



RUBY ON RAILS 2.1

WHAT'S NEW ?

```
def temp_p
  p = temp_p.s
  p.respond_to?(:path) ? p.path : p.s
end

def render_partial(partial_path, object_assigns = nil, local_assigns = nil)
  case partial_path
  when String, Symbol, NilClass
    path, partial_name = partial_pieces(partial_path)
    object = extracting_object(partial_name, object_assigns)
    local_assigns = local_assigns ? local_assigns.clone : {}
    add_counter_to_local_assigns!(partial_name, local_assigns)
    add_object_to_local_assigns!(partial_name, local_assigns, object)

    if logger && logger.debug?
      ActionController::Base.benchmark("Rendered #{path}/_#{partial_name}")
      render("#{path}/_#{partial_name}", local_assigns)
    end
  end
end
```

CARLOS BRANDO REVIEW: MARCOS TAPAJÓS - COVER: DANIEL LOPES

TRANSLATION: IceskYsl, jesse.cai, suave.su, dongbin, rubyjin, 404, ashchan, cash.zhao

EDITOR: Libin Pan

CHINESE VERSION

Ruby on Rails 2.1

新特性介绍

第二版 (中文版)

Ruby on Rails 2.1

新特性介绍

第二版 (中文版)

Carlos Brando
Marcos Tapajós

© Copyright 2008 Carlos Brando. All Rights Reserved.

Second edition: June 2008

Carlos Brando

Website: www.nomedojogo.com

Marcos Tapajós

Website: www.improveit.com.br/en/company/tapajos

Chapter 1

概述(Introduction)

2004年7月，David Heinemeier Hansson从其项目 Basecamp中抽取并发布了 Ruby On Rails框架。三年后的2007年12月7日，具有划时代意义的 Ruby On Rails 2.0版本发布了，其中包含了一系列的新内容。

接下来的六个月，全世界1400多名开发者为 Rails贡献了1600多个patch，今天,2008年6月1日,Ruby On Rails 2.1发布了。

本次发布包含但不限于如下功能点:

- Timezones
- Dirty tracking
- Gem Dependencies
- Named scope
- UTC-based migrations
- Better caching

Ruby on Rails 2.1 - What's New

和以往一样，更新Rails很简单：

```
gem install rails
```

致谢

感谢Marcos Tapajós,如果没有他，我们到现在肯定看不到这本书；感谢Daniel Lopes给本书制作了漂亮的封面；还有Ruby on Rails Brazilian中那些直接或者间接给本书提供帮助的朋友们，您的评论和建议都弥足珍贵，正像我以往说的一样，Rails中最精华的是其充满激情、创造力和分享精神的社区；还有chinaonrails.com社区中的朋友们，正是大家的辛勤工作，才使得我们能这么短的时间内即可完成翻译，谢谢你们。

中文译者

本书正是由China on Rails社区中一些朋友翻译成中文的,我们是：

lceskYsl <http://iceskysl.1sters.com/>

第1章(Introduction)，第9章(Rake Tasks, Plugins and Scripts)，第11章(Ruby 1.9)，第14章(Additional Information).

jesse.cai <http://www.caiwangqin.com/>

第5章(ActionPack)，第12章(Debug)

suave.su <http://chinaonrails.com/u/suave>

第1章(Introduction)

dongbin <http://dongbin.org/>

第3章(ActiveSupport)

海阳 <http://rubyjin.cn/>

第6章(ActionController)

404 <http://chinaonrails.com/u/404>

第8章(Railties)

ashchan <http://blog.ashchan.com/>

第4章(ActiveResource),第10章(Prototype and script.aculo.us)

cash.zhao <http://www.cashplk.com/>

第7章(ActionView),第13章(Bugs and Fixes)

snow zhang <http://blog.snowonrails.com>

第2章(ActiveRecord)

Libin Pan <http://blog.libinpan.com>

Markdown Editor

Chapter 2

ActiveRecord

ActiveRecord是一个对象-关系映射层，主要负责应用层和数据层之间的互操作性(解耦)以及数据抽象。(wikipedia)

SUM方法

sum方法中的表达式

现在我们可以ActiveRecord方法当中使用表达式来处理诸如sum等各种计算，如：

```
Person.sum("2 * age")
```

sum方法默认返回值的改变

在之前的版本中，当我们使用ActiveReocrd的sum方法来计算表中所有记录的和的时候，如果没有跟所需条件匹配的记录的时候，则默认的返回值是nil。在 Rails2.1 中，默认返回值(当没有匹配的记录的时候)是0。如：

```
Account.sum(:balance, :conditions => '1 = 2') #=> 0
```

HAS_ONE

支持 through 选项

has_one方法现在支持through选项。他的用法与has_many:through相同，不过代表的是和单一一个ActiveRecord对象的关联。

```
class Magazine < ActiveRecord::Base
  has_many :subscriptions
end

class Subscription < ActiveRecord::Base
  belongs_to :magazine
  belongs_to :user
end

class User < ActiveRecord::Base
  has_many :subscriptions
  has_one :magazine, :through => :subscriptions,
    :conditions => ['subscriptions.active = ?', true]
end
```

Ruby on Rails 2.1 - What's New

Has_one :source_type 选项

上边提到的has_one :through方法还能接收一个:source_type选项，我会试着通过一些例子来解释。我们先来看看这个类：

```
class Client < ActiveRecord::Base
  has_many :contact_cards

  has_many :contacts, :through => :contact_cards
end
```

上边的代码是一个Client类，has_many种联系人(contacts)，由于ContactCard类具有多态性，下一步，我们创建2个类来代表ContractCard：

```
class Person < ActiveRecord::Base
  has_many :contact_cards, :as => :contact
end

class Business < ActiveRecord::Base
  has_many :contact_cards, :as => :contact
end
```

Person和Business通过ContactCard表与Client类关联，换句话说，我有两类联系人，私人的(personal)和工作上的(business)。然而，这样做却行不通，看看当我试图获取一个contact时候发生了什么：

```
>> Client.find(:first).contacts
# ArgumentError: /.../active_support/core_ext/hash/keys.rb:48:
# in `assert_valid_keys': Unknown key(s): polymorphic
```

为了让上述代码成功，我们需要使用:source_type。我们更改一下Client类：

```

class Client < ActiveRecord::Base
  has_many :people_contacts,
           :through => :contact_cards,
           :source => :contacts,
           :source_type => :person

  has_many :business_contacts,
           :through => :contact_cards,
           :source => :contacts,
           :source_type => :business
end

```

注意到现在我们有两种获得联系人的方式，我们可以选择我们期待哪种:source_type。

```

Client.find(:first).people_contacts
Client.find(:first).business_contacts

```

NAMED_SCOPE

has_finder gem已经添加到Rails当中了，有一个新名字:named_scope。

为了全面了解一下这为Rails带来了什么，我们看看下边的例子：

```

class Article < ActiveRecord::Base
  named_scope :published, :conditions => {:published => true}
  named_scope :containing_the_letter_a, :conditions => "body LIKE '%a%'"
end

Article.published.paginate(:page => 1)
Article.published.containing_the_letter_a.count

```

Ruby on Rails 2.1 - What's New

```
Article.containing_the_letter_a.find(:first)
Article.containing_the_letter_a.find(:all, :conditions => {...})
```

通常会创建一个新的叫做published的方法来返回所有的已经发布的帖子，不过在这里我是用了named_scope来做同样的事情，而且还能得到其他的效果。看看另一个例子：

```
named_scope :written_before, lambda { |time|
  { :conditions => ['written_on < ?', time] }
}

named_scope :anonymous_extension do
  def one
    1
  end
end

named_scope :named_extension, :extend => NamedExtension

named_scope :multiple_extensions,
  :extend => [MultipleExtensionTwo, MultipleExtensionOne]
```

用PROXY_OPTIONS来测试NAMED_SCOPE

Named scopes是Rails 2.1中很有趣的新功能，不过使用一段时间以后你就会发现想创建一些复杂情况的测试会有点麻烦，看个例子：

```
class Shirt < ActiveRecord::Base
  named_scope :colored, lambda { |color|
    { :conditions => { :color => color } }
  }
end
```

如何创建一个能测试scope结果的测试呢？

为了解决这个问题，proxy_options被创建了出来。它允许我们来检测named_scope使用的选项。为了测试上述代码，我们可以这样写测试：

```
class ShirtTest < Test::Unit
  def test_colored_scope
    red_scope = { :conditions => { :colored => 'red' } }
    blue_scope = { :conditions => { :colored => 'blue' } }
    assert_equal red_scope, Shirt.colored('red').scope_options
    assert_equal blue_scope, Shirt.colored('blue').scope_options
  end
end
```

INCREMENT 和 DECREMENT

ActiveRecord的方法increment,increment!,decrement和decrement!现在支持一个新的可选参数。之前版本的Rails中你可以通过这些方法指定的属性值加1或减1。在Rails 2.1中，你可以指定要增加或者减少的值，像这样：

```
player1.increment!(:points, 5)
player2.decrement!(:points, 2)
```

上边的例子中，我向player1加了5分，从player2减了2分。由于这是一个可选的参数，所以之前的代码不会受到影响。

Ruby on Rails 2.1 - What's New

FIND

Conditions

从现在开始，你可以向ActiveRecord的find方法中传一个对象作为参数。看下边的例子：

```
class Account < ActiveRecord::Base
  composed_of :balance, :class_name => "Money", :mapping => %w(balance amount)
end
```

这个例子中，你能向Account类的find方法中传入一个Money实例作为参数，像这样：

```
amount = 500
currency = "USD"
Account.find(:all, :conditions => { :balance => Money.new(amount, currency) })
```

Last

到现在为止我们只能在ActiveRecord的find方法中使用三个操作符来查找数据，他们是:first,:all和对象自己的id(这种强况下，我们除了id以外不再传入其他的参数)。

在Rails 2.1当中，有了第四个操作符:last，几个例子：

```
Person.find(:last)
Person.find(:last, :conditions => [ "user_name = ?", user_name])
Person.find(:last, :order => "created_on DESC", :offset => 5)
```

为了能明白这个新的操作符如何工作，我们看看下边的测试：


```
def test_find_last
  last = Developer.find :last
  assert_equal last, Developer.find(:first, :order => 'id desc')
end
```

All

类方法all是另一个类方法find(:all)的别名。如：

`Topic.all` is the same as `Topic.find(:all)`

First

类方法first是另一个类方法find(:first)的别名。如：

`Topic.first` is the same as `Topic.find(:first)`

Last

类方法last是另一个类方法find(:last)的别名。如：

`Topic.last` is the same as `Topic.find(:last)`

在NAMED_SCOPE中使用FIRST和LAST方法

所有上述的方法同样适用于named_scope。比如我们创建一个叫recnet的named_scope，下列代码是有效的：

Ruby on Rails 2.1 - What's New

```
post.comments.recent.last
```

EAGER LOADING

为了解释这个新的功能，我们看如下代码：

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

我在查询authors这个表的记录，同时通过author_id包含进posts和comments表。这个查询原来会产生这样的sql查询语句：

```
SELECT
  authors."id"           AS t0_r0,
  authors."created_at"   AS t0_r1,
  authors."updated_at"   AS t0_r2,
  posts."id"             AS t1_r0,
  posts."author_id"      AS t1_r1,
  posts."created_at"     AS t1_r2,
  posts."updated_at"     AS t1_r3,
  comments."id"          AS t2_r0,
  comments."author_id"   AS t2_r1,
  comments."created_at"  AS t2_r2,
  comments."updated_at"  AS t2_r3
FROM
  authors
  LEFT OUTER JOIN posts ON posts.author_id = authors.id
  LEFT OUTER JOIN comments ON comments.author_id = authors.id
```

这个sql可真是够长的，在authors，posts和comments三个表之间用了joins。我们管这叫做笛卡尔乘积(cartesian product)。

这类查询往往在效率上边不高，所以在Rails 2.1中有了一些改进。同样的对于Author表的查询，现在使用了一种不同的方式从所有三个表当中取得信息。原来用了一条sql语句获得三个表记录，现在Rails用三条不同的查询语句，每个表一条，这要比之前生成的查询语句更短。新的结果可以在执行了上述代码后的log中看到：

```
SELECT * FROM "authors"
SELECT posts.* FROM "posts" WHERE (posts.author_id IN (1))
SELECT comments.* FROM "comments" WHERE (comments.author_id IN (1))
```

绝大多数情况下，三个简单的查询要比一个复杂的场查询语句执行的更快。

BELONGS_TO

为了能在关联中使用:dependent=>:destroy和:delete, belongs_to方法做了一些更改，比如：

```
belongs_to :author_address
belongs_to :author_address, :dependent => :destroy
belongs_to :author_address_extra, :dependent => :delete,
      :class_name => "AuthorAddress"
```

POLYMORPHIC URL

一些多态URL的辅助方法也被引入到新的Rails当中，用来提供一种更为简洁优雅的操作routes的方式。

这些方法在你想生成基于RESTful资源的URL，同时又不必显示指定资源的类型的时候，会显得十分有用。

使用方面，非常的简单，来看看一些例子（注释的部分是Rails 2.1之前的做法）

Ruby on Rails 2.1 - What's New

```
record = Article.find(:first)
polymorphic_url(record) #-> article_url(record)

record = Comment.find(:first)
polymorphic_url(record) #-> comment_url(record)

# it can also identify recently created elements
record = Comment.new
polymorphic_url(record) #-> comments_url()
```

注意到polymorphic_url方法是如何确认传入参数的类型并且生成正确的routes。内嵌资源 (Nested resources) 和namespaces也同样支持：

```
polymorphic_url([:admin, @article, @comment])
#-> this will return:
admin_article_comment_url(@article, @comment)
```

你同样能够使用new, edit, formatted等前缀。看看下边的例子：

```
edit_polymorphic_path(@post)
#=> /posts/1/edit

formatted_polymorphic_path([@post, :pdf])
#=> /posts/1.pdf
```

只读关联 (READONLY RELATIONSHIPS)

一个新的功能被添加到了models之间的关联当中。为了避免更改关联模型的状态，你现在可以使用:readonly来描述一个关联。我们看几个例子：

```

has_many :reports, :readonly => true

has_one :boss, :readonly => :true

belongs_to :project, :readonly => true

has_and_belongs_to_many :categories, :readonly => true

```

这样，关联的models就能够避免在model中被更改，如果试图更改他们，那么将得到一个ActiveRecord::ReadOnlyRecord异常

ADD_TIMESTAMPS和REMOVE_TIMESTAMPS方法

现在我们有新的方法add_timestamps和remove_timestamps，他们分别添加，删除timestamp列。看个例子：

```

def self.up
  add_timestamps :feeds
  add_timestamps :urls
end

def self.down
  remove_timestamps :urls
  remove_timestamps :feeds
end

