use this contentType. Default is "application/x-www-form-urlencoded", which is fine.

4 HTTP methods, 4 URL helper, 7 actions

Helper	GET	POST	PUT	DELETE
event_path(@event)	/events/ show		/events/ update	/events/ destroy
events_path	/events index	/events create		
edit_event_path(@event)	/events/ edit			
new_events_path	/events/new new			

It's beauty, but RESTful can handle the real & complex world ?

我想要加自己的 action 之問題一：
event has many attendees

Model design

```
class Event < ActiveRecord::Base
  has_many :attendees
end

class Attendee < ActiveRecord::Base
  belongs_to :event
end
```

nested resources(I)

預設的 controller
會是 attendees

```
map.resources :events do |event|
  event.resources :attendees, :controller => 'event_attendees'
end
```

/events/2/attendees

```
<%= link_to 'event attendees', event_attendees_path(@event) %>
```

```
class EventAttendeesController < ApplicationController
```

```
  def index
```

```
    @attendees = Event.find(params[:event_id]).attendees
```

```
  end
```

```
  ...
```

```
  end
```

nested resources(2)

/events/2/attendees/3

```
<%= link_to 'show', event_attendees_path(@event,@attendee) %>
```

```
class EventAttendeesController < ApplicationController  
  before_filter :find_event  
  
  def show  
    @attendees = @event.attendees.find(params[:id])  
  end  
  
  protected  
  
  def find_event  
    @event = Event.find(params[:event_id])  
  end  
end
```

Q: 為什麼不這樣寫??
Attendee.find(params[:id])

Ans:
Scope Access

Deep Nesting?

Resources should never be nested more than one level deep.

我想要加自己的 action 之問題二：
event memberships

Model design

```
class Event < ActiveRecord::Base
  has_many :memberships
  has_many :users, :through => :memberships
end

class User < ActiveRecord::Base
  has_many :memberships
  has_many :events, :through => :memberships
end

class Membership < ActiveRecord::Base
  belongs_to :event
  belongs_to :user
end
```

RESTful design I

```
map.resources :memberships

class MembershipsController < ApplicationController
  # POST /memberships?group_id=2&user_id=1
  def create end

  # DELETE /memberships/3
  def destroy end
end
```

RESTful design 2

```
map.resources :groups do |group|
  group.resources :memberships
end
```

```
class MembershipsController < ApplicationController
  # POST /group/2/memberships/?user_id=1
  def create end

  # DELETE /group/2/memberships/3
  def destroy end
end
```

我想要加自己的 action 之問題三：
event has one map

singular resource route

- 一般來說，resources 皆為複數，例如對群集操作

```
map.resources :events
```

- 但也可以定義單數的 resource

```
map.resources :events do |event|
  event.resource :map, :controller => 'event_maps'
end
```

RESTful 的 controller
一定都是複數結尾

singular resource route (cont.)

- 所有的 URL helper 皆為單數
- 也就沒有 index action
- show, edit 跟 update 的 URL Helper 也無須傳入 id

```
<%= link_to 'Login', event_map_path(@event) %>  
<% form_for :event_map, :url => event_map_path(@event) do |f| %>
```

我想要加自己的 action 之問題四：
operate event state
(open/closed)

```
map.resources :events do |event|
  event.resource :closure, :controller => 'event_closures'
end
```

```
class EventClosuresController < ApplicationController
```

```
  # POST /events/3/closure
  def create
    Event.find(params[:event_id]).close!
  end
```

```
  # DELETE /events/3/closure
  def destroy
    Event.find(params[:event_id]).open!
  end
```

```
end
```

```
<%= link_to 'close', event_closure_path(@event), :method => :post %>
```

```
<%= link_to 'open', event_closure_path(@event), :method => :delete %>
```

why not

PUT closed=1 to /events/2

這要看你怎麼想

“a separate resource” or “an attribute of event”

我想要加自己的 action 之問題五：
search event

Extra Collection Routes

```
map.resources :events, :collection => { :search => :get }
```

```
class EventsController < ApplicationController
  def search
    @events = Event.find_by_keyword(params[:keyword])
  end
end
```

```
<%= link_to 'search', search_events_path, :keyword => 'osdc' %>
```

我想要加自己的 action 之問題六：
a event dashboard

Extra Member Routes

```
map.resources :events, :member => { :dashboard => :get }
```

```
class EventsController < ApplicationController
  def dashboard
    @event = Event.find(params[:id])
  end
end
```

```
<%= link_to 'dashboard', dashboard_event_path(event) %>
```

Route Customizations is not RESTful ??

- you can think of it as a sub-resource of events resource. (and the sub-resource has only one action)
- If you have too many extra routes, you should consider another resources.

我想要加自己的 action 之問題七：
sorting event

Use query variables

- Need not new resource

```
def index
  sort_by = (params[:order] == 'name') ? 'name' : 'created_at'
  @events = Event.find(:all, :order => sort_by)
end
```

```
<%= link_to 'search', events_path, :order => 'name' %>
```

我想要加自己的 action 之問題八：
event admin

namespace

```
map.namespace :admin do |admin|
  admin.resources :events
end
```

```
# /app/controllers/admin/events_controller.rb
class Admin::EventsController < ApplicationController
  before_filter :require_admin

  def index
    ....
  end
end
```

Considerations(I)

- a REST resource does not map directly to model. It's high-level abstractions of what's available through your web app.
(Not always 1-to-1, maybe 1-to-many or 1-to-zero)
- You don't need to use all 7 actions if you don't need them.

map.resource :session

```
# This controller handles the login/logout function of the site.
class SessionsController < ApplicationController

  def create
    self.current_user = User.authenticate(params[:login], params[:password])
    if logged_in?
      redirect_back_or_default('/')
    else
      render :action => 'new'
    end
  end

  def destroy
    self.current_user.forget_me if logged_in?
    cookies.delete :auth_token
    reset_session
    redirect_back_or_default('/')
  end
end
```

使用者登入 => 建立 session

Considerations(2)

- a RESTful controller may represent the creation or delete of only a concept.

For example, a SpamsController create spam by changing a comment's status to spam without adding any records to the DB.

Considerations(3)

- one resources should be associated with one controller.
(well, you can use one controller handle more than one resources)
- offload privileged views into either a different controller or action.

`map.resources :attendees`

for event manager

```
class AttendeeController < ApplicationController  
  
  before_filter :manager_required  
  
  def show  
    @person = @event.attendees.find(params[:id])  
  end  
  
end
```

`map.resources :registers`

for attendeeing user

1 attendee Model
2 Resources related
(2 Controller)

```
class RegistersController < ApplicationController  
  
  before_filter :login_required  
  
  def show  
    @person = current_user.registers.find(params[:id])  
  end  
  
end
```

new, edit
或客製?

