

# Rails 2.1

xdite  
2008 COSCUP

# 自我介紹

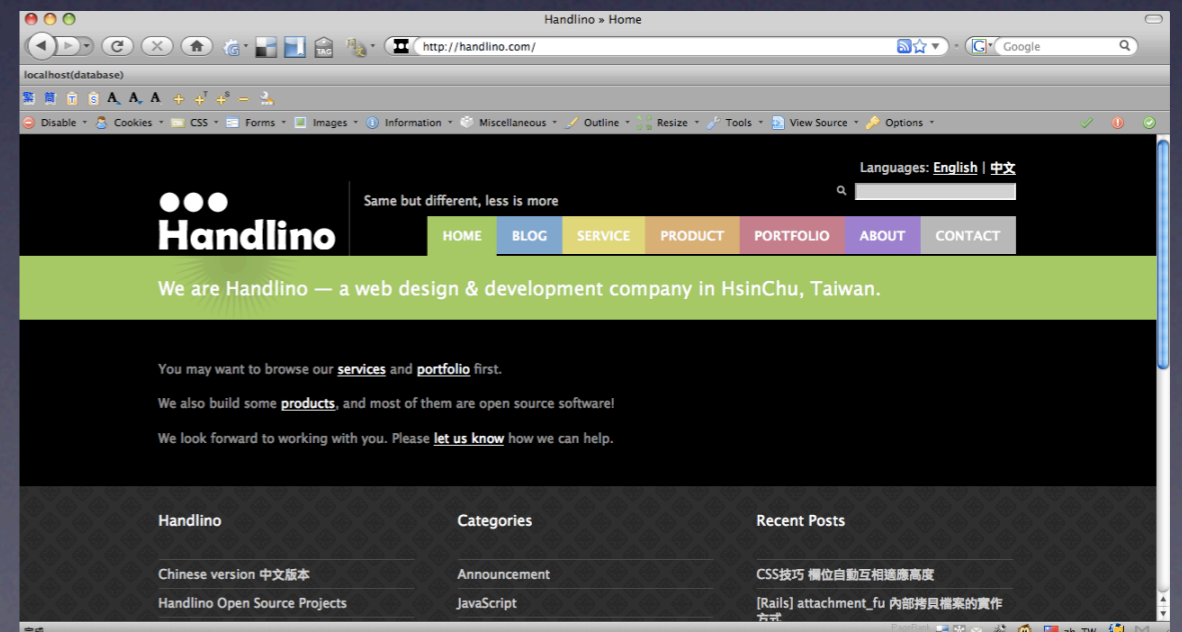


<http://blog.xdite.net>

低調部落客

<http://handlino.com>

Rails Developer



# Project Using Rails 2.1



Registrano.com

sudomake.com



VeryXD.net



# Ruby on Rails

- Ruby Web Framework
- 2004/07 DHH
- Don't Repeat Yourself
- Convention Over Configuration
- 1.2.6 / 2.0 / 2.1

為什麼要使用 Rails 2.x ?

668 → 328

<http://blog.xdite.net/?p=597>

# 程式碼大幅減少

- 使用 REST 減少 action 的使用  
( 缺點 : controller 變多 )
- named\_route 的廣泛使用  
( blah\_path )
- 利用 named\_scope 取代 find 條件的重複  
撰寫

# Practical Rails2

2008 OSDC.tw ihowe



Rails 2.1 only today.

# Rails 2.1 Include...

- Named scope
- Dirty Tracking
- UTC-based migration
- TimeZone
- Gem Dependencies
- Better Caching

# Named Scope

- 過去
  - 使用 Method
  - 無法串接
- 現在
  - 使用 `named_scope`
  - 可以無限串接

```
class Loli < ActiveRecord::Base
```

```
  belongs_to :parents
```

```
  named_scope :cute, :conditions => { :face_type => "cute" }
```

```
  named_scope :young, lambda { |*args| { :conditions => ["born  
    _at < ? and born at > ?", Time.now , 12.years.ago]} } }
```

```
  named_scope :catchable, :include => :parents, :conditions =>  
    { 'parents.around' => false }
```

```
end
```

```
@lolis = Loli.cute.young.catchable
```

當然還可以 paginate ...

```
class Loli < ActiveRecord::Base
```

```
  named_scope :uncaught, :conditions => { :caught => "false" } do
```

```
    def catch
```

```
      each { |p| p.updated_attributes(:caught, true) }
```

```
    end
```

```
  end
```

```
end
```

```
kindergarten.lolis.uncaught.catch
```

# Dirty Tracking

- 能夠知道物件是否被修改
- 並且能追蹤到最新的變更
- 延伸出 Partial Update

```
loli = Loli.find(20)
```

```
loli.status # => “活蹦亂跳”
```

```
loli.status = “乾癟地磚”
```

```
loli.status_changed? # => true
```

```
loli.status_was # => “活蹦亂跳”
```

```
loli.status_change # [“活蹦亂跳”, “乾癟地磚”]
```

```
loli.changes # [“status” => [“活蹦亂跳”, “乾癟地磚”]]
```

```
loli = Loli.find(20)
```

```
loli.status # => “活蹦亂跳”
```

```
loli.status = “乾癢地磚”
```

```
loli.save
```