```

CALCULATIONS

ActiveRecord::Calculations做了一些更改以支持数据库表名。这个功能在几个不同表之间存在关联且相关列名相同时会非常有用。我们有两个选项可选：

Ruby on Rails 2.1 - What's New

```
authors.categories.maximum(:id)
authors.categories.maximum("categories.id")
```

ACTIVERECORD::BASE.CREATE接受BLOCKS

我们已经习惯了ActiveRecord::Base.new接受block作为参数了，现在create也同样接受blocks了：

```
# Creating an object and passing it a block describing its attributes
User.create(:first_name => 'Jamie') do |u|
  u.is_admin = false
end
```

我们也能用同样的方法一次创建多个对象：

```
# Creating an array of new objects using a block.
# The block is executed once for each of object that is created.
User.create([{:first_name => 'Jamie'}, {:first_name => 'Jeremy'}]) do |u|
  u.is_admin = false
end
```

同样在关联当中可以使用：

```
author.posts.create!(:title => "New on Edge") { |p| p.body = "More cool stuff!"}

# or

author.posts.create!(:title => "New on Edge") do |p|
  p.body = "More cool stuff!"
end
```

CHANGE_TABLE

在Rails 2.0当中，创建的migrations要比之前版本更为性感，不过要想用migrations修改一个表可就不那么性感了。

在Rails 2.1中，修改表也由于新方法change_table而变得同样性感了。我们来看个例子：

```
change_table :videos do |t|
  t.timestamps # this adds columns created_at and updated_at
  t.belongs_to :goat # this adds column goat_id (integer)
  t.string :name, :email, :limit => 20 # this adds columns name and email
  t.remove :name, :email # this removes columns name and email
end
```

新方法change_table的使用就和他的表兄create_table一样，只不过不是创建一个新表，而是通过添加或者删除列或索引来更改现有的表。

```
change_table :table do |t|
  t.column # adds an ordinary column. Ex: t.column(:name, :string)
  t.index # adds a new index.
  t.timestamps
  t.change # changes the column definition. Ex: t.change(:name, :string, :limit => 80)
  t.change_default # changes the column default value.
  t.rename # changes the name of the column.
  t.references
  t.belongs_to
  t.string
  t.text
  t.integer
  t.float
  t.decimal
  t.datetime
```

Ruby on Rails 2.1 - What's New

```
t.timestamp  
t.time  
t.date  
t.binary  
t.boolean  
t.remove  
t.remove_references  
t.remove_belongs_to  
t.remove_index  
t.remove_timestamps  
end
```

DIRTY OBJECTS

在新Rails当中，我们同样可以跟踪对ActiveRecord所做的更改。我们能够知道是否一个对象被进行了修改，如果有更改，那么我们就跟踪到最新的更改。我们来看几个例子：

```
article = Article.find(:first)
```

```
article.changed? #=> false
```

```
article.title #=> "Title"
```

```
article.title = "New Title"
```

```
article.title_changed? #=> true
```

```
# shows title before change
```

```
article.title_was #=> "Title"
```

```
# before and after the change
```

```
article.title_change #=> ["Title", "New Title"]
```


可以看到，使用上边非常的简单，同时你也能够通过下列两种方法的任意一种列出对一个对象的所有更改：

```
# returns a list with all of the attributes that were changed
article.changed #=> ['title']

# returns a hash with attributes that were changed
# along with its values before and after
article.changes #=> { 'title' => ["Title", "New Title"] }
```

注意到当一个对象被保存后，他的状态也随之改变：

```
article.changed? #=> true
article.save #=> true
article.changed? #=> false
```

如果你不通过attr=来更改一个对象的状态，那么你需要显示的调用attr_name_will_change!方法(用对象的实际属性名称替换attr)来通知属性已经被更改。我们再看最后一个例子：

```
article = Article.find(:first)
article.title_will_change!
article.title.upcase!
article.title_change #=> ['Title', 'TITLE']
```

PARTIAL UPDATES

Dirty Objects的实现让另一个非常有趣的功能变为可能。

由于我们现在可以跟踪一个对象的状态是否发生改变，那么为什么不用它来避免那些不必要的对数据库的更新呢？

Ruby on Rails 2.1 - What's New

在之前版本的Rails当中，当我们对一个已经存在的ActiveRecord对象调用save方法的时候，所有数据库中的字段都会被更新，即使那些没有做任何更改的字段。

这种方式在使用了Dirty Objects以后应该会有很大的改进，而实际情况也的确如此。看看在保存一个有一点更改的对象时，Rails 2.1生成的SQL查询语句：

```
article = Article.find(:first)
article.title #=> "Title"
article.subject #=> "Edge Rails"

# Let's change the title
article.title = "New Title"

# it creates the following SQL
article.save
#=> "UPDATE articles SET title = 'New Title' WHERE id = 1"
```

注意到，只有那些在应用中被更改的属性才在被更新。如果没有属性被更改，那么ActiveRecord就不执行任何更新语句。

为了开启/关闭这个新功能，你要更改model的partial_updates属性。

```
# To enable it
MyClass.partial_updates = true
```

如果希望对所有的models开启/关闭这个功能，那么你必须编辑config/initializers/new_rails_defaults.rb：

```
# Enable it to all models
ActiveRecord::Base.partial_updates = true
```

别忘了如果你不通过attr=更改字段，同样要通过config/initializers/new_rails_defaults.rb来通知Rails，像这样：

```

# If you use **attr=**,
# then it's ok not informing
person.name = 'bobby'
person.name_change # => ['bob', 'bobby']

# But you must inform that the field will be changed
# if you plan not to use **attr=**
person.name_will_change!
person.name << 'by'
person.name_change # => ['bob', 'bobby']

```

如果你不通知Rails，那么上述的代码同样会更改对象的属性，但是却不能被跟踪到，从而也就无法正确的更新数据库中的相应字段。

MYSQL中使用SMALLINT, INT还是BIGINT ?

现在在创建或者更改整型列的时候ActiveRecord的MySQL适配器会处理的更为聪明，它可以根据:limit属性确定一个字段的类型应该是smallint，int还是bigint。我们来看个实现上述功能的例子：

```
case limit when 1; 'tinyint' when 2; 'smallint' when 3; 'mediumint' when 4, nil; 'int(11)' else; 'bigint' end
```

现在我们在migration中使用它，看看每一个字段应该匹配什么类型：

```

create_table :table_name, :force => true do |t|

  # 2: smallint
  t.integer :column_one, :limit => 2 # smallint(2)

  # 4: int(11)

```

Ruby on Rails 2.1 - What's New

```
t.integer :column_two, :limit => 4 # int(11)

# 5 - : bigint
t.integer :column_three, :limit => 15 # bigint(15)

# if :limit is not informed: int(11)
t.integer :column_four # int(11)
end
```

PostgreSQL适配器已经有这个功能了，现在MySQL也不甘落后了。

HAS_ONE和BELONGS_TO中的:SELECT选项

已经为人熟知的has_one和belongs_to方法现在接收一个新属性:select。

它的默认值是“”(??"SELECT FROM table")，不过你可以更改默认值来获得任何你希望的列。

别忘了包括进主键和外键，否则你会得到一个错误。

belongs_to方法不再支持:order选项了，不过不要担心，因为那基本上没什么用处。

使用单表继承(STI)的时候存储类的全名

当我们的models有namespace，并且是单表继承(STI)的时候，ActiveRecord仅仅将类名，而不是包括namespace(demodulized)在内的全名存起来。这种情况仅仅当单表继承的所有类在一个namespace的时候有效，看个例子：

```

class CollectionItem < ActiveRecord::Base; end
class ComicCollection::Item < CollectionItem; end

item = ComicCollection::Item.new
item.type # => 'Item'

item2 = CollectionItem.find(item.id)
# returns an error, because it can't find
# the class Item

```

新的Rails添加了一个属性，从而使ActiveRecord能存储类的全名。可以在environment.rb当中添加如下代码来启动/关闭这个功能：

```
ActiveRecord::Base.store_full_sti_class = true
```

默认值是true。

TABLE_EXISTS?方法

AbstractAdapter类有个新方法table_exists，用法非常简单：

```

>> ActiveRecord::Base.connection.table_exists?("users")
=> true

```

基于时间戳的MIGRATIONS (TIMESTAMPED MIGRATIONS)

当你一个人使用Rails开发的时候，migrations似乎是所有问题的最好解决方案。不过，当和团队的其他成员共同开发一个项目的时候，你就会发现(如果你尚未发现)处理migrations同步的问题非常棘手。Rails 2.1中基于时间戳的migrations解决方案很好的解决了这个问题。

在基于时间戳的migrations引入之前，创建每一个migration都会在其名字之前生成一个数字，如果两个migrations分别由两个开发者生成，并且都没有即时的提交到版本库中，那么最后就有可能存在相同前缀数字，但是不同内容的migrations，这时你的schema_info表就会过期，同时在版本控制系统中出现冲突。

试图解决这个问题的尝试有很多，人们创建了很多plugins以不同的方式解决这个问题。尽管有一些plugins可用，不过一点是非常清楚的，旧的方式不能满足我们的要求了。

如果你使用Git，那么你可能在给自己挖一个更大的陷阱，因为你的团队可能同时有几个working branches，过期了的migrations则在每一个branch中都存在。这样当合并这些branches的时候就会有严重的冲突问题。

为了解决这个大问题，Rails核心团队已经改变了migrations的工作方式。他们废弃了原有的以当前schema_info中version列的值作为migration前缀的依据的方法，取而代之的是基于UTC时间，按照YYYYMMDDHHMMSS格式的字符串表达方式作为前缀。

同时创建了一个新的叫schema_migrations的表，表中存着哪些migrations已经被执行了，这样如果发现有人创建了一个有较小值的migration，rails会回滚migrations到之前的那个版本，然后重新执行所有的migration直到当前的版本。

显然，这样做解决了migrations带来的冲突问题。

有两个新的和migrations相关的rake命令：

```
rake db:migrate:up  
rake db:migrate:down
```

Chapter 3

ActiveSupport

Active Support 是一个非常有用的类集合，以及对 Ruby on Rails 应用非常有用的核心库扩展集合。(wikipedia)

ACTIVESUPPORT::COREEXTENSIONS::DATE::CALCULATIONS

Time#end_of_day

返回当日结束时间 11:59:59 PM.

Time#end_of_week

返回周末时间 (Sunday 11:59:59 PM).

`Time#end_of_quarter`

返回一个 Date 对象，代表本季度最后一天。换句话说，它根据当前日期返回三月，六月，九月或者十二月的最后一天。

`Time#end_of_year`

返回十二月31日 11:59:59 PM

`Time#in_time_zone`

本方法类似于 `Time#localtime`，除了它使用 `Time.zone` 而不是底层操作系统的时区。你可以传入 `TimeZone` 或者 `String` 作为参数。看下面的例子：

```
Time.zone = 'Hawaii'
Time.utc(2000).in_time_zone
# => Fri, 31 Dec 1999 14:00:00 HST -10:00

Time.utc(2000).in_time_zone('Alaska')
# => Fri, 31 Dec 1999 15:00:00 AKST -09:00
```

`Time#days_in_month`

方法 `days_in_month` 中的一个 bug 被修正了，当没有指定年的时候，对于2月它返回错误的天数。

这个改变使得在没有指定年的情况下，当前年在方法调用的作为默认值。假设你处于闰年，看下面的例子：

Ruby on Rails 2.1 - What's New

```
Loading development environment (Rails 2.0.2)
>> Time.days_in_month(2)
=> 28

Loading development environment (Rails 2.1.0)
>> Time.days_in_month(2)
=> 29
```

DateTime#to_f

DateTime 类得到了一个新的方法名为 to_f，它把日期以浮点数的形式返回。这个浮点数代表从 Unix 纪元 (1970,1月1日午夜)开始的秒数。

Date.current

Date 类得到了一个新方法名为 current 来代替 Date.today, 因为它考虑到 config.time_zone 中设置的时区，如果它设置了，返回 Time.zone.today. 如果没有设置，它返回Date.today。

FRAGMENT_EXIST?

两个新方法被加入到 cache_store 中: fragment_exist? 和 exist?。

方法 fragment_exist? 顾名思义, 它检验一个 key 所指定的缓存片段是否存在。基本上代替了著名的：

```
read_fragment(path).nil?
```

exist? 方法被加入到 cache_store, 而 fragment_exist? 是一个你能够在 Controller 中使用的 helper。

UTC OR GMT?