有無 nested

單數?複數?

[namespace]_[custom route]_[parent resource]_event[s]_path([p,] e)

:method => GET | POST | PUT | DELETE

conclusion

standardization on action name

The heart of the Rails's REST support is a technique for creating bundles of named routes automatically

From Rails Way Chap.4

not yet...

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 => @user.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....

Registrano | Attendees

<http://registrano.com/events/538/ae/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 紙參與者](#)

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
<input type="checkbox"/>	8		test@test.com		2008/03/17 4:19 PM	Confirmed	詳細顯示 編輯 移除
<input type="checkbox"/>	6		duckie@gmail.com		2008/02/24 4:44 AM	Confirmed	詳細顯示 編輯 移除
<input type="checkbox"/>	3		jojo@gmail.com		2007/10/20 2:01 PM	Pending	詳細顯示 編輯 移除
<input type="checkbox"/>	2				2007/12/19 3:47 PM	Pending	詳細顯示 編輯 移除
<input type="checkbox"/>	1		idv@gmail.com		2007/12/19 3:47 PM	Pending	詳細顯示 編輯 移除

<http://registrano.com/events/538/ae/attendees.csv>

<http://registrano.com/events/538/ae/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

在原有的架構上
新增不同格式的支援
(甚至是不同 UI 介面)

XML API
JSON API

Adobe Flex

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) %>
```



產生

```
<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

```
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
```

自訂格式 custom format

```
# config/initializers/mime_types.rb
Mime::Type.register 'audio/mpeg', :mp3?Mime::Type.register 'audio/mpegurl', :m3u
```

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

```
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
```

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

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 template

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

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



```
<h1>OSDC 2008</h1>
```

builder template

- 用 Ruby 產生 XML

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

show.xml.builder

```
<?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>
```

builder template (cont.)

- 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
```

[index.atom.builder](#)

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!!

Ajax on Rails

最簡單的 Ajax 用法

```
<% =link_to 'Terms', terms_path, :update => 'content' %>
```

```
<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

Server

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

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

RJS template

```
<%= link_to_remote 'ajax', :url => note_path(@note) %>
```

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

- 用 Ruby 來產生 Javascript

```
# show.rjs.js
page.replace_html 'content', :partial =>'note'
page.visual_effect :highlight, ' content'
```

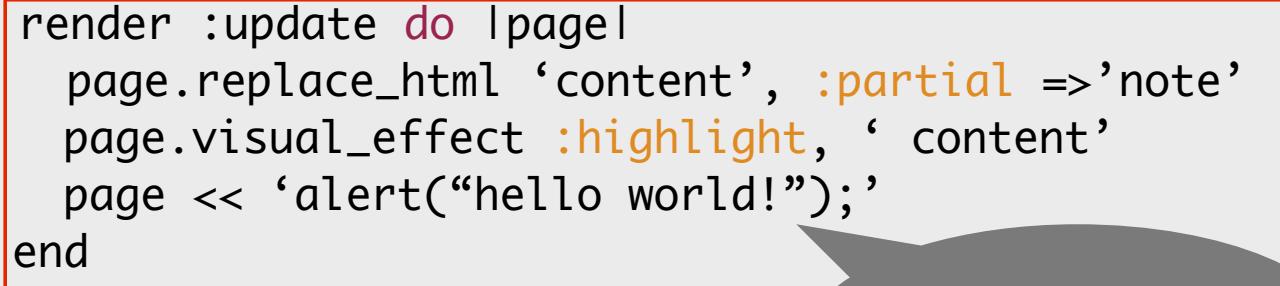
產生

```
try {
  new Element.update("content", "blah");
  new Effect.Highlight("content", {});
} catch (e) { alert('RJS error:\n\n' + e.toString()); alert('new
Element.update(\"content\", \"blah\");\nnew Effect.Highlight(\"content\", {});');
throw e }
```

Browser 執行 Server 傳回
來的 Javascript code

inline RJS

```
def show
  @note = Note.find(params[:id])
  respond_to |format|
    format.js {
      render :update do |page|
        page.replace_html 'content', :partial => 'note'
        page.visual_effect :highlight, 'content'
        page << 'alert("hello world!");'
      end
    }
  end
end
```



Write raw JavaScript

js.erb template

```
<%=link_to_remote 'ajax', :url => posts_path %>
```

```
def index
...
  respond_to |format|
    format.js
end
end
```

- Write JavaScript directly
- Rails 1.x 需要 hack!
(google MinusMOR plugin)

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

jQuery on Rails

- We love jQuery
- script/plugin install
<http://ennerchi.googlecode.com/svn/trunk/plugins/jrails>

A screenshot of a web browser window. The title bar says "jQuery on Rails". The main content area displays the jRails homepage. The page has a header with the "jRails" logo and navigation links for "Intro", "News", "Features", "Effects", "How to use", "Install", "Changelog", and "Contact". Below the header, there's an "Intro" section with a brief description of jRails as a drop-in jQuery replacement for Prototype. The "News" section features a heading for "jRails 0.3 : jQuery Enchant!" with a paragraph about the release notes. The "Features" section lists the methods provided by jRails, comparing them to Prototype, Scriptaculous, and RJS. A sidebar on the left contains a link to report bugs.

jRails
jQuery on Rails

Intro

jRails is a drop-in jQuery replacement for Prototype/script.aculo.us on Rails. Using jRails, you can get all of the same default Rails helpers for javascript functionality using the lighter jQuery library.

News

jRails 0.3 : jQuery Enchant!

This release gets all of the most recent changes and updates to the jquery Enchant and jquery UI libraries. I wrote most of the core effects that will ultimately become part of the new jQuery Enchant effect library and these put jquery on par with scriptaculous effects. This release includes some additonal fixes and updates to RJS. Next release will include jQuery.noConflict!

Features

jRails provides drop-in functionality for these existing Rails methods.

Prototype	Scriptaculous	RJS
↳ form_remote_for	↳ draggable_element	↳ hide
↳ form_remote_tag	↳ drop_receiving_element	↳ insert_html
↳ link_to_remote	↳ sortable_element	↳ remove
↳ observe_field	↳ visual_effect	↳ replace

To report a bug or feature, please use the [project site](#). Thanks!