# Partial Update

舊法 SQL 產生語句：

```
UPDATE 'lolis' SET 'status' = '乾癢地磚', 'owner' = 'slzpz' where 'id' = 20
```

新法 SQL 產生語句：

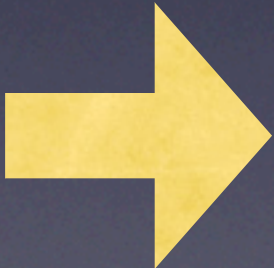
```
UPDATE 'lolis' SET 'status' = '乾癢地磚' where 'id' = 20
```



# UTC-based migration

- 解決多人共同開發時的 migration 問題  
(SVN / GIT)

```
db/migrate/012_create_lolis.rb  
db/migrate/012_create_parents.rb
```



```
db/migrate/20080613041714_create_lolis.rb  
db/migrate/20080613041918_create_parents.rb
```

# TimeZone

在 config/environment.rb 設定

```
config.time_zone = "Taipei"
```

儲進資料庫 => UTC 時間

讀出資料庫 => Taipei 時間

# Gems Dependencies

在 environment.rb 設定

```
config.gem "hpricot", :version => "0.6"  
               :source => "http://blah.blah.com"
```

```
config.gem "aws-s3", :lib => "aws/s3"
```

```
rake gems:install  
rake gems:build
```

# Better Caching

- 可自行撰寫 Cache module
- 另外指定 Cache engine

```
config.cache_store = :memory_store
```

```
config.cache_store = :file_store, "/path/to/cache/directory"
```

```
config.cache_store = :drb_store, "druby://localhost:9192"
```

```
config.cache_store = :mem_cache_store, "localhost"
```

# ActiveRecord

- `has_one :through`
- Eager Loading
- Polymorphic URL
- Change Table

# Eager Loading

```
Author.find(:all, :include => [:posts, :comments])
```

```
SELECT authors."id"           AS t0_r0
       authors."created_at"    AS t0_r0
       posts."id"              AS t1_r0
       posts."author_id"       AS t1_r1
       posts."created_at"      AS t1_r2
       comments."id"           AS t2_r0
       comments."author_id"    AS t2_r1
       comments."created_at"   AS t2_r2
FROM
  authors
LEFT OUTER JOIN posts ON post.author_id = author_id
LEFT OUTER JOIN comments ON comments.author_id = authors.id
```

# Eager Loading

```
Author.find(:all, :include => [:posts, :comments])
```

```
SELECT * from "authors"
```

```
SELECT posts.* FROM "posts" WHERE  
(posts.author_id IN(I))
```

```
SELECT comments.* FROM "comments" WHERE  
(comments.author_id IN(I))
```

# Polymorphic URL

```
polymorphic_url (article)  
#-> article_url(article)
```

```
@article = Article.find(I)  
@comment = Comment.find(I)
```

```
polymorphic_path (comment)  
#->comment_path(comment)
```

```
polymorphic_url ([:admin],@article,@comment)  
#-> admin_article_url(@article,@comment)
```

```
edit_polymorphic_url (comment)  
#->edit_comment_path(comment)
```



# Change Table

```
change_table :videos do |t|
  t.timestamps
  t.belongs_to :goat
  t.string :name, :email, :limit => 20
  t.remove, :name, :email
end
```

column, index, timestamps, change, change\_default,  
rename, references, blah ....

# Active Support

- Json Escape

`json_escape` ( 類似 `html_escape` )

```
ActiveSupport.escape_html_entities_in_json = true
```

- Removing whitespaces with 'squish' method

```
a = " 122233 44 55 666 "  
a.strip = "123344 44 55 666"  
a.squish = "122233 44 55 666"
```

# Action Pack

- 在 views 之外存取 Helper

`ApplicationController.helpers.buddy_icon(user)`

# Action Controller

- Routes Recognition

之前的是一個一個檢查是否符合規則  
現在的作法是依照路徑生成路由樹  
只要尋找路徑上的路由規則即可

recognition\_optimization.rb



快了 2.7 倍！

# Action View

- `register_javascript_expansion`
- `register_stylesheet_expansion`

在 `init.rb` 中設定

```
ActionView::Helper::AssetTagHelper.register_javascript_expansion :monkey => ["head", "body", "tail"]
```

在 View 裡使用

```
<%= javascript_include_tag :monkey %>
```

# Railties

- Freeze `rake rails:freeze:edge RELEASE = 1.2.0`
- 支援 Git `script/plugin install git://blah.git`
- 常數更直覺

# 常數

- ‘#{RAILS\_ROOT}/public\_html’  
=> `Rails.public_path`
- `RAILS_DEFAULT_LOGGER`  
=> `Rails.logger`
- `RAILS_ROOT` => `Rails.root`
- `RAILS_ENV` => `Rails.env`
- `RAILS_CACHE` => `Rails.cache`
- `RAILS::VERSION::STRING` => `Rails.version`

What's new in Rails 2.2 ?



# Rails 2.2

- Thread Safe
- Simple Localization

Thank You!