这是一个修正，但是很有趣。迄今为止，Rails 使用 UTC 缩写很频繁，但是当 TimeZone 的 to_s 方法被调用的时候，它打印 GMT，而不是 UTC。这是因为 GMT 缩写 在对于最终用户最熟悉。

如果你观察 Windows 控制面板，其中你可以选择时区，你会注意到缩写是 GMT。Google 和 Yahoo 也在他们的产品中使用 GMT。

```
TimeZone['Moscow'].to_s #=> "(GMT+03:00) Moscow"
```

JSON ESCAPE

json_escape 方法行为类似 html_escape。在我们想要在 HTML 页面中显示 JSON 字符串的时候非常有用。例如，在一个文档处理中：

```
puts json_escape("is a > 0 & a < 10?")
# => is a \u003E 0 \u0026 a \u003C 10?
```

我们也能使用简写 j 在 ERB 中：

```
<%= j @person.to_json %>
```

如果你想所有的 JSON 代码默认都被 'escaped'，在你的 environment.rb 文件中包含下面的代码：

```
ActiveSupport.escape_html_entities_in_json = true
```

MEM_CACHE_STORE NOW ACCEPTS OPTIONS

Memcache-Client 被包含在 ActiveSupport::Cache 中使得事情变得比以前更容易了,但是它也剥夺了灵活性,它除了 memcached 服务器的 IP 之外什么都不允许我们配置。

Jonathan Weiss 提交给 Rails 一个补丁, 允许额外的选项比如:

```
ActiveSupport::Cache.lookup_store :mem_cache_store, "localhost"

ActiveSupport::Cache.lookup_store :mem_cache_store, "localhost", '192.168.1.1',
  :namespace => 'foo'
```

或者

```
config.action_controller.fragment_cache_store = :mem_cache_store, 'localhost',
  {:compression => true, :debug => true, :namespace => 'foo'}
```

TIME.CURRENT

Time 类中的新方法。current 方法返回值依赖 config.time_zone, 如果之前指定过, 方法返回 Time.zone.now, 否则返回 Time.now。

```
# return value depends on config.time_zone
Time.current
```

since 和 ago 方法也改变了他们的返回值, 如果 config.time_zone 已经指定, 它返回一个 TimeWithZone。

它让 Time.current 方法作为新的默认方法来获取当前时间。代替了 Time.now (这个方法仍然存在, 但是它不考虑指定的时区)。

`datetime_select`方法，`select_datetime` 和 `select_time` 也已被更新以默认返回 `Time.current`。

REMOVING WHITESPACES WITH SQUISH METHOD

两个新方法被加入到 `String` 对象中，`squish` 和 `squish!`。

这两个方法和 `strip` 方法一样。它删除文本前后的空格，它也删除文本中间无用的空格。看这个例子：

```
“ A text full of spaces “.strip
#=> “A text full of spaces”

“ A text full of spaces “.squish
#=> “A text full of spaces”
```

Chapter 4

ActiveResource

ActiveResource是RESTful系统中的客户端实现。使用类似代理或远程服务的对象，可以调用RESTful服务。

使用EMAIL作为用户名

某些服务使用Email作为用户名，这会要求使用如下形式的URL：

```
http://ernesto.jimenez@negonation.com:pass@tractis.com
```

这个URL中有两个"@”，这会带来问题：解释器无法正确解析这个URL。为此，对 ActiveResource 的使用方式作了扩展，以方便使用Email进行身份验证。可以这样来使用：

```
class Person < ActiveResource::Base
  self.site = "http://tractis.com"
  self.user = "ernesto.jimenez@negonation.com"
```

```

    self.password = "pass"
  end

```

CLONE 方法

现在我们可以复制已有的resource：

```

ryan = Person.find(1)
not_ryan = ryan.clone
not_ryan.new? # => true

```

要注意复制出来的对象并不复制任何类属性，而是仅复制resource属性。

```

ryan = Person.find(1)
ryan.address = StreetAddress.find(1, :person_id => ryan.id)
ryan.hash = {:not => "an ARes instance"}

not_ryan = ryan.clone
not_ryan.new?           # => true
not_ryan.address        # => NoMethodError
not_ryan.hash           # => {:not => "an ARes instance"}

```

超时

由于ActiveResource使用 HTTP 来访问RESTful API，当服务器响应缓慢或者服务器不工作的时候会出问题。在某些情况下，调用ActiveResource会超时失效。现在可以使用timeout属性来设置失效时间了。

```

class Person < ActiveRecord::Base
  self.site = "http://api.people.com:3000/"
end

```

Ruby on Rails 2.1 - What's New

```
self.timeout = 5 # waits 5 seconds before expire  
end
```

本例中将超时设置为5秒钟。推荐的作法是将该值设得小一些以使系统能快速检测到失败，以避免相关错误引发服务器出错。

ActiveResource内部使用Net::HTTP库来发起HTTP请求。当对timeout属性设值时，该值同时被设置到所使用的Net::HTTP对象实例的 read_timeout 属性上。

该属性的默认值为60秒。

Chapter 5

ActionPack

包含 ActionView (为最终用户生成显示, 象 HTML,XML,JavaScript) 和 ActionController (业务流程控制)。
(adapted from wikipedia)

TIMEZONE

定义一个默认的时区

一个新的选项被加入到 time_zone_select 方法, 在你的用户没有选择任何 TimeZone 的时候, 或当数据库字段为空时, 你现在可以显示一个默认值。它已经创建了一个 :default 选项, 你可以按照下面的方式使用这个方法:

```
time_zone_select("user", "time_zone", nil, :include_blank => true)

time_zone_select("user", "time_zone", nil,
```

Ruby on Rails 2.1 - What's New

```
:default => "Pacific Time (US & Canada)" )

time_zone_select( "user", 'time_zone', TimeZone.us_zones,
  :default => "Pacific Time (US & Canada)")
```

如果我们使用 :default 选项，它必须显示哪一个 TimeZone 已经被选择。

formatted_offset 方法

formatted_offset 方法被包含在 Time 和 DateTime 类中，返回 +HH : MM 格式的 UTC 时差。例如，在我们的时区（北京时间），这个方法返回的时差是一个字符串 "+08:00"。

让我们看看一些示例：

从一个 DateTime 得到时差：

```
datetime = DateTime.civil(2000, 1, 1, 0, 0, 0, Rational(-6, 24))
datetime.formatted_offset      # => "-06:00"
datetime.formatted_offset(false) # => "-0600"
```

从 Time：

```
Time.local(2000).formatted_offset      # => "-06:00"
Time.local(2000).formatted_offset(false) # => "-0600"
```

注意这个方法返回字符串，可以格式化或不依赖于一个被给予的参数。

with_env_tz 方法

with_env_tz 方法允许我们以非常简单的方式测试不同的时区：

```
def test_local_offset
  with_env_tz 'US/Eastern' do
    assert_equal Rational(-5, 24), DateTime.local_offset
  end
  with_env_tz 'US/Central' do
    assert_equal Rational(-6, 24), DateTime.local_offset
  end
end
```

这个 Helper 可以调用 with_timezone, 但为了避免使用 ENV['TZ'] 和 Time.zone 时混乱，它被重命名为 with_env_tz.

Time.zone_reset!

该方法已经被删除

Time#in_current_time_zone

该方法修改为当 Time.zone 为空时返回 self

Time#change_time_zone_to_current

该方法修改为当 Time.zone 为空时返回 self

Ruby on Rails 2.1 - What's New

TimeZone#now

该方法修改为返回 ActiveSupport::TimeWithZone 显示当前在 TimeZone#now 中已设定的时区。例如：

```
Time.zone = 'Hawaii' # => "Hawaii"  
Time.zone.now        # => Wed, 23 Jan 2008 20:24:27 HST -10:00
```

Compare_with_coercion

在 Time 和 DateTime 类中新增加了一个方法 compare_with_coercion(和一个别名 <=>), 它使在 Time, DateTime 类和 ActiveSupport::TimeWithZone 实例之间可以按年代顺序排列的比较。为了更好的理解, 请看下面的示例 (行尾注释显示器结果) :

```
Time.utc(2000) <=> Time.utc(1999, 12, 31, 23, 59, 59, 999) # 1  
Time.utc(2000) <=> Time.utc(2000, 1, 1, 0, 0, 0) # 0  
Time.utc(2000) <=> Time.utc(2000, 1, 1, 0, 0, 0, 001) # -1  
  
Time.utc(2000) <=> DateTime.civil(1999, 12, 31, 23, 59, 59) # 1  
Time.utc(2000) <=> DateTime.civil(2000, 1, 1, 0, 0, 0) # 0  
Time.utc(2000) <=> DateTime.civil(2000, 1, 1, 0, 0, 1) # -1  
  
Time.utc(2000) <=> ActiveSupport::TimeWithZone.new(Time.utc(1999, 12, 31, 23, 59, 59) )  
Time.utc(2000) <=> ActiveSupport::TimeWithZone.new(Time.utc(2000, 1, 1, 0, 0, 0) )  
Time.utc(2000) <=> ActiveSupport::TimeWithZone.new(Time.utc(2000, 1, 1, 0, 0, 1) )
```

TimeWithZone#between?

在 TimeWithZone 类中包含 between? 方法检验一个实例被创建在两个日期之间。

```
@twz.between?(Time.utc(1999,12,31,23,59,59),
               Time.utc(2000,1,1,0,0,1))
```

TimeZone#parse

这个方法从字符串创建一个 ActiveSupport::TimeWithZone 实例。例如：

```
Time.zone = "Hawaii"
# => "Hawaii"
Time.zone.parse('1999-12-31 14:00:00')
# => Fri, 31 Dec 1999 14:00:00 HST -10:00
```

```
Time.zone.now
# => Fri, 31 Dec 1999 14:00:00 HST -10:00
Time.zone.parse('22:30:00')
# => Fri, 31 Dec 1999 22:30:00 HST -10:00
```

TimeZone#at

这个方法可以从一个 Unix epoch 数字创建一个 ActiveSupport::TimeWithZone 实例，例如：

```
Time.zone = "Hawaii" # => "Hawaii"
Time.utc(2000).to_f # => 946684800.0

Time.zone.at(946684800.0)
# => Fri, 31 Dec 1999 14:00:00 HST -10:00
```

Ruby on Rails 2.1 - What's New

其他方法

`to_a`, `to_f`, `to_i`, `httpdate`, `rfc2822`, `to_yaml`, `to_datetime` 和 `eq?` 被加入 `TimeWithZone` 类中。更多关于这些方法的信息请查阅相当的 Rails 文档。

`TimeWithZone` 为 Ruby 1.9 准备

在 Ruby 1.9 中，我们有了一些新的方法在 `Time` 类中，如：

```
Time.now
# => Thu Nov 03 18:58:25 CET 2005
```

```
Time.now.sunday?
# => false
```

一周中的每一天都有对应的方法。

另一个新的是 `Time` 的 `to_s` 方法将会有有一个不同的返回值。现在当我们执行 `Time.new.to_s`，将得到：

```
Time.new.to_s
# => "Thu Oct 12 10:39:27 +0200 2006"
```

在 Ruby 1.9 中我们将得到：

```
Time.new.to_s
# => "2006-10-12 10:39:24 +0200"
```

Rails 2.1 拥有哪些相关的东西？答案是所有东西，从 Rails 开始准备这些修改。`TimeWithZone` 类，例如，刚收到一个为第一个示例工作的方法实现。

AUTO LINK

为那些不知道这个方法的人，`auto_link` 方法接收所有文本参数，如果这个文本包含一个 e-mail 地址或一个网址，它将返回相同的文本，但有了超链接。

例如：

```
auto_link("Go to this website now: http://www.rubyonrails.com")  
# => Go to this website now: http://www.rubyonrails.com
```

一些网站，象 Amazon, 使用 "=" 号在 URL 中，该方法不认可这个字符，看这个方法怎样处理这种情况：

```
auto_link("http://www.amazon.com/Testing/ref=pd_bbs_sr_1")  
# => http://www.amazon.com/Testing/ref
```

注意该方法截断链接地址在 "=" 号前，因为它不支持这个符号。我产意思是，它通常是不被支持的，在 Rails 2.1 中，我们解决了这个问题。

同样的方法将在稍后更新，允许在 URL's 中使用括号。

一个使用括号的示例：

```
http://en.wikipedia.org/wiki/Sprite_(computer_graphics)
```

LABELS

当使用 `scaffold` 生成一个新表单时，它将创建下面的代码：

Ruby on Rails 2.1 - What's New

```
<% form_for(@post) do |f| %>
  <p>
    <%= f.label :title %><br />
    <%= f.text_field :title %>
  </p>
  <p>
    <%= f.label :body %><br />
    <%= f.text_area :body %>
  </p>
  <p>
    <%= f.submit "Update" %>
  </p>
<% end %>
```

这种方式更有意义，它包含了 label 方法。该方法在HTML标签中返回一个标题列。

```
>> f.label :title
=> <label for="post_title">Title</label>

>> f.label :title, "A short title"
=> <label for="post_title">A short title</label>

>> label :title, "A short title", :class => "title_label"
=> <label for="post_title" class="title_label">A short title</label>
```

你在标签中注意了 for 参数吗？"post_title" 是包含了我们的 post tile 的文本框。这个标签实际是一个关联到 *post_title** 对象。当有人点击这个 label (非链接) 时，被关联的控件接收到焦点。

Robby Russell 在他的Blog中写了一个有趣的关于这个主题的文章。你可以从这里阅读：
<http://www.robbyonrails.com/articles/2007/12/02/that-checkbox-needs-a-label>

在 FormTagHelper 中同样也拥有 label_tag 方法。该方法的工作方式和 label 一样，但更简单：


```
>> label_tag 'name'
=> <label for="name">Name</label>

>> label_tag 'name', 'Your name'
=> <label for="name">Your name</label>

>> label_tag 'name', nil, :class => 'small_label'
=> <label for="name" class="small_label">Name</label>
```

该方法同样接收 :for 选项，看一个示例：

```
label(:post, :title, nil, :for => "my_for")
```

这将返回这样的结果：

```
<label for="my_for">Title</label>
```

一种使用 PARTIALS 的新方法

在 Rails 开发过程中使用 partials 避免重复代码是很正常的方式。例如：

```
<% form_for :user, :url => users_path do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Create' %>
<% end %>
```

Partial 是一个代码片断（模板）。使用 partial 的是避免不需要的代码重复。使用 partial 非常简单，你可以这样开始 :render :partial => "name". 之后，你必须创建一个与你的 partial 同名的文件，但使用一个下划线在这之前。

上面的代码是我们通常的方式，但在新的 Rails 版本中，我们使用不同的方式做相同的事，如：

Ruby on Rails 2.1 - What's New

```
<% form_for(@user) do |f| %>
  <%= render :partial => f %>
  <%= submit_tag 'Create' %>
<% end %>
```

在这个示例中我们使用了 `partial "users/_form"`, 将接收到一个名为 "form" 的被 FormBuilder 创建的变量。

以前的方式将继续工作。

ATOM FEED 中新的 NAMESPACES

你知道 `atom_feed` 方法吗？这是 Rails 2.0的一个新特性，使创建 Atom feeds 变得更容易。看一个使用方法：

在一个 `index.atom.builder` 文件中：

```
atom_feed do |feed|
  feed.title("Nome do Jogo")
  feed.updated((@posts.first.created_at))

  for post in @posts
    feed.entry(post) do |entry|
      entry.title(post.title)
      entry.content(post.body, :type => 'html')

      entry.author do |author|
        author.name("Carlos Brando")
      end
    end
  end
end
```

什么是一个 Atom feed? Atom 的名字是基于 XML 样式的 meta 数据。在互联网中它是一个发布经常更新的内容的协议，如 Blog。例如，Feeds 经常以 XML 或 Atom 的格式发布标示为 application/atom+xml 类型。

在 Rails 2.0 的第一个版本中，该方法允许 :language, :root_url 和 :url 参数，你可以从 Rails 文档中获得更多关于这些方法的信息。但基于这一个更新，我们可以包含新的命名空间在一个 Feed 的 root 元素中，例如：

```
atom_feed('xmlns:app' => 'http://www.w3.org/2007/app') do |feed|
```

将返回：

```
<feed xml:lang="en-US" xmlns="http://www.w3.org/2005/Atom"
      xmlns:app="http://www.w3.org/2007/app">
```