`respond_to` 的另個好處： 支援 Graceful Degradation

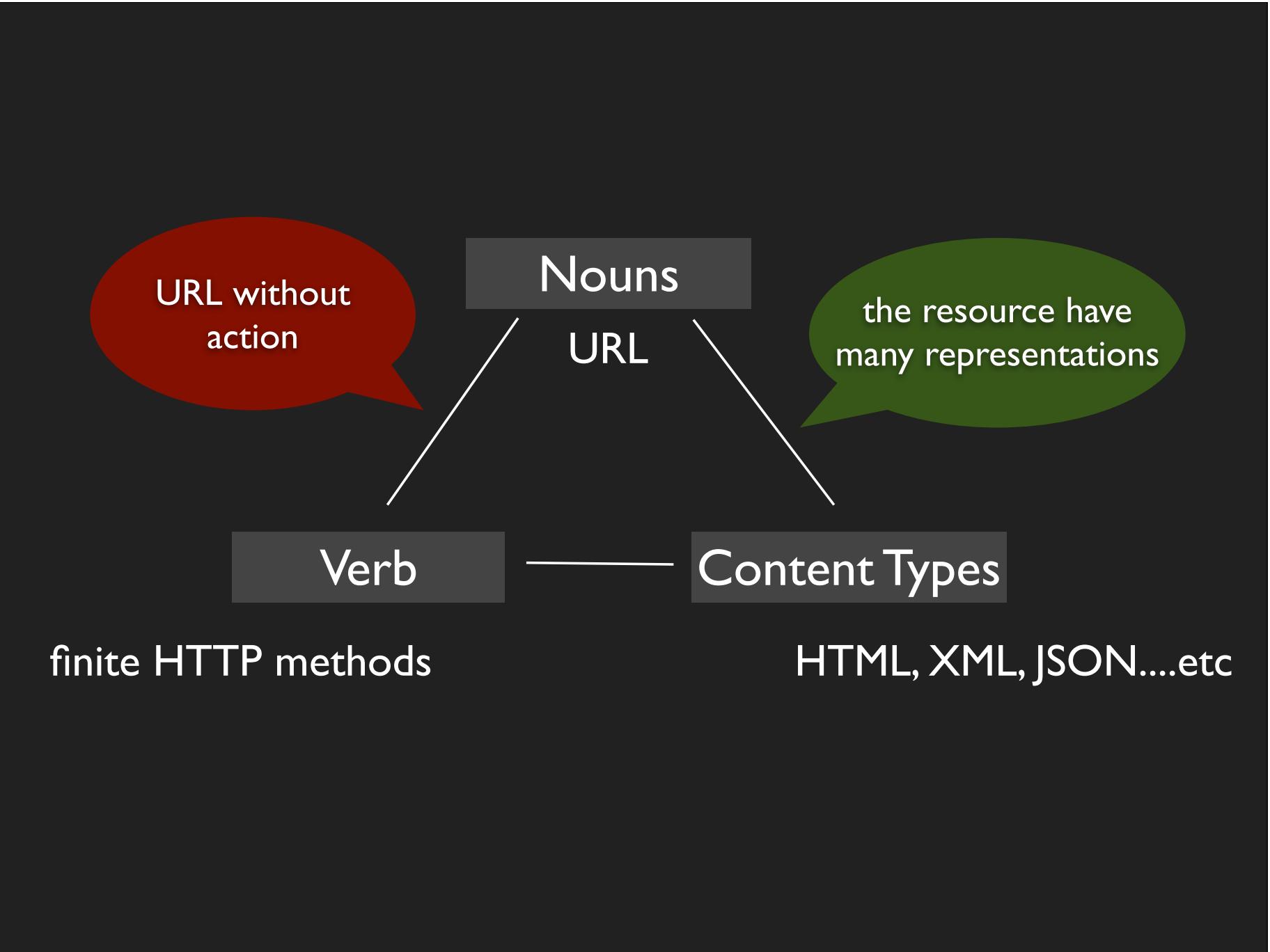
```
def index
  @users = User.find(:all)
  respond_to do |format|
    format.js {
      render :update do |page|
        page.replace_html 'content', '<p>blah</p>'
      end
    }
    format.html #index.html.erb
  end
end
```

Browser 不支援 Javascript

Browser 支援 Javascript

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

so, what's REST style?



正題講完了，來換話題：

Front-end optimization

Question:
網站放國外，連線速度很慢...

Asset Caching

- fewer request
- merge JavaScript & CSS files

```
<%= stylesheet_link_tag "layout", "basic", "base", :cache => true %>
# => public/stylesheets/all.css

<%= stylesheet_link_tag "layout", "basic", "base", "admin" :cache => "admin" %>
# => public/javascripts/admin.css

<%= javascript_include_tag "jquery", "jquery-ui", "jquery-
fx", "jrails", "application", :cache => true %>
# => # public/javascripts/all.css
```

Asset Caching(cont.)

```
<link href="/stylesheets/blueprint/lib/reset.css?1193302860" media="screen" rel="stylesheet" type="text/css">
<link href="/stylesheets/blueprint/lib/typography.css?1193302860" media="screen" rel="stylesheet" type="text/css">
<link href="/stylesheets/blueprint/lib/grid.css?1193302860" media="screen" rel="stylesheet" type="text/css">
<link href="/stylesheets/blueprint/plugins/buttons/buttons.css?1193302860" media="screen" rel="stylesheet"
type="text/css">
<link href="/stylesheets/blueprint/plugins/css-classes/css-classes.css?1193302860" media="screen" rel="stylesheet"
type="text/css">
<link href="/stylesheets/stickies.css?1190808294" media="screen" rel="stylesheet" type="text/css">
<link href="/stylesheets/facebook.css?1203995080" media="screen" rel="stylesheet" type="text/css">
<link href="/stylesheets/main.css?1206452830" media="screen" rel="stylesheet" type="text/css">
<script src="/javascripts/jquery.js?1206433332" type="text/javascript"></script>
<script src="/javascripts/jrails.js?1206433332" type="text/javascript"></script>
<script src="/javascripts/facebook.js?1204792223" type="text/javascript"></script>
<script src="/javascripts/nicEdit/bkLib.js?1206450862" type="text/javascript"></script>
<script src="/javascripts/nicEdit/nicConfig.js?1206450862" type="text/javascript"></script>
<script src="/javascripts/nicEdit/nicEdit.js?1206450862" type="text/javascript"></script>
<script src="/javascripts/application.js?1206440609" type="text/javascript"></script>
```

↓
服用 :cache => true

```
<link href="/stylesheets/all.css?1206812340" media="screen" rel="stylesheet" type="text/css">
<script src="/javascripts/all.js?1206450935" type="text/javascript"></script>
```

JS minification

- Compress Javascript using `bundle_fu` plugin

```
<% bundle do %>
  ...
  <%= javascript_include_tag "prototype" %>
  <%= stylesheet_link_tag "basic.css" %>
  <%= calendar_date_select_includes "red" %>
  <script src="javascripts/application.js" type="text/javascript"></script>
  ...
<% end %>
```



TURN THIS:

```
<script src="/javascripts/prototype.js?1187726365" type="text/javascript"></script>
<script src="/javascripts/effects.js?1187726365" type="text/javascript"></script>
<script src="/javascripts/builder.js?1187726365" type="text/javascript"></script>
<script src="/javascripts/application.js?1187726365" type="text/javascript"></script>
```



GZIP compression

- add this to your Apache VirtualHost section:

```
# deflate_module
AddOutputFilterByType DEFLATE text/html text/xml text/plain text/
css application/x-javascript text/javascript;
BrowserMatch ^Mozilla/4 gzip-only-text/html
BrowserMatch ^Mozilla/4\.0[678] no-gzip
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
```

Add expires Header

- add this to your Apache VirtualHost section:

```
<Directory "/your_rails/public">
  FileETag none
  ExpiresActive On
  ExpiresDefault "access plus 1 year"
</Directory>
```

Asset Servers

- Browser is limited to make only two connections to any single domain.

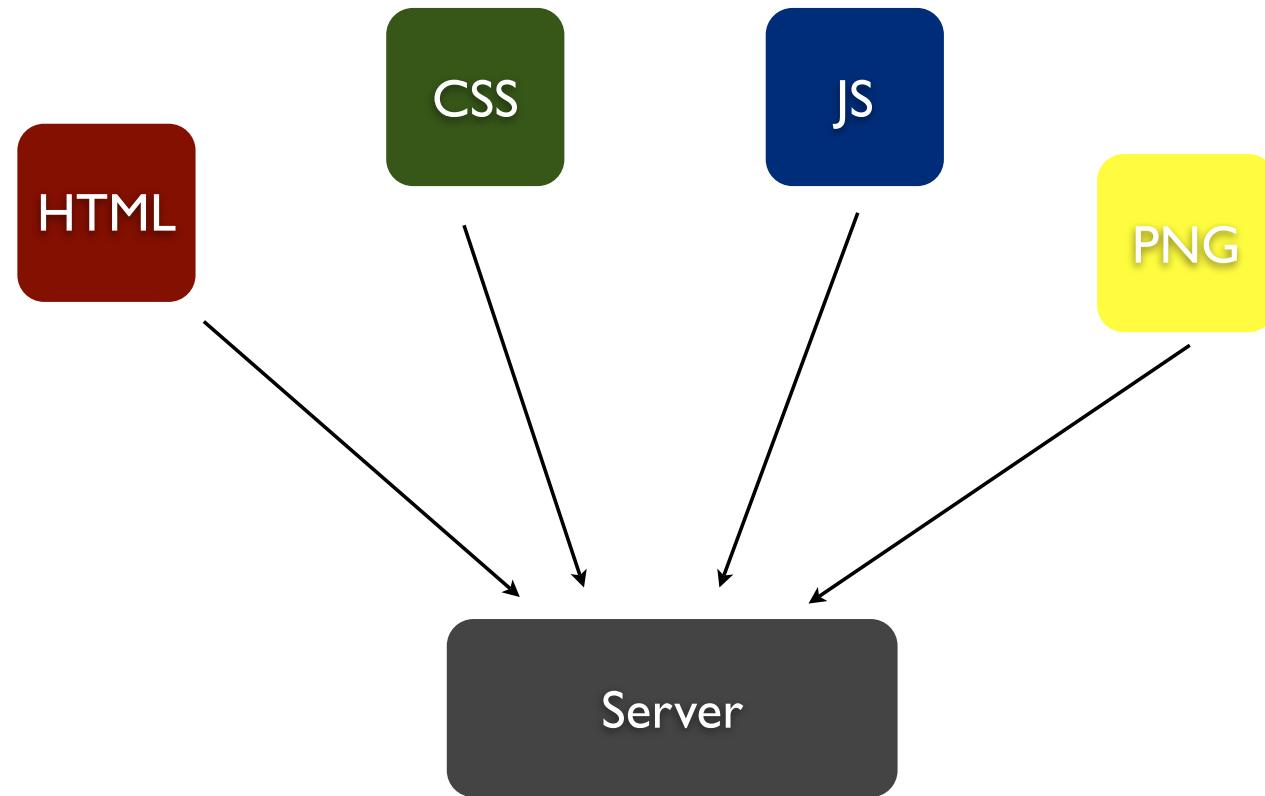
google
"HTTP 連線管理"

```
# Enable serving of images, stylesheets, and javascripts from an asset server
config.action_controller.asset_host = "http://asset%d.example.com"
```

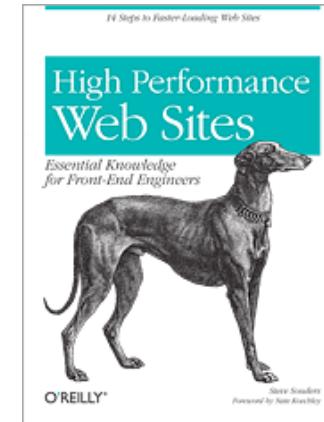


```
http://asset0.example.com
http://asset1.example.com
http://asset2.example.com
http://asset3.example.com
```

Asset Servers



- **High Performance Web Sites:**
Essential Knowledge for Front-End Engineers
- **YSlow Firefox extension**
<http://developer.yahoo.com/yslow/>



YSlow for Firebug

http://developer.yahoo.com/yslow/

Developer Network Home Help

YAHOO! DEVELOPER NETWORK

YSlow for Firebug

Get Started

- Home
- About Us
- Developer Network Blog
- YDN Theater
- Frequently Asked Questions
- Support Communities
- Got Suggestions?
- Developer Central**
- Java Developer Center
- JavaScript Developer Center
- Flash Developer Center
- ColdFusion Developer Center
- .NET Developer Center
- PHP Developer Center
- Python Developer Center
- Ruby Developer Center
- Silverlight Developer Center
- Lessons and Libraries**
- Applications Gallery
- Browser Based Authentication
- Design Patterns
- Exceptional Performance
- Security Best Practices
- YUI - Yahoo! User Interface

Site Search Search

find code

Speed up your web pages with YSlow

YSlow analyzes web pages and tells you why they're slow based on the [rules for high performance web sites](#). YSlow is a [Firefox add-on](#) integrated with the popular [Firebug](#) web development tool. YSlow gives you:

- Performance report card
- HTTP/HTML summary
- List of components in the page
- Tools including [JSLint](#)

[Download Now](#)

Install [Firebug](#) first!