修改这个示例之前，我们这样使用它：

```
atom_feed({'xmlns:app' => 'http://www.w3.org/2007/app',
          'xmlns:openSearch' => 'http://a9.com/-/spec/opensearch/1.1/'}) do |feed|

  feed.title("Nome do Jogo")
  feed.updated((@posts.first.created_at))
  feed.tag!(openSearch:totalResults, 10)

  for post in @posts
    feed.entry(post) do |entry|
      entry.title(post.title)
      entry.content(post.body, :type => 'html')
      entry.tag!('app:edited', Time.now)

      entry.author do |author|
        author.name("Carlos Brando")
      end
    end
  end
end
```

Ruby on Rails 2.1 - What's New

```
end  
end
```

CACHE

现在所有的 `fragment_cache_key` 方法默认返回 'view/' 前缀命名。

所有缓存储存已经从 `ActionController::Caching::Fragments::` 删除，并替换为 `ActiveSupport::Cache::`。在这种情况下，如果你指定一个储存地址，象 `ActionController::Caching::Fragments::MemoryStore`，你需要这样写：
`ActiveSupport::Cache::MemoryStore`。

`ActionController::Base.fragment_cache_store` 已经不再使用，`ActionController::Base.cache_store` 取代了它的位置。

由于这个新的 `ActiveSupport::Cache::` 库，它使在 `ActiveRecord::Base` 中的 `cache_key` 方法容易缓存一个 `Active Records`，它这样工作：

```
>> Product.new.cache_key  
=> "products/new"  
  
>> Product.find(5).cache_key  
=> "products/5"  
  
>> Person.find(5).cache_key  
=> "people/5-20071224150000"
```

它包含了 `ActiveSupport::Gzip.decompress/compress` 使得用 `Zlib` 压缩更容易。

现在你可以在 `environment` 选项中使用 `config.cache_store` 指定一个默认的缓存地址。有价值提起的是，如果 `tmp/cache` 目录存在，默认的缓存地址是 `FileStore`，否则使用 `MemoryStore`。你可以用以下的方式配置它：

```
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"
config.cache_store = MyOwnStore.new("parameter")
```

为了把事情变得更容易，它在 `environments/production.rb` 文件中包含了以下注释，为了提醒你记得这个选项：

```
# Use a different cache store in production
# config.cache_store = :mem_cache_store
```

在字符串中应用格式化标题

以前当你在一个包含了 's 的字符串中使用 `String#titleize` 方法时有一个 bug。这个 bug 返回大写的 'S，看一个示例：

```
>> "brando's blog".titleize
=> "Brando'S Blog"
```

看相当的示例，但已经修复了这个bug：

```
>> "brando's blog".titleize
=> "Brando's Blog"
```

ACTION_NAME

现在，要知道在运行时哪一个 view 被调用，我们只需要使用 `action_name` 方法：

```
<%= action_name %>
```

返回值将和使用 `params[:action]` 一样，但更优雅。

CACHES_ACTION 支持条件

`cache_action` 方法现在支持 `:if` 选项，允许使用条件指定一个 cache 可以被缓存。例如：

```
cache_action :index, :if => Proc.new { |c| !c.request.format.json? }
```

在上面的例子中，只有当请求不是 JSON 的时候 action index 将被缓存。

在 CACHES_PAGE METHOD 方法中的条件

`cache_page` 方法现在支持 `:if` 选择，例如：

```
# The Rails 2.0 way
cache_page :index

# In Rails 2.1 you can use :if option
cache_page :index, :if => Proc.new { |c| !c.request.format.json? }
```

FLASH.NOW 现在可以在 TESTS 中工作

谁没有因为这而头痛过？这个问题在我们测试期间，无法确定信息已经存储到了 Flash 中，因为它在到你的测试代码之前就被 Rails 清除了。

在 Rails 2.1 中这个问题已经被解决。现在你可以包含下面的代码行在你的测试中：

```
assert_equal '>value_now<', flash['test_now']
```

在 VIEWS 之外访问 HELPERS

有多少次你创建了一个 helper 希望希望它在一个 controller 中使用？要做到这样，你需要包含这个 helper module 到这个 controller 中，但这使你的代码看起来不干净。

Rails 2.1 已经开发了一个方法在 Views 之外的 Helpers. 它以很简单的方式工作：

```
# To access simple_format method, for example  
ApplicationController.helpers.simple_format(text)
```

简单而干净。

JSON

Rails 现在允许 POST's 一个 JSON 内容的请求，例如，你可以象这样发送一个 POST：

```
POST /posts  
{"post": {"title": "Breaking News"}}
```

Ruby on Rails 2.1 - What's New

所有参数都将到 params 中。例如：

```
def create
  @post = Post.create params[:post]
  # ...
end
```

为了那些不知道JSON是一个XML竞争者的人，它在JavaScript数据交换中使用相当广泛，因为它呈现为这种语言。它的名字来源于：JavaScript Object Notation.

PATH NAMES

我的 Blog(<http://www.nomadojogo.com>) 读者们应该知道我的 Custom Resource Name 插件，这想它很快就要死亡了...:(

在 Rails中你已经包含了 :as 选项在 routes (一些我实现在插件中保持兼容的东西) 中，现在你将也拥有 :path_names 选项改变你 actions 的名字。

```
map.resource :schools, :as => 'escolas', :path_names => { :new => 'nova' }
```

当然，我的插件当继续对早期的 Rails 版本有用。

定义你的 ROUTES 文件地址

在 Rails 2.1你可以定义你的 routes 存在哪一个文件中，包含以下行在你的 *environment.rb* 文件中：

```
config.routes_configuration_file
```


这将有用于当你拥有两种分开的前端共享相同models时，libraries 和 plugins。

例如，getsatisfaction.com 和 api.getsatisfaction.com 共用相同的 models, 但使用不同的 controllers, helpers 和 views.getsatisfaction 拥有它自己针对 SEO 优化的 routes 文件，但 API routes 不需要任何关于 SEO 的优化。

SESSION(:ON)

或许你还不知道这个， Rails 可以关闭 sessions：

```
class ApplicationController < ActionController::Base
  session :off
end
```

注意在我的示例中，我关闭了所有 controllers 中的 session(ApplicationController)，但我也能单独关闭某一个 controller 的 Session。

但如果我只想打开一个 controller 的 session, 在 Rails 2.1中，该方法允许 :on 选项，这样做：

```
class UsersController < ApplicationController
  session :on
end
```

简单的 TESTING HELPERS

Rails 早期版本的确 helpers 是一个非常烦人的事。我早已经非常痛苦的保证 100% 覆盖，创建 tests 为一些 helpers.

Ruby on Rails 2.1 - What's New

由于 `ActionView::TestCase` 类，在 Rails 2.1 中这变得简单得多了。看个示例：

```
module PeopleHelper
  def title(text)
    content_tag(:h1, text)
  end

  def homepage_path
    people_path
  end
end
```

现在看我们在 Rails 2.1 中怎样做同样的事：

```
class PeopleHelperTest < ActionController::Routing::Routes.draw do |map|
  map.people 'people', :controller => 'people', :action => 'index'
  map.connect ':controller/:action/:id'
end

def test_title
  assert_equal "<h1>Ruby on Rails</h1>", title("Ruby on Rails")
end

def test_homepage_path
  assert_equal "/people", homepage_path
end
```

Chapter 6

ActionController

ActionController 从网络当中接受请求，并决定将请求传给或者重定向给一个 action 去处理。

一个 action 其实是 controller 里的一个 public 方法，并且通过 Rails 的路由规则可以自动被执行。

ACTIONCONTROLLER::ROUTING

Map.root

现在你可以通过别名，更加 DRY 的用 map.root。

在早期的 Rails 版本里边，你可能是像下边这样用的：

Ruby on Rails 2.1 - What's New

```
map.new_session :controller => 'sessions', :action => 'new'  
map.root :controller => 'sessions', :action => 'new'
```

现在的你可以这样用：

```
map.new_session :controller => 'sessions', :action => 'new'  
map.root :new_session
```

路由识别 (Routes recognition)

路由识别的早期实现是一个接着一个的遍历所有路由，这样做是非常耗时的。新的路由识别则更加聪明，它生成一棵路由树，并通过前缀来跳过相似的路由。这种实现将路由识别的时间降低了近2.7倍。

recognition_optimisation.rb 文件中新的方法和他们的工作细节都在注释中写得很详尽。通过直接阅读源代码的方份可以获得这些实现的更多信息。

Assert_routing

现在可以通过一个 HTTP 方法来测试路由，看下面的例子：

```
assert_routing({ :method => 'put',  
                 :path => '/product/321' },  
               { :controller => "product",  
                 :action => "update",  
                 :id => "321" })
```

Map.resources

假设你的网站全部都是用非英文写的，你想在路由当中也使用和网站同样的语言。换句话说，你想用：

```
http://www.mysite.com.br/produtos/1234/comentarios
```

而不是：

```
http://www.mysite.com.br/products/1234/reviews
```

现在已经支持了，虽然用起来并不简单，但是没有违反 Rails 的约束。

现在我们可以使用 map.resources 里的 :as 来个性化我们的路由。比如上边葡萄牙语的那个例子：

```
map.resources :products, :as => 'produtos' do |product|
  # product_reviews_path(product) ==
  # '/produtos/1234/comentarios'
  product.resources :product_reviews, :as => 'comentarios'
end
```

ACTIONCONTROLLER::CACHING::SWEEPING

在早期的 Rails 版本里，当我们在声明 sweeper 的时候，我们必须在类里边用 symbols：

```
class ListsController < ApplicationController
  caches_action :index, :show, :public, :feed
  cache_sweeper :list_sweeper,
                :only => [ :edit, :destroy, :share ]
end
```

Ruby on Rails 2.1 - What's New

现在我们可以清楚的申明一个类而不是用 symbol.比如你的 `sweeper` 藏在一个 `model` 里，这么做是必须的。虽然你可以仍然在其他情况当中使用 `symbol`,但是从今天开始，你可以这么做：

```
class ListsController < ApplicationController
  caches_action :index, :show, :public, :feed
  cache_sweeper OpenBar::Sweeper,
                :only => [ :edit, :destroy, :share ]
end
```

Chapter 7

ActionView

ActionView 是一个表现层，负责将生成的可视的界面展现给用户，可以通过修改 ERB 模板来进行更改。

ACTIONVIEW::HELPERS::FORMHELPER

fields_for form_for with index option.

#fields_for 和 form_for 方法接受 :index 选项，在 form 对象中，如果需要去掉就必须使用 :index => nil。

下面是示例代码：

```
<% fields_for "project[task_attributes][]", task do |f| %>
  <%= f.text_field :name, :index => nil %>
  <%= f.hidden_field :id, :index => nil %>
<% end %>
```

Ruby on Rails 2.1 - What's New

```
<%= f.hidden_field :should_destroy, :index => nil %>
<% end %>
```

紧随的是新的方法：

```
<% fields_for "project[task_attributes][]", task,
  :index => nil do |f| %>
  <%= f.text_field :name %>
  <%= f.hidden_field :id %>
  <%= f.hidden_field :should_destroy %>
<% end %>
```

ACTIONVIEW::HELPERS::DATEHELPER

现在，所有的与处理时间有关的模块方法 (date_select, time_select, select_datetime, etc.) 都接受 HTML 选项。请看下面使用 date_select 的例子：

```
<%= date_select 'item', 'happening', :order => [:day], :class => 'foobar'%>
```

date_helper

通过使用 Date.current，用来定义默认值的 date_helper 方法也被更新。

ACTIONVIEW::HELPERS::ASSETTAGHELPER

register_javascript_expansion

当一个被程序员定义的符号作为一个参数，赋值给 `javascript_include_tag` 这个方法。 `register_javascript_expansion` 方法用来注册一个或者多个 javascript 文件被引用。这个是在 `init.rb` 中调用你的方法，将位于文件夹 `public/javascript` 下面的 javascript 文件注册进来。让我们看看它是如何进行工作的：

```
# In the init.rb file
ActionView::Helpers::AssetTagHelper.register_javascript_expansion
  :monkey => ["head", "body", "tail"]

# In our view:
javascript_include_tag :monkey

# We are going to have:
<script type="text/javascript" src="/javascripts/head.js"></script>
<script type="text/javascript" src="/javascripts/body.js"></script>
<script type="text/javascript" src="/javascripts/tail.js"></script>
```

register_stylesheet_expansion

这个方法实际上类似于 `ActionView::Helpers::AssetTagHelper#register_javascript_expansion` 方法。不同的是它针对的是 CSS 而不是 Javascript。看下面的例子：

```
# In the init.rb file
ActionView::Helpers::AssetTagHelper.register_stylesheet_expansion
  :monkey => ["head", "body", "tail"]

# In our view:
```

Ruby on Rails 2.1 - What's New

```
stylesheet_link_tag :monkey

# We are going to have:
<link href="/stylesheets/head.css" media="screen" rel="stylesheet"
      type="text/css" />
<link href="/stylesheets/body.css" media="screen" rel="stylesheet"
      type="text/css" />
<link href="/stylesheets/tail.css" media="screen" rel="stylesheet"
      type="text/css" />
```

ACTIONVIEW::HELPERS::FORMTAGHELPER

submit_tag

一个 :confirm 选项已经被添加在 #submit_tag 方法中，同样的选项仍然可以在 link_to 方法中使用。看下面的例子：

```
submit_tag('Save changes', :confirm => "Are you sure?")
```

ACTIONVIEW::HELPERS::NUMBERHELPER

number_to_currency

number_to_currency 方法接收 :format 选项作为参数，允许我们格式化方法返回值。在之前的版本中，当我们不得不对本地的货币进行格式化时，我们需要包含在 :unit 选项前面包含一个空格，使得输出格式正确。看下面的例子：

```
# R$ is the symbol for Brazilian currency
number_to_currency(9.99, :separator => ",", :delimiter => ".", :unit => "R$")
# => "R$9,99"

number_to_currency(9.99, :format => "%u %n", :separator => ",",
  :delimiter => ".", :unit => "R$")
# => "R$ 9,99"
```

随后，我们优化成另一个 form，例如：

```
number_to_currency(9.99, :format => "%n in Brazilian reais", :separator => ",",
  :delimiter => ".", :unit => "R$")
# => "9,99 em reais"
```

当需要创建你自己的字符串格式，只需使用以下的参数：

```
%u For the currency
%n For the number
```

ACTIONVIEW::HELPERS::TEXTHELPER

excerpt

excerpt 方法是一个帮助方法，去搜索一个单词在一个段落，同时返回一个包含给定的数字参数的缩写，在词的前后，必须使用"..."。请看随后的例子：

```
excerpt('This is an example', 'an', 5)
# => "...s is an examp..."
```