Visit Yahoo! [Exceptional Performance](#) on YDN.

Podcast, Screencast

Take a look at the [YSlow Podcast and Screencast](#).

Performance View

YSlow analyzes any web page and generates a grade for each rule and an overall grade. If a page can be improved, YSlow lists the specific changes to be made.

Inspect | Performance | Stats | Components | Tools ▾ | Help ▾

Console | HTML | CSS | Script | DOM | Net | YSlow

Options ▾

Performance Grade: A (92)

Expand All | Collapse All

C 1. Make fewer HTTP requests ▾

Session in Rails

Session in Rails(I)

- Rails 1.x, default is filesystem-based session
- easy configure
- Poor performance
- Not scale, difficult to maintain
- Crashing halt if thousands of session files

Session in Rails(2)

- we have ActiveRecord-based session
- can scale
- query DB every request
- purge session records regularly

```
ruby script/generate session_migration  
rake db:migrate
```

```
config.action_controller.session_store = :active_record_store
```

Session in Rails(3)

- Rails2's default is cookie-based session
- easy configure
- performance
- scale
- limited to 4k

```
config.action_controller.session = {  
  :session_key => '_my_app_session',  
  :secret       => 'some_really_long_and_hashed_key'  
}
```

enough for user ID and
flash message

REST end.

(不知道還剩多少時間?)

Bonus:

Write your rake

rake

- 依照 dependence 順序執行任務
- rake -T
- write your rake in /lib/tasks/foo.rake

```
task(:a => [:b, :c] ) do
    # do some thing
end
```

rake a → execute task b, task c first

example 1: “rake send_emails”

```
task :environment do
  require(File.join(RAILS_ROOT, 'config', 'environment'))
end

desc "Send latest announcement mail to all users"
task(:send_emails => :environment) do
  # Find users to email
  User.find(:all).each do |user|
    puts "Emailing #{user.name}"
    UserNotifier.deliver_latest_announcement(user)
  end
end
```

example 2: “rake dev:build”

```
namespace :dev do

  desc "Rebuild system"
  task :build => ["tmp:clear", "db:drop", "db:create", "db:migrate", :setup, :fake]

  desc "Setup system data"
  task :setup => :environment do
    puts "Attempts to create system user"
    u = User.new( :login => "root", :password => 'password' )
    u.save!
  end

  desc "Create fake data"
  task :fake => :environment do
    Event.create!( :title => 'osdc 2008' )
    Event.create!( :title => 'osdc 2009' )
    Event.create!( :title => 'osdc 2010' )

    ...
    ...
  end

end
```

example 2: “rake dev:build”

```
namespace :dev do

  desc "Rebuild system"
  task :build => ["tmp:clear", "db:drop", "db:create", "db:migrate", :setup, :fake]

  desc "Setup system data"
  task :setup => :environment do
    puts "Attempts to create system user"
    u = User.new( :login => "root", :password => 'password' )
    u.save!
  end

  desc "Create fake data"
  task :fake => :environment do
    Event.create!( :title => 'osdc 2008' )
    Event.create!( :title => 'osdc 2009' )
    Event.create!( :title => 'osdc 2010' )

    ...
    ...
  end

end
```

example 2: “rake dev:build”

```
namespace :dev do

  desc "Rebuild system"
  task :build => ["tmp:clear", "db:drop", "db:create", "db:migrate", :setup, :fake]

  desc "Setup system data"
  task :setup => :environment do
    puts "Attempts to create system user"
    u = User.new( :login => "root", :password => 'password' )
    u.save!
  end

  desc "Create fake data"
  task :fake => :environment do
    Event.create!( :title => 'osdc 2008' )
    Event.create!( :title => 'osdc 2009' )
    Event.create!( :title => 'osdc 2010' )

    ...
    ...
  end
end
```

other usage

- cron 定期執行，例如產生報表
- loop { do_some_thing; sleep 5;} end
一個簡單的 daemon
- Avoid Heavy Response Processing

Bonus:
Capistrano

自動化 deploy 步驟

1. ssh to production server
2. svn checkout
3. run some your script (link file/copy config file/ clear cache...etc)
4. restart mongrel cluster

目錄結構



目錄結構



cap deploy

The screenshot shows a web browser window with the title "Capistrano: Home" at the top. The address bar contains the URL "http://capify.org/". The browser's toolbar includes standard icons for back, forward, search, and refresh. Below the toolbar, there is a tab bar with several open tabs, including "Gmail - Inbox - ihow...", "Twitter", "Yahoo!奇摩字典", and "Capistrano: Home". The main content area displays the Capistrano homepage. At the top of the page is a navigation bar with links to "Home", "Download", "Install", "Community", "Getting Started", and "Upgrading". A red banner below the navigation bar reads "Upgrading from 1.x to 2.x? Click here for important information!". The central feature is a large blue button with the word "Capistrano" in white. Above the button is a stylized illustration of a tropical island with a palm tree, a sunset, and a small figure. Below the button, the text "Simple. The way it should be." is displayed in a bold, italicized font. A purple header bar contains the text "Capistrano is...". Below this, a bulleted list describes the software:

- Great for automating tasks via SSH on remote servers, like **software installation, application deployment, configuration management**, ad hoc **server monitoring**, and more.
- Ideal for **system administrators**, whether professional or incidental.
- Easy to customize. Its configuration files use the **Ruby** programming language syntax, but you don't need to know Ruby to do most things with **Capistrano**.
- Easy to extend. **Capistrano** is written in the **Ruby** programming language, and may be extended easily by writing additional Ruby modules.

another choice:

Vlad



The screenshot shows a web browser window titled "Vlad the Deployer". The address bar contains the URL http://rubyhitsquad.com/Vlad_the_Deployer.html. The browser's toolbar includes standard icons for back, forward, search, and refresh. The menu bar has a "Google" search field. The main content area displays the Vlad the Deployer website. At the top, there is a navigation bar with links to "Ruby Hit Squad", "Vlad the Deployer", and "About the Hit Squad". Below this is a large logo featuring a silhouette of a person climbing a pole, with the text "Vlad the Deployer" in a large, bold, sans-serif font. A subtitle "Pragmatic application deployment automation, without mercy." is present. To the right of the logo is a video player window showing a thumbnail of a video titled "Vlad the Deployer" with a play button. Below the video player is a link "Watch the screencast!". Further down, there are sections for "Project Goals:" and "Download & Install". The "Project Goals:" section lists nine bullet points. The "Download & Install" section features the command "sudo gem install vlad" in a stylized green font.

Vlad the Deployer

http://rubyhitsquad.com/Vlad_the_Deployer.html

iGoogle Twitter Blog Flickr GitHub PlanetRoR RubyFlow Yahoo! Google Translate Yahoo!字典 Trac ST myCATCH Registrano localhost

Gmail – Inbox – ihow... Twitter Yahoo!奇摩字典 {ihower.idv.tw} blog ... vlad – Google Search Capistrano: Home Vlad the Deployer

Ruby Hit Squad Vlad the Deployer About the Hit Squad

Vlad the Deployer

Pragmatic application deployment automation, without mercy.

Project Goals:

- Do the simplest thing that could possibly work.
- Nothing to 1.0 in four(ish) days.
- Targets the 80% use case.
- Uses [Rake](#), as god intended.
- Use the right tool for the job ([ssh](#), [rsync](#), etc).
- Fold in the [Rails Machine recipes](#).
- Clever is **bad**. Period.

Download & Install

sudo gem install vlad

Bonus:

Facebox Render plugin

Facebox

- Facebox is a JQuery-based lightbox
<http://famspam.com/facebox/>



Rails 無縫使用 Facebox

```
class ApplicationController < ActionController::Base
  include FaceboxRender
end
```

```
<%= facebox_link_to "Login", :url => login_url %>
```

```
def new
  # do some thing you want
  respond_to do |format|
    format.html
    format.js { render_facebox }
  end
end
```

Registrano | Hosting events
http://localhost:3000/hosting_events#/events/03087d/attendees

iGoogle Twitter Blog Flickr GitHub PlanetRoR RubyFlow Yahoo! Google Translate Yahoo字典 Trac ST myCATCH Registrano localhost

iGoogle Twitter / natekoe... Yahoo!奇摩字典 {ihower.idv.tw| bl... Rails Envy: Ruby o... Ruby on Rails中的... Rake -- Ruby Make Registrano | Host...

Registrano

你的活動平台

Hi, Admin (settings). Logout?

Home Hosting Events Attending Events My Groups Admin

Hosting Events > Java Conference > View Attendees

Java Conference

Dashboard View Attendees View Cancelled Attendees Reports Integration

Create new attendee E-Mail to attendees

Batch Action: Mail Batch

20 records per page, sorted by Register time Submit

all / none	Reg	<% form_remote_tag :url => batch_event_attendees_path(@event) do %>							
<input checked="" type="checkbox"/>	26	wilbur	wilbur@registrano.com		2008/06/08 4:34 PM	Show	Edit	Remove	
<input checked="" type="checkbox"/>	25	webster	webster@registrano.com		2008/06/05 4:34 PM	Show	Edit	Remove	
<input type="checkbox"/>	24	wayne	wayne@registrano.com		2008/06/04 4:34 PM	Show	Edit	Remove	
<input type="checkbox"/>	23	wade	wade@registrano.com		2008/05/29 4:34 PM	Show	Edit	Remove	
<input type="checkbox"/>	22	william	william@registrano.com		2008/05/27 4:34 PM	Show	Edit	Remove	
<input type="checkbox"/>	21	warren	warren@registrano.com		2008/05/25 4:34 PM	Show	Edit	Remove	
<input type="checkbox"/>	20	thomas	thomas@registrano.com		2008/05/17 4:34 PM	Approved	Show	Edit	Remove
<input type="checkbox"/>	19	tom	tom@registrano.com		2008/05/16 4:34 PM	Approved	Show	Edit	Remove
<input type="checkbox"/>	18	stuart	stuart@registrano.com		2008/05/11 4:34 PM	Approved	Show	Edit	Remove
<input type="checkbox"/>	17	rex	rex@registrano.com		2008/04/22 4:34 PM	Approved	Show	Edit	Remove
<input type="checkbox"/>	16	randy	randy@registrano.com		2008/04/21 4:34 PM	Approved	Show	Edit	Remove
<input type="checkbox"/>	15	patrick	patrick@registrano.com		2008/04/18 4:34 PM	Approved	Show	Edit	Remove
<input type="checkbox"/>	14	oliver	oliver@registrano.com		2008/04/15 4:34 PM	Approved	Show	Edit	Remove
<input type="checkbox"/>	13	ken	ken@registrano.com		2008/03/31 4:34 PM	Approved	Show	Edit	Remove

Ajax form submit

The screenshot shows a web application interface for managing event attendees. The main page has a navigation bar with 'Home', 'Hosting Events' (selected), 'Attending Events', 'My Groups', and 'Admin'. Below this, a breadcrumb trail shows 'Hosting Events > Java Conference > View Attendees > Remove Attendees'. A modal dialog is open, titled 'Send message to 2 attendees'. It contains fields for 'Email Subject' (set to 'Java Conference') and 'Reply Email Address' (set to 'registrano@handlino.com'). A table lists 26 attendees with checkboxes. The first two rows have checkboxes checked. A large text area labeled 'Message' is present, but its content is not visible. A 'Submit' button is at the bottom left of the modal. A 'CLOSE' button with an 'X' is at the bottom right. The background shows a list of attendees with columns for ID, Name, Email, Date, Status, and actions ('Show', 'Edit', 'Remove').

```
def batch
...
respond_to do |format|
  format.html
  format.js { render_to_facebox }
end
end
```

batch.html.erb

Usage

gem install facebox-render & unpack it

```
render_facebox #default is action_name.html.erb  
  
render_facebox :template => 'other.html.erb'  
render_facebox :action => :index  
render_facebox :partial => 'foo.html.erb'  
render_facebox :html => '<p>bar</p>'
```

Blog:

<http://handlino.com/blog/2008/02/26/57/>

Source code:

<http://github.com/ihower/facebox-render/>

Bonus:
SPAkit plugin

Transform existed website to single page application (SPA)

所有的操作皆為 Ajax request，不 reload 頁面



Ajax update
#content

`<%= spakit_link_to("Login" , :url => new_session_path) %>`

```
def new
  respond_to do |format|
    format.html
  end
end
```

Usage

gem install spakit & unpack it

```
<%= spakit_link_to 'new person', :url => new_person_path %>
```

```
<%= spakit_form_for @person, :url => people_path %>
```

Source code:

<http://github.com/ihower/spakit/>

Blog:

<http://handlino.com/blog/2008/02/17/52/>

Really Simple History (RSH): Ajax history and bookmarking library

reallysimplehistory – Google Code

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

iGoogle Twitter Blog Flickr GitHub PlanetRoR RubyFlow Yahoo! Google Translate Yahoo字典 Trac ST myCATCH Registrano localhost

Gmail - Inbox (...) Twitter Yahoo!奇摩字典 ihower.idv.tw... Rails Envy: Rub... Ruby on Rails中... Rake -- Ruby ... ihower's faceb... reallysimplehis...

ihower@gmail.com | Settings | What's new? | Help | My Account | Sign out

Search Projects Search the Web

reallysimplehistory
Really Simple History (RSH): Ajax history and bookmarking library

Project Home Downloads Wiki Issues Source

Really Simple History is a lightweight JavaScript library for the management of bookmarking and browser history in Ajax/DHTML applications. RSH serializes application data in an internal JavaScript cache so that bookmarks and the back button can be used to return your application to an earlier state.