Ruby on Rails 2.1 - What's New

但是这个问题很烦。如果你去计数的话，你需要看每个方法返回6个字符而不是5个。这个bug已经解决了。看下面的正确的输出方法代码：

```
excerpt('This is an example', 'an', 5)
# => "...s is an exam..."
```

`simple_format`

`simple_format` 方法基本上接受任何文本参数和用简单的方式格式化为 HTML。它接受文本并取代换行符 (`\n`) 采用 HTML 标记 "

"。同时当我们有2个换行符像这样的 (`\n\n`)，将会采用

标记将它为一个段落。

```
simple_format("Hello Mom!", :class => 'description')
# => "<p class='description'>Hello Mom!</p>"
```

HTML 属性将会添加 "

" 标记到对应段落上。

Chapter 8

Raities

CONFIG.GEM

新特性 `config.gem` 使项目在运行时加载所有必需的 gems 成为可能。在 *environment.rb* 文件里可以指定你项目依赖的 gems。如下示例：

```
config.gem "bj"

config.gem "hpricot", :version => '0.6',
                    :source => "http://code.whytheluckystiff.net"

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

要一次性安装所有的 gem 依赖，我们只需运行如下一个 Rake 任务：

Ruby on Rails 2.1 - What's New

```
# Installs all specified gems  
rake gems:install
```

你也可以在项目运行时列出正在被使用的 gems :

```
# Listing all gem dependencies  
rake gems
```

如果其中有个gem含有文件 rails/init.rb 并且你想将它存放在你的项目中，可以用：

```
# Copy the specified gem to vendor/gems/nome_do_gem-x.x.x  
rake gems:unpack GEM=gem_name
```

这将会拷贝这个 gem 到 vendor/gems/gem_name-x.x.x。若不指定 gem 的名称，Rails 将拷贝所有 gems 包到 vendor/gem 目录中。

在插件里配置 GEM (CONFIG.GEM IN PLUGINS)

新特性 config.gem 也同样适合在插件中使用。

一直到 Rails 2.0插件里边的 init.rb 文件都是按如下方式使用：

```
# init.rb of plugin open_id_authentication  
require 'yadis'  
require 'openid'  
ActionController::Base.send :include, OpenIdAuthentication
```

而在 Rails 2.1中则是这样：

```

config.gem "ruby-openid", :lib => "openid", :version => "1.1.4"
config.gem "ruby-yadis", :lib => "yadis", :version => "0.3.4"

config.after_initialize do
  ActionController::Base.send :include, OpenIdAuthentication
end

```

那么，当你运行该任务来安装所需要的 gems 时，这些 gems 将在包含之中。

GEMS:BUILD

gems:build 任务可以用来编制通过 gems:unpack 安装的所有本地 gems 扩展。以下是语法：

```

rake gems:build # For all gems
rake gems:build GEM=mygem # I'm specifying the gem

```

RAILS 服务启动时有了新信息 (NEW MESSAGE WHEN STARTING SERVER)

Rails 服务启动时做了一点改进，当加载成功后会显示 Rails 的版本号。

```

Rails 2.1 application starting on http://0.0.0.0:3000

```

RAILS 公开访问目录的路径 (RAILS.PUBLIC_PATH)

添加了快捷方式 Rails.public_path，用于获取项目 public 目录的路径。

```

Rails.public_path

```

Ruby on Rails 2.1 - What's New

RAILS 的日志记录，根目录，环境变量和缓存 (RAILS.LOGGER, RAILS.ROOT, RAILS.ENV AND RAILS.CACHE)

在 Rails 2.1 里面有新方式可以替代常量：RAILS_DEFAULT_LOGGER, RAILS_ROOT, RAILS_ENV 和 RAILS_CACHE。取而代之的是：

```
# RAILS_DEFAULT_LOGGER  
Rails.logger
```

```
# RAILS_ROOT  
Rails.root
```

```
# RAILS_ENV  
Rails.env
```

```
# RAILS_CACHE  
Rails.cache
```

RAILS 的版本 (RAILS.VERSION)

在早期的 Rails 版本中，程序运行期间我们可以使用如下方式获得 Rails 的版本号：

```
Rails::VERSION::STRING
```

Rails 2.1 里边已经改成这样：

```
Rails.version
```


获取一个插件的相关信息 (GETTING INFORMATION ABOUT A PLUGIN)

Rails 2.0 的新特性之一，或许你从未用过。我是说“大概，或许”，可能在一些比较特殊情况下会有用，举个例子，比如获知一个插件的版本号。

不妨来尝试下，我们要在 plugin 目录里面新建一个 *about.yml* 文件，写入如下一些内容：

```
author: Carlos Brando
version: 1.2.0
description: A description about the plugin
url: http://www.nomadojogo.com
```

然后我们可以使用如下方式来获取相关信息：

```
plugin = Rails::Plugin.new(plugin_directory)
plugin.about["author"] # => "Carlos Brando"
plugin.about["url"] # => "http://www.nomadojogo.com"
```

如果你能在这个新特性中找到一些好的用处并愿与我分享，也许将改变我对于它的一些看法若真有需要的话。

Chapter 9

Rake 任务，插件，脚本 (Rake Tasks, Plugins and Scripts)

TASKS

`rails:update`

从 Rails 2.1 开始，每次你运行 `rake rails:freeze:edge` 命令的时候，它将自动运行 `rails:update` 来更新配置文件 (`config`) 和 *JavaScripts* 文件。

Chapter 9: Rake 任务，插件，脚本 (Rake Tasks, Plugins and Scripts)

Database in 127.0.0.1

databases.rake 以前只操作 local 数据库，现在增加对 IP 为 127.0.0.1 的数据库的操纵。其主要用在创建 (create) 和删除(drop)任务。databases.rake 采取 refactored 避免代码重复。

冻结指定的 Rails 版本 (Freezing a specific Rails release)

在 Rails 2.1 之前，你不能在项目组指定一个需要冻结的 Rails 版本库，你只能使用版本信息做为参数；而在 Rails 2.1 后，我们可以在如下的命令中直接指定版本号：

```
rake rails:freeze:edge RELEASE=1.2.0
```

时区 (TIMEZONE)

rake time:zones:all

按照 offset 分组返回 Rails 支持的所有时区，你可以使用 OFFSET 参数来过滤返回值，例如：OFFSET=-6

rake time:zones:us

显示美国的所有时区，OFFSET 依然有效。

Ruby on Rails 2.1 - What's New

`rake time:zones:local`

返回你本地 OS 上 Rails 支持的时区信息。

SCRIPTS

`plugin`

命令 `script/plugin install` 现在可以使用 `-e/--export` 参数来导出 SVN 库，另外，增加了对 Git 库的支持。

`dbconsole`

这个脚本和 `script/console` 一样，但是其操作的是你的数据库，换句话说，它采用命令行的方式连接到你的数据库。

另外，注意代码中说到了，目前只支持 `mysql`, `postgresql` 和 `sqlite(3)`，当你在 `database.yml` 中配置了其他类型的数据库时，会提示你 “not supported for this database type” (不支持这个数据库类型)。

插件 (PLUGINS)

Gems可插件化 (Gems can now be plugins)

现在，任何包含 `rails/init.rb` 文件的 `gem` 都可以以插件的方式安装在 `vendor` 目录。

Chapter 9: Rake 任务，插件，脚本 (Rake Tasks, Plugins and Scripts)

使用插件中的生成器 (Using generators in plugins)

可以配置 Rails 使得其在除了 `vendor/plugins` 之外的地方加载插件，配置方法为在 `environment.rb` 中添加如下代码：

```
config.plugin_paths = ['lib/plugins', 'vendor/plugins']
```

Rails 2.0 中这个地方存在一个 Bug，该 Bug 是其只在 `vendor/plugins` 中寻找有生成器的插件，而其上配置的路径下的插件的生成器并不生效。在 Rails 2.1 中已经修复。

Chapter 10

Prototype 和 script.aculo.us

PROTOTYPE

Rails 2.1 使用 Prototype 1.6.0.1 版本和script.aculo.us 1.8.1 版本。

Chapter 11

Ruby 1.9

详细信息 (DETAILS)

Rails 的修改还集中体现在对 Ruby 1.9 的支持，对新版 Ruby 中的细微改变都做了相应的调整以更好地符合要求，例如把 `File.exists?` 修改为 `File.exist?`。

另外，在 Ruby 1.9 中，去掉了 Base64 模块(`base64.rb`)，因此，在 Rails 中，所有使用这个模板的都相应的修改为 `ActiveSupport::Base64`。

DATETIME 类的新方法 (NEW METHODOS FOR DATETIME CLASS)

为了保证对 Time 类的兼容性 (duck-typing)，为 DateTime 添加了三个新方法，分别为：`#utc`，`#utc?` 和 `#utc_offset`，看个例子吧：

Ruby on Rails 2.1 - What's New

```
>> date = DateTime.civil(2005, 2, 21, 10, 11, 12, Rational(-6, 24))
#=> Mon, 21 Feb 2005 10:11:12 -0600

>> date.utc
#=> Mon, 21 Feb 2005 16:11:12 +0000

>> DateTime.civil(2005, 2, 21, 10, 11, 12, Rational(-6, 24)).utc?
#=> false

>> DateTime.civil(2005, 2, 21, 10, 11, 12, 0).utc?
#=> true

>> DateTime.civil(2005, 2, 21, 10, 11, 12, Rational(-6, 24)).utc_offset
#=> -21600
```


Chapter 12

Debug

本地 RUBY-DEBUG (NATIVE RUBY-DEBUG)

Rails 2.1 重新允许在测试中使用 ruby-debug 选项。现在，从你安装了 gem 开始，就可以直接使用 debugger 方法。

Chapter 13

Bugs 和 Fixes

ADD COLUMNS IN POSTGRESQL

在 PostgreSQL 中，仍然存在一个 bug。当你对一个已存在的表进行添加 column 的迁移方法。看下面这个例子：

File: *db/migrate/002_add_cost.rb*

```
class AddCost < ActiveRecord::Migration
  def self.up
    add_column :items, :cost, :decimal, :precision => 6,
      :scale => 2
  end

  def self.down
    remove_column :items, :cost
  end
end
```

```
end
end
```

注意我们创建了一个 column cost，:precision => 6 和 :scale => 2，现在开始运行 rake:db:migrate。下面是我们数据库中的表。

| Column | Type | Modifiers |
|--------|------------------------|-----------|
| id | integer | not null |
| desc | character varying(255) | |
| price | numeric(5,2) | |
| cost | numeric | |

看着 cost 这个我们刚创建的 column。这是一个常见的数字，但是更加类似于这个 column 类似于 "price"，基于这个，更精确的描述是 numeric(6,2)。在 Rails 2.1 中，这个错误不会产生，这个 column 将会以正确的方式别创建到数据库中。

MIME TYPES

不允许定义分配过的属性 request.format 使用符号的 bug 已经被解决了。现在你可以使用下面的代码：

```
request.format = :iphone
assert_equal :iphone, request.format
```

BUG FIXES IN CHANGE_COLUMN

一个已经存在的 bug，当使用 `change_column` 方法时，使用 `:null=>true` 在 column 中，创建过使用 `:null=>false` 也已经被解决了。当你使用这个方法的时候，这个 bug 不会造成任何改变。

Chapter 14

附录 (Additional Information)

防止跨站攻击 (PROTECTING FROM CROSS SITE SCRIPTING)

在 Rails 2.0 的 *application.rb* 中，你应该留意到如下这段代码：

```
class ApplicationController < ActionController::Base
  helper :all

  protect_from_forgery
end
```

请注意上面这段代码中对 `protect_from_forgery` 的调用。

你听说过跨站 (XSS) 么？最近一段时间，XSS 日益风行，就目前而言，在大多数的网站中都存在或多或少的 XSS 缺陷；而 XSS 缺陷会被一些怀有恶意的人利用，可以修改网站内容，钓鱼，甚至通过 JS 来控制其他用户的浏览

Ruby on Rails 2.1 - What's New

器等。尽管攻击方式不同，但是其主要目的都是使得用户在不知情的情况下做了一些“邪恶”的事情。其最新的攻击手段为“cross-site request forgery”。Cross Site Request Forgery 和前面说的 XSS 原理差不多，但是其更有危害性，随着 Ajax 的日渐盛行，这类漏洞的利用空间和手段更加灵活。

(补充：CSRF在这里介绍的不是很多，我以前写了一篇介绍CSRF的文章，感兴趣的请自行查看《CSRF: 不要低估了我的危害和攻击能力》)

`protect_from_forgery` 用来确保您的系统接收到的 form 信息都来自于你系统本身，而不会是从第三方提交过来的；其实现的原理是在你的 form 中和 Ajax 请求中添加一个基于 session 的标识 (token)，控制器接收到 form 的时候检查这个 token 是否匹配来决定如何响应这个 Post 请求。

另外，值得一提的是这个方法并不保护 Get 方式的请求，不过也无所谓，Get 方式的只是取数据，而不会存数据到数据库中。

如果你想更多的了解 CSRF (Cross-Site Request Forgery)，请参考如下两个地址：

- <http://www.nomadojogo.com/2008/01/14/como-um-garoto-chamado-samy-pode-derrubar-seu-site/isc.sans.org/diary.html?storyid=1750>
- <http://www.nomadojogo.com/2008/01/14/como-um-garoto-chamado-samy-pode-derrubar-seu-site/isc.sans.org/diary.html?storyid=1750>

请切记，这个方法不能保证万无一失，就像我们平时喜欢说的那样，他并不是银弹！

小心使用 METHOD_MISSING (USED METHOD_MISSING, THEN DON'T LEAVE LOOSE ENDS)

由于 Ruby 是动态语言，这样就使得 respond_to? 非常重要，你是否经常检查某个对象是否拥有某个方法？你是否经常使用 is_a? 来检查某个对象正是我们需要的。

然而，人们常常忘记这么做，先看个例子使用 method_missing 的例子吧：

```
class Dog
  def method_missing(method, *args, &block)
    if method.to_s =~ /^bark/
      puts "woofwoof!"
    else
      super
    end
  end
end

rex = Dog.new
rex.bark #=> woofwoof!
rex.bark! #=> woofwoof!
rex.bark_and_run #=> woofwoof!
```

我认为你肯定知道 method_missing，在上面的例子中，我创建了一个 Dog 类的实例变量，然后调用三个并不存在的方法 bark，bark! 和 bark_and_run，他就会去调用 method_missing 按照我用正则表达式规定的只要方法是以 bark 开头，就输出 woofwoof!。

没有什么问题，是吧？那么请继续看我使用 respond_to? 来检查下：