Originally developed by [Brad Neuberg](#) in 2005, RSH has won a large user base for its simplicity, ease of use and lack of dependency on any specific Ajax framework. It's written in plain old JavaScript and can be included on any site according to the terms of its BSD license.

In September 2007, [Brian Dillard](#) of [Pathfinder Development](#) came on board as maintainer and ongoing code steward for the project. A new version, with support for additional modern browsers (IE7, Safari, Opera), was released as RSH 0.6 on December 3, 2007.

Quick Links

- [Current stable production release \(12.03.2007\)](#)
- [Usage instructions and examples \(11.09.2007\)](#)
- [Overview and version history](#)
- [Google Group for user discussion and feedback](#)
- [Ajax bookmarking links page](#)
- [RSH tag on Brian's Blog, Agile Ajax](#)
- Brian's email: bdillard(at)pathf(dot)com.

News

- 12.03.2007: Released RSH 0.6; deprecated 0.4 and 0.6 RC1.
- 11.12.2007: Continuing with minor, backwards-compatible enhancements in the SVN repository.
- 11.11.2007: Updated [Usage Instructions](#) page to point out dangers of trying to initialize RSH from DOMContentLoaded rather than window.onload.
- 11.09.2007: 0.6RC1 now available for download; previous beta now deprecated.
- 11.06.2007: Continue to clean up bugs and enhancements; release candidate coming.
- 10.30.2007: Committed changes to SVN to fix Opera implementation for maintenance of history after navigating to another domain and back.
- 10.28.2007: Committed changes to SVN that fix one of the major issues with Safari/Mac: maintenance of history after navigating to another domain and back.
- 10.28.2007: Created a [Google Group](#) to foment user discussion, especially of the new beta release.
- 10.24.2007: RSH now has an [SVN repo](#); small fixes to 0.6 have already been committed.
- 10.23.2007: RSH 0.6 is now ready for beta-testing. See the [0.6Beta release notes](#) for more info.
 - [Ajaxian post](#)

License: [New BSD License](#)
Labels: [javascript](#), [dhtml](#), [ajax](#), [browserhistory](#)

Featured Downloads:
[RSH0.6FINAL.zip](#) Show all

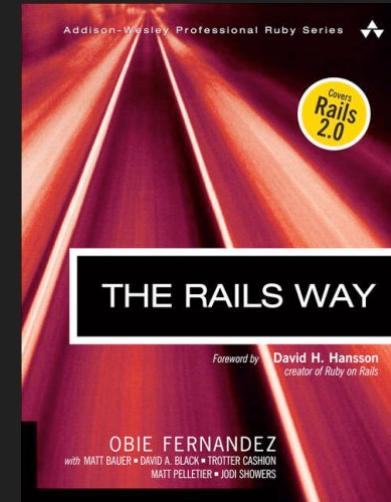
Links: [RSH Training & Consulting](#)
Blogs: [Brian Dillard @ Agile Ajax](#)
[Brad Neuberg @ Coding in Paradise](#)

Groups: [Really Simple History](#)

Project owners: [Join project](#)
[bdpathfinder](#), [bradneuberg](#)

Reference books:

- **Rails Way** (Addison Wesley)
- **Advanced Rails** (Pragmatic)
- **Code Review PDF** (peepcode.com)
- **Rails2 PDF** (peepcode.com)
- **RESTful Web Services** (O'REILLY)



thank you.

Q: 用 Rails 接不到案子
怎麼辦?

Acts as ASP.NET

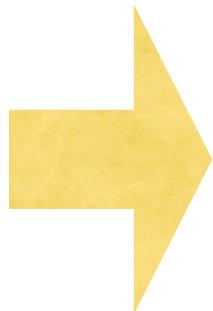
script/plugin install

http://actsasaspdotnet.googlecode.com/svn/trunk/acts_as_aspdotnet

Viewstate

```
<form action="/people" class="new_person" id="new_person" method="post">
<p>
<b>Name</b><br />
<input id="person_name" name="person[name]" size="30" type="text" />
</p>

<p>
<b>Email</b><br />
<input id="person_email" name="person[email]" size="30" type="text" />
</p>
</form>
```

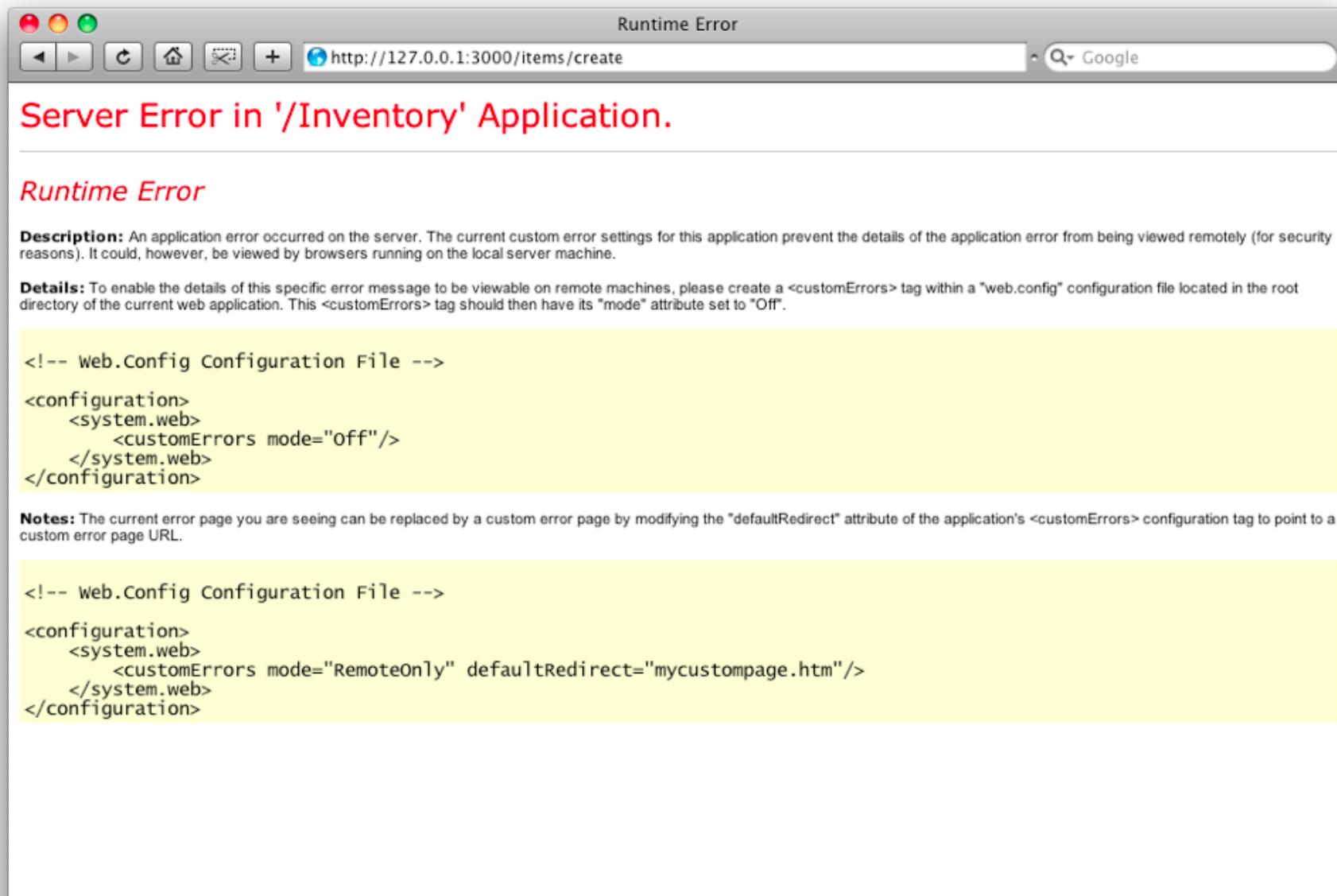


```
<form action="/People/Default.aspx" class="new_person" id="new_person" method="post">
<input id="__EVENTTARGET" name="__EVENTTARGET" type="hidden" value="" />
<input id="__EVENTARGUMENT" name="__EVENTARGUMENT" type="hidden" value="" />
<input id="__VIEWSTATE" name="__VIEWSTATE" type="hidden" value="/
XtplIo7SVYvjZC8ffNqtkHIhP5TdNKRLNnFL5mTJuprao/NDMrAz0XmYnpUv56K0gBNK+sN0yT5EJsxU
+BtzhGUEb6Pmn5u1wGQHm4nt0RTpCZ1mPGkaQ5cgs6jbu0C21lQcb68SkVH0h3EDyx6PCn3tkg4+BgVfm/
h0j4U1520V81VVPoJLnmuTrhUZ+GRvbTR7btLYWqCUozv2Da9yR0dQmUGD6PohHm+/es3kSQLgXH
+EWYee9EaHW2dyTNm46YhRVuHgbCHkko5kI
+7bwyNmtrUT70h20q0YFiU2whPf0z7e9iK30xfuCqxV8vW1MULKv4QldC6s1a/ZQ1HdIxBr/
wwvTIEQkN6bCm92DgKEQNZT5T+W5pQV2NHugb9p/d2VFN+poEvet3nNvq5693qsg0R7B0xiXIc2vhDmI7KkB/iXos6obvs/
X+WX1llJkgzfCezhLNCTBoBheGhy69cGz8doRxqzId9BFdaI8Y4Ng+HgRgFLWcqdeat34dv20zhxSJn4Svv=" />

<p>
<b>Name</b><br />
<input id="Page11__ctl1_person_name" name="person[name]" size="30" type="text" />
</p>

<p>
<b>Email</b><br />
<input id="Page12__ctl8_person_email" name="person[email]" size="30" type="text" />
</p>
</form>
```

Beautiful Error Pages



The screenshot shows a web browser window titled "Runtime Error". The address bar displays "http://127.0.0.1:3000/items/create". The main content area shows a red header: "Server Error in '/Inventory' Application." Below it, a red section is titled "Runtime Error". It contains two paragraphs of black text: "Description: An application error occurred on the server. The current custom error settings for this application prevent the details of the application error from being viewed remotely (for security reasons). It could, however, be viewed by browsers running on the local server machine." and "Details: To enable the details of this specific error message to be viewable on remote machines, please create a <customErrors> tag within a "web.config" configuration file located in the root directory of the current web application. This <customErrors> tag should then have its "mode" attribute set to "Off".". Two code snippets are shown in yellow boxes: the first snippet shows the XML configuration for "mode='off'", and the second snippet shows the XML configuration for "mode='RemoteOnly'" with a "defaultRedirect" attribute.

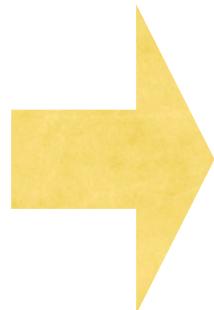
```
<!-- Web.Config Configuration File -->
<configuration>
    <system.web>
        <customErrors mode="off"/>
    </system.web>
</configuration>
```

Notes: The current error page you are seeing can be replaced by a custom error page by modifying the "defaultRedirect" attribute of the application's <customErrors> configuration tag to point to a custom error page URL.

```
<!-- Web.Config Configuration File -->
<configuration>
    <system.web>
        <customErrors mode="RemoteOnly" defaultRedirect="mycustompage.htm"/>
    </system.web>
</configuration>
```

.aspx URL

map.resources :people



/People/Default.aspx
/People/Default.aspx

/People/I>Show.aspx
/People/I>Show.aspx
/People/I>Show.aspx

/People/New.aspx
/People/I>Edit.aspx



Listing people

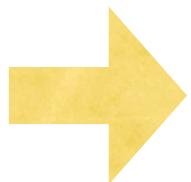
Name

zzzz222 [Show](#) [Edit](#) [Destroy](#)
3123 [Show](#) [Edit](#) [Destroy](#)

[New person](#)

Meta tag

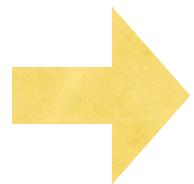
```
<%= aspdotnet_includes %>
```



```
<meta name="vs_showGrid" content="False">
<meta name="GENERATOR" Content="Microsoft Visual Studio 7.0">
<meta name="CODE_LANGUAGE" Content="C#">
<meta name="vs_defaultClientScript" content="JavaScript">
<meta name="vs_targetSchema" content="http://
schemas.microsoft.com/intellisense/ie5">
```

Element ID

```
<input id="person_name" name="person[name]" size="30" type="text" />
```



```
<input id="Page17__ctl3_person_name" name="person[name]" size="30"  
type="text" />
```

end.

(真的沒有了)