```
rex.respond_to? :bark #=> false
rex.bark #=> woofwoof!
```

Ruby on Rails 2.1 - What's New

看到没有，它返回 false，也就是说，他认为该对象并没有 bark 方法！怎么办？是时候来按照我们的规则来完善 respond_to? 了。

```
class Dog
  METHOD_BARK = /^bark/

  def respond_to?(method)
    return true if method.to_s =~ METHOD_BARK
    super
  end

  def method_missing(method, *args, &block)
    if method.to_s =~ METHOD_BARK
      puts "woofwoof!"
    else
      super
    end
  end
end

rex = Dog.new
rex.respond_to?(:bark) #=> true
rex.bark #=> woofwoof!
```

OK，没问题了！这样的问题在 Rails 中普遍存在，你可以试试用 respond_to? 来检查下 find_by_name 方法就很明白了。

Ruby 的扩展性能让人称奇，但如果你不注意，会把自己搞得一头雾水。

当然，你应该猜到我要说什么了，在 Rails 2.1 中，这个问题已经修复了，不信的话，你可以使用 respond_to? 再检测下 find_by_something 试试看。

POSTGRESQL

在 Rails 2.0 中，PostgreSQL 的适配器支持的版本从 8.1 到 8.3，而在 Rail 2.1 中可以支持的 PostgreSQL 版本扩展到 7.4 到 8.3。

Chapter 15

CHANGELOG

ACTIONMAILER

- * Fixed that a return-path header would be ignored #7572 [joost]
- * Less verbose mail logging: just recipients for :info log level; the whole email for :debug only. #8000 [iaddict, Tarmo Tänav]
- * Updated TMail to version 1.2.1 [raasdnii]
- * Fixed that you don't have to call super in ActionMailer::TestCase#setup #10406 [jamesgolick]

ACTIONPACK

- * `InstanceTag#default_time_from_options` overflows to `DateTime` [Geoff Buesing]
- * Fixed that forgery protection can be used without session tracking (Peter Jones) [#139]
- * Added `session(:on)` to turn session management back on in a controller subclass if the superclass turned it off (Peter Jones) [#136]
- * Change the request forgery protection to go by `Content-Type` instead of `request.format` so that you can't bypass it by POSTing to `"#{request.uri}.xml"` [rick] * `InstanceTag#default_time_from_options` with hash args uses `Time.current` as default; respects hash settings when time falls in system local spring DST gap [Geoff Buesing]
- * `select_date` defaults to `Time.zone.today` when `config.time_zone` is set [Geoff Buesing]
- * Fixed that `TextHelper#text_field` would corrupt when raw HTML was used as the value (mchenryc, Kevin Glowacz) [#80]
- * Added `ActionController::TestCase#rescue_action_in_public!` to control whether the action under test should use the regular `rescue_action` path instead of simply raising the exception inline (great for error testing) [DHH]
- * Reduce number of instance variables being copied from controller to view. [Pratik]
- * `select_datetime` and `select_time` default to `Time.zone.now` when `config.time_zone` is set [Geoff Buesing]
- * `datetime_select` defaults to `Time.zone.now` when `config.time_zone` is set [Geoff Buesing]
- * Remove `ActionController::Base#view_controller_internals` flag. [Pratik]

Ruby on Rails 2.1 - What's New

- * Add conditional options to `caches_page` method. [Paul Horsfall]
- * Move missing template logic to `ActionView`. [Pratik]
- * Introduce `ActionView::InlineTemplate` class. [Pratik]
- * Automatically parse posted JSON content for `Mime::JSON` requests. [rick]

```
POST /posts
{"post": {"title": "Breaking News"}}

def create
  @post = Post.create params[:post]
  # ...
end
```

- * add `json_escape` ERB util to escape html entities in json strings that are output in HTML pages. [rick]
- * Provide a helper proxy to access helper methods from outside views. Closes #10839 [Josh Peek] e.g. `ApplicationController.helpers.simple_format(text)`
- * Improve documentation. [Xavier Noria, leethal, jerome]
- * Ensure `RJS redirect_to` doesn't html-escapes string argument. Closes #8546 [josh, eventualbuddha, Pratik]
- * Support `render :partial => collection` of heterogeneous elements. #11491 [Zach Dennis]
- * Avoid `remote_ip` spoofing. [Brian Candler]

- * Added support for regexp flags like ignoring case in the :requirements part of routes declarations #11421 [NeilW]
- * Fixed that ActionController::Base#read_multipart would fail if boundary was exactly 10240 bytes #10886 [ariejan]
- * Fixed HTML::Tokenizer (used in sanitize helper) didn't handle unclosed CDATA tags #10071 [esad, packagethief]
- * Improve documentation. [Radar, Jan De Poorter, chuyeow, xaviershay, danger, miloops, Xavier Noria, Sunny Ripert]
- * Fixed that FormHelper#radio_button would produce invalid ids #11298 [harlancrystal]
- * Added :confirm option to submit_tag #11415 [miloops]
- * Fixed NumberHelper#number_with_precision to properly round in a way that works equally on Mac, Windows, Linux (closes #11409, #8275, #10090, #8027) [zhangyuanyi]
- * Allow the #simple_format text_helper to take an html_options hash for each paragraph. #2448 [Francois Beausoleil, thechrisoshow]
- * Fix regression from filter refactoring where re-adding a skipped filter resulted in it being called twice. [rick]
- * Refactor filters to use Active Support callbacks. #11235 [Josh Peek]
- * Fixed that polymorphic routes would modify the input array #11363 [thomas.lee]
- * Added :format option to NumberHelper#number_to_currency to enable better localization support #11149 [lylo]

Ruby on Rails 2.1 - What's New

- * Fixed that `TextHelper#excerpt` would include one character too many #11268 [Irfy]
- * Fix more obscure nested parameter hash parsing bug. #10797 [thomas.lee]
- * Added `ActionView::Helpers::register_javascript/stylesheets_expansion` to make it easier for plugin developers to inject multiple assets. #10350 [lotswholetime]
- * Fix nested parameter hash parsing bug. #10797 [thomas.lee]
- * Allow using named routes in `ActionController::TestCase` before any request has been made. Closes #11273 [alloy]
- * Fixed that sweepers defined by `cache_sweeper` will be added regardless of the `perform_caching` setting. Instead, control whether the sweeper should be run with the `perform_caching` setting. This makes testing easier when you want to turn `perform_caching` on/off [DHH]
- * Make `MimeResponds::Responder` any work without explicit types. Closes #11140 [jaw6]
- * Better error message for type conflicts when parsing params. Closes #7962 [spicycode, matt]
- * Remove unused `ActionController::Base.template_class`. Closes #10787 [Pratik]
- * Moved template handlers related code from `ActionView::Base` to `ActionView::Template`. [Pratik]
- * Tests for `div_for` and `content_tag_for` helpers. Closes #11223 [thechrisoshow]
- * Allow file uploads in Integration Tests. Closes #11091 [RubyRedRick]
- * Refactor partial rendering into a `PartialTemplate` class. [Pratik]

- * Added that requests with JavaScript as the priority mime type in the accept header and no format extension in the parameters will be treated as though their format was :js when it comes to determining which template to render. This makes it possible for JS requests to automatically render action.js.rjs files without an explicit respond_to block [DHH]
- * Tests for distance_of_time_in_words with TimeWithZone instances. Closes #10914 [ernesto.jimenez]
- * Remove support for multivalued (e.g., '&'-delimited) cookies. [Jamis Buck]
- * Fix problem with render :partial collections, records, and locals. #11057 [lotswholetime]
- * Added support for naming concrete classes in sweeper declarations [DHH]
- * Remove ERB trim variables from trace template in case ActionView::Base.erb_trim_mode is changed in the application. #10098 [tpope, kampers]
- * Fix typo in form_helper documentation. #10650 [xaviershay, kampers]
- * Fix bug with setting Request#format= after the getter has cached the value. #10889 [cch1]
- * Correct inconsistencies in RequestForgeryProtection docs. #11032 [mislav]
- * Introduce a Template class to ActionView. #11024 [lifofifo]
- * Introduce the :index option for form_for and fields_for to simplify multi-model forms (see <http://railscasts.com/episodes/75>). #9883 [rmm5t]
- * Introduce map.resources :cards, :as => 'tarjetas' to use a custom resource name in the URL: cards_path == '/tarjetas'. #10578 [blj]

Ruby on Rails 2.1 - What's New

- * TestSession supports indifferent access. #7372 [tamc, Arsen7, mhackett, julik, jean.helou]
- * Make assert_routing aware of the HTTP method used. #8039 [mpalmer] e.g. `assert_routing({ :method => 'put', :path => '/product/321' }, { :controller => "product", :action => "update", :id => "321" })`
- * Make map.root accept a single symbol as an argument to declare an alias. #10818 [bscofield]
e.g. `map.dashboard '/dashboard', :controller=>'dashboard'`
`map.root :dashboard`
- * Handle corner case with image_tag when passed 'messed up' image names. #9018 [duncanbeever, mpalmer]
- * Add label_tag helper for generating elements. #10802 [DefV]
- * Introduce TemplateFinder to handle view paths and lookups. #10800 [Pratik Naik]
- * Performance: optimize route recognition. Large speedup for apps with many resource routes. #10835 [oleganza]
- * Make render :partial recognise form builders and use the _form partial. #10814 [djanowski]
- * Allow users to declare other namespaces when using the atom feed helpers. #10304 [david.calavera]
- * Introduce send_file :x_sendfile => true to send an X-Sendfile response header. [Jeremy Kemper]
- * Fixed ActionView::Helpers::ActiveRecordHelper::form for when protect_from_forgery is used #10739 [jeremyevans]

- * Provide nicer access to HTTP Headers. Instead of `request.env["HTTP_REFERER"]` you can now use `request.headers["Referrer"]`. [Koz]
 - * `UrlWriter` respects `relative_url_root`. #10748 [Cheah Chu Yeow]
 - * The `asset_host` block takes the controller request as an optional second argument. Example: use a single asset host for SSL requests. #10549 [Cheah Chu Yeow, Peter B, Tom Taylor]
 - * Support `render :text => nil`. #6684 [tjennings, PotatoSalad, Cheah Chu Yeow]
 - * `assert_response` failures include the exception message. #10688 [Seth Rasmussen]
 - * All fragment cache keys are now by default prefixed with the "views/" namespace [DHH]
 - * Moved the caching stores from `ActionController::Caching::Fragments::*` to `ActiveSupport::Cache::*`. If you're explicitly referring to a store, like `ActionController::Caching::Fragments::MemoryStore`, you need to update that reference with `ActiveSupport::Cache::MemoryStore` [DHH]
 - * Deprecated `ActionController::Base.fragment_cache_store` for `ActionController::Base.cache_store` [DHH]
 - * Made fragment caching in views work for `rjs` and `builder` as well #6642 [zsombor]
 - * Fixed rendering of partials with layout when done from site layout #9209 [antramm]
 - * Fix `atom_feed_helper` to comply with the atom spec. Closes #10672 [xaviershay]
- The tags created `do not` contain a date (<http://feedvalidator.org/docs/error/InvalidTAG.html>)
IDs are `not` guaranteed unique

Ruby on Rails 2.1 - What's New

A default `self` link was `not` provided, contrary to the documentation

NOTE: `This` changes tags `for` existing atom entries, but at least they validate now.

- * Correct indentation in tests. Closes #10671 [l.guidi]
- * Fix that `auto_link` looks for `=`'s in url paths (Amazon urls have them). Closes #10640 [bgreenlee]
- * Ensure that test case setup is run even if overridden. #10382 [Josh Peek]
- * Fix HTML Sanitizer to allow trailing spaces in CSS style attributes. Closes #10566 [wesley.moxam]
- * Add `:default` option to `time_zone_select`. #10590 [Matt Aimonetti]

ACTIVERECORD

- * Add `ActiveRecord::Base sti_name` that checks `ActiveRecord::Base#store_full_sti_class?` and returns either the full or demodulized name. [rick]
- * Add `first/last` methods to `associations/named_scope`. Resolved #226. [Ryan Bates]
- * Added SQL escaping for `:limit` and `:offset` #288 [Aaron Bedra, Steven Bristol, Jonathan Wiess]
- * Added `first/last` methods to `associations/named_scope`. Resolved #226. [Ryan Bates]
- * Ensure `hm:t` preloading honours reflection options. Resolves #137. [Frederick Cheung]
- * Added protection against duplicate migration names (Aslak Hellesøy) [#112]

* Base#~~instantiate_time_object~~: eliminate check for Time.zone, since we can assume this is set if time_zone_aware_attributes is set to true [Geoff Buesing]

* Time zone aware attribute methods use Time.zone.parse instead of #to_time for String arguments, so that offset information in String is respected. Resolves #105. [Scott Fleckenstein, Geoff Buesing]

* Added change_table for migrations (Jeff Dean) [#71]. Example:

```
change_table :videos do |t|
  t.timestamps                # adds created_at, updated_at
  t.belongs_to :goat          # adds goat_id integer
  t.string :name, :email, :limit => 20 # adds name and email both with a 20 char limit
  t.remove :name, :email      # removes the name and email columns
end
```

* Fixed has_many :through .create with no parameters caused a "can't dup NilClass" error (Steven Soroka) [#85]

* Added block-setting of attributes for Base.create like Base.new already has (Adam Meehan) [#39]

* Fixed that pessimistic locking you reference the quoted table name (Josh Susser) [#67]

* Fixed that change_column should be able to use :null => true on a field that formerly had false [Nate Wiger] [#26]

* Added that the MySQL adapter should map integer to either smallint, int, or bigint depending on the :limit just like PostgreSQL [DHH]

* Change validates_uniqueness_of :case_sensitive option default back to true (from [9160]). Love your database columns, don't LOWER them. [rick]

Ruby on Rails 2.1 - What's New

- * Add support for interleaving migrations by storing which migrations have run in the new `schema_migrations` table. Closes #11493 [jordi]
- * `ActiveRecord::Base#sum` defaults to 0 if no rows are returned. Closes #11550 [kamal]
- * Ensure that `respond_to?` considers dynamic finder methods. Closes #11538. [floehopper]
- * Ensure that save on parent object fails for invalid `has_one` association. Closes #10518. [Pratik]
- * Remove duplicate code from associations. [Pratik]
- * Refactor `HasManyThroughAssociation` to inherit from `HasManyAssociation`. Association callbacks and `_ids=` now work with `hm:t`. #11516 [rubyruy]
- * Ensure `HABTM#create` and `HABTM#build` do not load entire association. [Pratik]
- * Improve documentation. [Xavier Noria, Jack Danger Canty, leethal]
- * Tweak `ActiveRecord::Base#to_json` to include a root value in the returned hash: `{"post": {"title": ...}}` [rick]

```
Post.find(1).to_json # => {"title": ...}
config.active_record.include_root_in_json = true
Post.find(1).to_json # => {"post": {"title": ...}}
```
- * Add efficient `#include?` to `AssociationCollection` (for `has_many/has_many :through/habtm`). [stopdropandrew]
- * PostgreSQL: `create_` and `drop_database` support. #9042 [ez, pedz, nicksieger]
- * Ensure that `validates_uniqueness_of` works with `with_scope`. Closes #9235. [nik.wakelin, cavalle]

- * Partial updates include only unsaved attributes. Off by default; set `YourClass.partial_updates = true` to enable. [Jeremy Kemper]
- * Removing unnecessary `uses_tzinfo` helper from tests, given that `TZInfo` is now bundled [Geoff Buesing]
- * Fixed that `validates_size_of :within` works in associations #11295, #10019 [cavalle]
- * Track changes to unsaved attributes. [Jeremy Kemper]
- * Switched to UTC-timebased version numbers for migrations and the schema. This will as good as eliminate the problem of multiple migrations getting the same version assigned in different branches. Also added `rake db:migrate:up/down` to apply individual migrations that may need to be run when you merge branches #11458 [jbarnette]
- * Fixed that `has_many :through` would ignore the hash conditions #11447 [milooops]
- * Fix issue where the `:uniq` option of a `has_many :through` association is ignored when `find(:all)` is called. Closes #9407 [cavalle]
- * Fix duplicate table alias error when including an association with a `has_many :through` association on the same join table. Closes #7310 [cavalle]
- * More efficient association preloading code that compacts a `through_records` array in a central location. Closes #11427 [danger]
- * Improve documentation. [Radar, Jan De Poorter, chuyeow, xaviershay, danger, milooops, Xavier Noria, Sunny Ripert]

Ruby on Rails 2.1 - What's New

* Fixed that ActiveRecord#Base.find_or_create/initialize would not honor attr_protected/accessible when used with a hash #11422 [miloops]

* Added ActiveRecord#Base.all/first/last as aliases for find(:all/:first/:last) #11413 [nkallen, thechrisoshow]

* Merge the has_finder gem, renamed as 'named_scope'. #11404 [nkallen]

```
class Article < ActiveRecord::Base
  named_scope :published, :conditions => {:published => true}
  named_scope :popular, :conditions => ...
end

Article.published.paginate(:page => 1)
Article.published.popular.count
Article.popular.find(:first)
Article.popular.find(:all, :conditions => {...})
```

See <http://pivots.pivotallabs.com/users/nick/blog/articles/284-hasfinder-it-s-now-easier-than-ever-to-create-complex-re-usable-sql-queries>

* Add has_one :through support. #4756 [thechrisoshow]

* Migrations: create_table supports primary_key_prefix_type. #10314 [student, thechrisoshow]

* Added logging for dependency load errors with fixtures #11056 [stuthulhu]

* Time zone aware attributes use Time#in_time_zone [Geoff Buesing]

* Fixed that scoped joins would not always be respected #6821 [Theory/Danger]

- * Ensure that ActiveRecord::Calculations disambiguates field names with the table name. #11027 [cavalle]
- * Added add/remove_timestamps to the schema statements for adding the created_at/updated_at columns on existing tables #11129 [jramirez]
- * Added ActiveRecord::Base.find(:last) #11338 [miloops]
- * test_native_types expects DateTime.local_offset instead of DateTime.now.offset; fixes test breakage due to dst transition [Geoff Buesing]
- * Add :readonly option to HasManyThrough associations. #11156 [miloops]
- * Improve performance on :include/:conditions/:limit queries by selectively joining in the pre-query. #9560 [dasil003]
- * Perf fix: Avoid the use of named block arguments. Closes #11109 [adymo]
- * PostgreSQL: support server versions 7.4 through 8.0 and the ruby-pg driver. #11127 [jdavis]
- * Ensure association preloading doesn't break when an association returns nil. ##11145 [GMFlash]
- * Make dynamic finders respect the :include on HasManyThrough associations. #10998. [cpytel]
- * Base#instantiate_time_object only uses Time.zone when Base.time_zone_aware_attributes is true; leverages Time#time_with_datetime_fallback for readability [Geoff Buesing]
- * Refactor ConnectionAdapters::Column.new_time: leverage DateTime failover behavior of Time#time_with_datetime_fallback [Geoff Buesing]

Ruby on Rails 2.1 - What's New

- * Improve associations performance by using symbol callbacks instead of string callbacks. #11108 [adymo]
- * Optimise the BigDecimal conversion code. #11110 [adymo]
- * Introduce the :readonly option to all associations. Records from the association cannot be saved. #11084 [miloops]
- * Multiparameter attributes for time columns fail over to DateTime when out of range of Time [Geoff Buesing]
- * Base#instantiate_time_object uses Time.zone.local() [Geoff Buesing]
- * Add timezone-aware attribute readers and writers. #10982 [Geoff Buesing]
- * Instantiating time objects in multiparameter attributes uses Time.zone if available. #10982 [rick]
- * Add note about how ActiveRecord::Observer classes are initialized in a Rails app. #10980 [fxn]
- * MySQL: omit text/blob defaults from the schema instead of using an empty string. #10963 [mdeiters]
- * belongs_to supports :dependent => :destroy and :delete. #10592 [Jonathan Viney]
- * Introduce preload query strategy for eager :includes. #9640 [Frederick Cheung, Aleksey Kondratenko, codafoo]
- * Support aggregations in finder conditions. #10572 [Ryan Kinderman]
- * Organize and clean up the Active Record test suite. #10742 [John Barnette]
- * Ensure that modifying has_and_belongs_to_many actions clear the query cache. Closes #10840 [john.andrews]

- * Fix issue where `Table#references` doesn't pass a `:null` option to a `*_type` attribute for polymorphic associations. Closes #10753 [railsjitsu]
- * Fixtures: removed support for the ancient pre-YAML file format. #10736 [John Barnette]
- * More thoroughly quote table names. #10698 [dimdenis, lotswholetime, Jeremy Kemper]
- * `update_all` ignores `scoped :order` and `:limit`, so `post.comments.update_all` doesn't try to include the comment order in the update statement. #10686 [Brendan Ribera]
- * Added `ActiveRecord::Base.cache_key` to make it easier to cache Active Records in combination with the new `ActiveSupport::Cache::* libraries` [DHH]
- * Make sure CSV fixtures are compatible with ruby 1.9's new csv implementation. [JEG2]
- * Added `by` parameter to `increment`, `decrement`, and their bang varieties so you can do `player1.increment!(:points, 5)` #10542 [Sam]
- * Optimize `ActiveRecord::Base#exists?` to use `#select_all` instead of `#find`. Closes #10605 [jamesh, fcheung, protocol]
- * Don't unnecessarily load `has_many` associations in `after_update` callbacks. Closes #6822 [stopdropandrew, canadaduane]
- * Eager `belongs_to :include` infers the foreign key from the association name rather than the class name. #10517 [Jonathan Viney]
- * SQLite: fix `rename_` and `remove_column` for columns with unique indexes. #10576 [Brandon Keepers]

Ruby on Rails 2.1 - What's New

* Ruby 1.9 compatibility. #10655 [Jeremy Kemper, Dirkjan Bussink]

ACTIVERESOURCE

2.1.0 (May 31st, 2008)*

* Fixed response logging to use length instead of the entire thing (seangeo) [#27]

* Fixed that to_param should be used and honored instead of hardcoding the id #11406 [gspiers]

* Improve documentation. [Radar, Jan De Poorter, chuyeow, xavierishay, danger, miloops, Xavier Noria, Sunny Ripert]

* Use HEAD instead of GET in exists? [bscofield]

* Fix small documentation typo. Closes #10670 [l.guidi]

* find_or_create_resource_for handles module nesting. #10646 [xavier]

* Allow setting ActiveResource::Base#format before #site. [rick]

* Support agnostic formats when calling custom methods. Closes #10635 [joerichsen]

* Document custom methods. #10589 [Cheah Chu Yeow]

* Ruby 1.9 compatibility. [Jeremy Kemper]

ACTIVESUPPORT

- * `TimeZone#to_s` shows offset as GMT instead of UTC, because GMT will be more familiar to end users (see time zone selects used by Windows OS, google.com and yahoo.com.) Reverts [8370] [Geoff Buesing]
- * `Hash.from_xml`: datetime xml types overflow to Ruby DateTime class when out of range of Time. Adding tests for utc offsets [Geoff Buesing]
- * `TimeWithZone #+` and `#-` : ensure overflow to DateTime with Numeric arg [Geoff Buesing]
- * `Time#to_json`: don't convert to utc before encoding. References #175 [Geoff Buesing]
- * Remove unused `JSON::RESERVED_WORDS`, `JSON.valid_identifier?` and `JSON.reserved_word?` methods. Resolves #164. [Cheah Chu Yeow]
- * Adding `Date.current`, which returns `Time.zone.today` if `config.time_zone` is set; otherwise returns `Date.today` [Geoff Buesing]
- * `TimeWithZone`: date part getter methods (`#year` `#mon` `#day` etc) are defined on class; no longer relying on `method_missing` [Geoff Buesing]
- * `Time.zone.parse` return nil for strings with no date information [Geoff Buesing]
- * `Time.zone.parse` respects offset information in string. Resolves #105. [Scott Fleckenstein, Geoff Buesing]
- * Added Ruby 1.8 implementation of `Process.daemon`

Ruby on Rails 2.1 - What's New

- * Duration `#since` and `#ago` with no argument (e.g., `5.days.ago`) return `TimeWithZone` when `config.time_zone` is set. Introducing `Time.current`, which returns `Time.zone.now` if `config.time_zone` is set, otherwise just returns `Time.now` [Geoff Buesing]
- * `Time#since` behaves correctly when passed a `Duration`. Closes #11527 [kemiller]
- * Add `#getutc` alias for `DateTime#utc` [Geoff Buesing]
- * Refactor `TimeWithZone`: don't send `#since`, `#ago`, `#+`, `#-`, `#advance` through `method_missing` [Geoff Buesing]
- * `TimeWithZone` respects `config.active_support.use_standard_json_time_format` [Geoff Buesing]
- * Add `config.active_support.escape_html_entities_in_json` to allow disabling of html entity escaping. [rick]
- * Improve documentation. [Xavier Noria]
- * Modified `ActiveSupport::Callbacks::Callback#call` to accept multiple arguments.
- * `Time #yesterday` and `#tomorrow` behave correctly crossing DST boundary. Closes #7399 [sblackstone]
- * `TimeWithZone`: Adding tests for dst and leap day edge cases when advancing time [Geoff Buesing]
- * `TimeWithZone#method_missing`: send to `utc` to advance with dst correctness, otherwise send to `time`. Adding tests for time calculations methods [Geoff Buesing]
- * Add `config.active_support.use_standard_json_time_format` setting so that `Times` and `Dates` export to ISO 8601 dates. [rick]
- * `TZInfo`: Removing unneeded `TimezoneProxy` class [Geoff Buesing]

- * TZInfo: Removing unneeded TimezoneIndexDefinition, since we're not including Indexes::Timezones [Geoff Buesing]
- * Removing unnecessary uses_tzinfo helper from tests, given that TZInfo is now bundled [Geoff Buesing]
- * Bundling abbreviated version of TZInfo gem 0.3.8: only the classes and zone definitions required to support Rails time zone features are included. If a recent version of the full TZInfo gem is installed, this will take precedence over the bundled version [Geoff Buesing]
- * TimeWithZone#marshal_load does zone lookup via Time.get_zone, so that tzinfo/Olson identifiers are handled [Geoff Buesing]
- * Time.zone= accepts TZInfo::Timezone instances and Olson identifiers; wraps result in TimeZone instance [Geoff Buesing]
- * TimeWithZone time conversions don't need to be wrapped in TimeOrDateTime, because TZInfo does this internally [Geoff Buesing]
- * TimeWithZone#usec returns 0 instead of error when DateTime is wrapped [Geoff Buesing]
- * Improve documentation. [Radar, Jan De Poorter, chuyeow, xaviershay, danger, miloops, Xavier Noria, Sunny Ripert]
- * Ensure that TimeWithZone#to_yaml works when passed a YAML::Emitter. [rick]
- * Ensure correct TimeWithZone#to_date [Geoff Buesing]

Ruby on Rails 2.1 - What's New

- * Make `TimeWithZone` work with `tzinfo 0.2.x`: use `TZInfo::Timezone#zone_identifier` alias for `#abbreviation`, silence warnings on tests. Raise `LoadError` when `TZInfo` version is `< 0.2` by sniffing for `TZInfo::TimeOrDateTime` constant. Move all `tzinfo`-dependent `TimeZone` tests into `uses_tzinfo` block [Geoff Buesing]
- * `Time`, `DateTime` and `TimeWithZone` `#in_time_zone` defaults to `Time.zone`. Removing now unneeded `#in_current_time_zone` [Geoff Buesing]
- * `TZInfo` caches `Timezone` instances in its own internal hash cache, so `TimeZone::MAPPING` doesn't need to cache them as well [Geoff Buesing]
- * Adding `TimeZone#parse` [Geoff Buesing]
- * Adding `TimeZone#at` and `DateTime#to_f` [Geoff Buesing]
- * `TimeWithZone` responds to Ruby 1.9 weekday-named query methods [Geoff Buesing]
- * `TimeWithZone` caches `TZInfo::TimezonePeriod` used for time conversion so that it can be reused, and enforces DST rules correctly when instance is created from a local time [Geoff Buesing]
- * Fixed that `BufferedLogger` should create its own directory if one doesn't already exist #11285 [lotswholetime]
- * Fix Numeric time tests broken by DST change by anchoring them to fixed times instead of `Time.now`. Anchor `TimeZone#now` DST test to time specified with `Time.at` instead of `Time.local` to work around platform differences with `Time.local` and DST representation [Geoff Buesing]
- * Removing unneeded `#change_time_zone` method from `Time`, `DateTime` and `TimeWithZone` [Geoff Buesing]
- * `TimeZone` `#local` and `#now` correctly enforce DST rules [Geoff Buesing]

- * TimeWithZone instances correctly enforce DST rules. Adding TimeZone#period_for_utc [Geoff Buesing]
- * test_time_with_datetime_fallback expects DateTime.local_offset instead of DateTime.now.offset [Geoff Buesing]
- * Adding TimeWithZone #marshal_dump and #marshal_load [Geoff Buesing]
- * Add OrderedHash#to_hash [josh]
- * Adding Time#end_of_day, _quarter, _week, and _year. #9312 [Juanjo Bazan, Tarmo Tānav, BigTitus]
- * Adding TimeWithZone#between? [Geoff Buesing]
- * Time.=== returns true for TimeWithZone instances [Geoff Buesing]
- * TimeWithZone #+ and #- behave consistently with numeric arguments regardless of whether wrapped time is a Time or DateTime; consistently answers false to #acts_like?(:date) [Geoff Buesing]
- * Add String#squish and String#squish! to remove consecutive chunks of whitespace. #11123 [jordi, Henrik N]
- * Serialize BigDecimals as Floats when using to_yaml. #8746 [ernesto.jimenez]
- * Adding TimeWithZone #to_yaml, #to_datetime, #eq!? and method aliases for duck-typing compatibility with Time [Geoff Buesing]
- * TimeWithZone #in_time_zone returns +self+ if zone argument is the same as #time_zone [Geoff Buesing]
- * Adding TimeWithZone #to_a, #to_f, #to_i, #httpdate, #rfc2822 [Geoff Buesing]
- * Pruning unneeded TimeWithZone#change_time_zone_to_current [Geoff Buesing]

Ruby on Rails 2.1 - What's New

- * `Time#zone=`, `#in_time_zone` and `#change_time_zone` accept a Duration [Geoff Buesing]
- * `Time#in_time_zone` handles `Time.local` instances correctly [Geoff Buesing]
- * Pruning unneeded `Time#change_time_zone_to_current`. Enhanced docs to `#change_time_zone` to explain the difference between this method and `#in_time_zone` [Geoff Buesing]
- * `TimeZone#new` method renamed `#local`; when used with `Time.zone`, constructor now reads: `Time.zone.local()` [Geoff Buesing]
- * Added `Base64.encode64s` to encode values in base64 without the newlines. This makes the values immediately usable as URL parameters or memcache keys without further processing [DHH]
- * Remove `:nodoc:` entries around the ActiveSupport test/unit assertions. #10946 [dancroak, jamesh]
- * Add `Time.zone_default` accessor for setting the default time zone. `Rails::Configuration.time_zone` sets this. #10982 [Geoff Buesing]
- * `cache.fetch(key, :force => true)` to force a cache miss. [Jeremy Kemper]
- * Support retrieving TimeZones with a Duration. `TimeZone[-28800] == TimeZone[-480.minutes]`. [rick]
- * `TimeWithZone#-` added, so that `#-` can handle a `Time` or `TimeWithZone` argument correctly [Geoff Buesing]
- * `with_timezone` test helper renamed `with_env_tz`, to distinguish between setting `ENV['TZ']` and setting `Time.zone` in tests [Geoff Buesing]
- * `Time#-` coerces `TimeWithZone` argument to a `Time` instance so that difference in seconds can be calculated. Closes #10914 [Geoff Buesing, yyyc514]

- * Adding UTC zone to TimeZone; TimeWithZone no longer has to fake UTC zone with nil [Geoff Buesing]
- * Time.get_zone refactored to private method, given that the encapsulated logic is only useful internally [Geoff Buesing]
- * Time.zone uses thread-local variable for thread safety. Adding Time.use_zone, for overriding Time.zone locally inside a block. Removing unneeded Time.zone_reset! [Geoff Buesing]
- * TimeZone#to_s uses UTC rather than GMT; reapplying change that was undone in [8679]. #1689 [Cheah Chu Yeow]
- * Time.days_in_month defaults to current year if no year is supplied as argument #10799 [Radar], uses Date.gregorian_leap? to determine leap year, and uses constant lookup to determine days in month [Geoff Buesing]
- * Adding Time and DateTime #compare_with_coercion, which layers behavior on #<=> so that any combination of Time, DateTime and ActiveSupport::TimeWithZone instances can be chronologically compared [Geoff Buesing]
- * TimeZone#now returns an ActiveSupport::TimeWithZone [Geoff Buesing]
- * Time #in_current_time_zone and #change_time_zone_to_current return self when Time.zone is nil [Geoff Buesing]
- * Remove unneeded #to_datetime_default_s alias for DateTime#to_s, given that we inherit a #to_default_s from Date that does exactly the same thing [Geoff Buesing]
- * Refactor Time and DateTime #to_formatted_s: use ternary instead of nested if/else [Geoff Buesing]

Ruby on Rails 2.1 - What's New

- * Adding Time and DateTime `#formatted_offset`, for outputting `+HH:MM` utc offset strings with cross-platform consistency [Geoff Buesing]
- * Adding `alternate_utc_string` option to `TimeZone#formatted_offset`. Removing unneeded `TimeZone#offset`. [Geoff Buesing]
- * Introduce `ActiveSupport::TimeWithZone`, for wrapping Time instances with a `TimeZone`. Introduce instance methods to Time for creating `TimeWithZone` instances, and class methods for managing a global time zone. [Geoff Buesing]
- * Replace non-dst-aware `TimeZone` class with dst-aware class from `tzinfo_timezone` plugin. `TimeZone#adjust` and `#unadjust` are no longer available; `tzinfo` gem must now be present in order to perform time zone calculations, via `#local_to_utc` and `#utc_to_local` methods. [Geoff Buesing]
- * Extract `ActiveSupport::Callbacks` from Active Record, test case setup and teardown, and `ActionController::Dispatcher`. #10727 [Josh Peek]
- * Introducing `DateTime#utc`, `#utc?` and `#utc_offset`, for duck-typing compatibility with Time. Closes #10002 [Geoff Buesing]
- * `Time#to_json` uses `Numeric#to_utc_offset_s` to output a cross-platform-consistent representation without having to convert to `DateTime`. References #9750 [Geoff Buesing]
- * Refactor number-to-HH:MM-string conversion logic from `TimeZone#formatted_offset` to a reusable `Numeric#to_utc_offset_s` method. [Geoff Buesing]
- * Continue evolution toward `ActiveSupport::TestCase`. #10679 [Josh Peek]

- * `TestCase`: introduce declared setup and teardown callbacks. Pass a list of methods and an optional block to call before setup or after teardown. Setup callbacks are run in the order declared; teardown callbacks are run in reverse. [Jeremy Kemper]
- * Added `ActiveSupport::Gzip.decompress/compress(source)` as an easy wrapper for Zlib [Tobias Luetke]
- * Included `MemCache-Client` to make the improved `ActiveSupport::Cache::MemCacheStore` work out of the box [Bob Cottrell, Eric Hodel]
- * Added `ActiveSupport::Cache::*` framework as an extraction from `ActionController::Caching::Fragments::*` [DHH]
- * Fixed `String#titleize` to work for strings with 's too #10571 [trek]
- * Changed the implementation of `Enumerable#group_by` to use a double array approach instead of a hash such that the insert order is honored [DHH/Marcel]
- * remove multiple enumerations from `ActiveSupport::JSON#convert_json_to_yaml` when dealing with date/time values. [rick]
- * `Hash#symbolize_keys` skips keys that can't be symbolized. #10500 [Brad Greenlee]
- * Ruby 1.9 compatibility. #1689, #10466, #10468, #10554, #10594, #10632 [Cheah Chu Yeow, Pratik Naik, Jeremy Kemper, Dirkjan Bussink, fxn]
- * `TimeZone#to_s` uses UTC rather than GMT. #1689 [Cheah Chu Yeow]
- * Refactor of `Hash#symbolize_keys!` to use `Hash#replace`. Closes #10420 [ReinH]
- * Fix `HashWithIndifferentAccess#to_options!` so it doesn't clear the options hash. Closes #10419 [ReinH]

Ruby on Rails 2.1 - What's New

RAILTIES

- * script/dbconsole fires up the command-line database client. #102 [Steve Purcell]
- * Fix bug where plugin init.rb files from frozen gem specs weren't being run. (pjb3) [#122 state:resolved]
- * Made the location of the routes file configurable with config.routes_configuration_file (Scott Fleckenstein) [#88]
- * Rails Edge info returns the latest git commit hash [Francesc Esplugas]
- * Added Rails.public_path to control where HTML and assets are expected to be loaded from (defaults to Rails.root + "/public") #11581 [nicksieger]
- * rake time:zones:local finds correct base utc offset for zones in the Southern Hemisphere [Geoff Buesing]
- * Don't require rails/gem_builder during rails initialization, it's only needed for the gems:build task. [rick]
- * script/performance/profiler compatibility with the ruby-prof >= 0.5.0. Closes #9176. [Catfish]
- * Flesh out rake gems:unpack to unpack all gems, and add rake gems:build for native extensions. #11513 [ddollar]

```
rake gems:unpack          # unpacks all gems
rake gems:unpack GEM=mygem # unpacks only the gem 'mygem'

rake gems:build           # builds all unpacked gems
rake gems:build GEM=mygem # builds only the gem 'mygem'
```

* Add `config.active_support` for future configuration options. Also, add more new Rails 3 config settings to `new_rails_defaults.rb` [rick]

* Add `Rails.logger`, `Rails.root`, `Rails.env` and `Rails.cache` shortcuts for `RAILS_*` constants [pratik]

* Allow files in plugins to be reloaded like the rest of the application. [rick]

Enables or disables plugin reloading.

```
config.reload_plugins = true
```

You can get around this setting per plugin.

If `#reload_plugins? == false` (DEFAULT), add this to your plugin's `init.rb` to make it reloadable:

```
Dependencies.load_once_paths.delete lib_path
```

If `#reload_plugins? == true`, add this to your plugin's `init.rb` to only load it once:

```
Dependencies.load_once_paths << lib_path
```

* Small tweak to allow plugins to specify gem dependencies. [rick]

```
# OLD open_id_authentication plugin init.rb
require 'yadis'
require 'openid'
ActionController::Base.send :include, OpenIdAuthentication

# NEW
config.gem "ruby-openid", :lib => "openid", :version => "1.1.4"
config.gem "ruby-yadis", :lib => "yadis", :version => "0.3.4"
```

Ruby on Rails 2.1 - What's New

```
config.after_initialize do
  ActionController::Base.send :include, OpenIdAuthentication
end
```

* Added config.gem for specifying which gems are required by the application, as well as rake tasks for installing and freezing gems. [rick]

```
Rails::Initializer.run do |config|
  config.gem "bj"
  config.gem "hpricot", :version => '0.6', :source => "http://code.whytheluckystiff.net"
  config.gem "aws-s3", :lib => "aws/s3"
end

# List required gems.
rake gems

# Install all required gems:
rake gems:install

# Unpack specified gem to vendor/gems/gem\_name-x.x.x
rake gems:unpack GEM=bj
```

* Removed the default .htaccess configuration as there are so many good deployment options now (kept it as an example in README) [DHH]

* config.time_zone accepts TZInfo::Timezone identifiers as well as Rails TimeZone identifiers [Geoff Buesing]

* Rails::Initializer#initialize_time_zone raises an error if value assigned to config.time_zone is not recognized. Rake time zone tasks only require ActiveSupport instead of entire environment [Geoff Buesing]

* Stop adding the antiquated test/mocks/* directories and only add them to the path if they're still there for legacy reasons [DHH]

* Added that gems can now be plugins if they include rails/init.rb #11444 [jbarnette]

* Added Plugin#about method to programmatically access the about.yml in a plugin #10979 [lazyatom]

```
plugin = Rails::Plugin.new(path\_to\_my\_plugin)
plugin.about["author"] # => "James Adam"
plugin.about["url"] # => "http://interblah.net"
```

* Improve documentation. [Radar, Jan De Poorter, chuyeow, xaviershay, danger, miloops, Xavier Noria, Sunny Ripert]

* Added config.time_zone = 'UTC' in the default environment.rb [Geoff Buesing]

* Added rake tasks time:zones:all, time:zones:us and time:zones:local for finding time zone names for config.time_zone option [Geoff Buesing]

* Add config.time_zone for configuring the default Time.zone value. #10982 [Geoff Buesing]

* Added support for installing plugins hosted at git repositories #11294 [danger]

* Fixed that script/generate would not look for plugin generators in plugin_paths #11000 [glv]

* Fixed database rake tasks to work with charset/collation and show proper error messages on failure. Closes #11301 [matt]

* Added a -e/--export to script/plugin install, uses svn export. #10847 [jon@blankpad.net])

Ruby on Rails 2.1 - What's New

- * Reshuffle load order so that routes and observers are initialized after plugins and app initializers. Closes #10980 [rick]
- * Git support for script/generate. #10690 [ssoroka]
- * Update scaffold to use labels instead of bold tags. Closes #10757 [zach-inglis-lt3]
- * Resurrect WordNet synonym lookups. #10710 [tom./, matt]
- * Added config.cache_store to environment options to control the default cache store (default is FileStore if tmp/cache is present, otherwise MemoryStore is used) [DHH]
- * Added that rails:update is run when you do rails:freeze:edge to ensure you also get the latest JS and config files #10565 [jeff]
- * SQLite: db:drop:all doesn't fail silently if the database is already open. #10577 [Cheah Chu Yeow, mrichman]
- * Introduce native mongrel handler and push mutex into dispatcher. [Jeremy Kemper]
- * Ruby 1.9 compatibility. #1689, #10546 [Cheah Chu Yeow, frederico]

RUBY ON RAILS 2.1

WHAT'S NEW ?

```
def temp_partials
  p = temp_partials.first
  p.respond_to?(:path) ? p.path : p.to_s
end

def render_partial(partial_path, object_assigns = nil, local_assigns = nil)
  case partial_path
  when String, Symbol, NilClass
    path, partial_name = partial_pieces(partial_path)
    object = extracting_object(partial_name, object_assigns)
    local_assigns = local_assigns ? local_assigns.clone : {}
    add_counter_to_local_assigns!(partial_name, local_assigns)
    add_object_to_local_assigns!(partial_name, local_assigns, object)

    if logger && logger.debug?
      ActionController::Base.benchmark("Rendered #{path}/_#{partial_name}")
      render("#{path}/_#{partial_name}", local_assigns)
    end
  end
end
```

CARLOS BRANDO REVIEW: MARCOS TAPAJÓS - COVER: DANIEL LOPES

TRANSLATION: IceskYsl, jesse.cai, suave.su, dongbin, rubyjin, 404, ashchan, cash.zhao

EDITOR: Libin Pan

CHINESE VERSION