



5 little-known Rails methods

Posted by **Eric Chapweske** on Wednesday, April 23

While the next release of Rails appears to be coming up, there's still plenty of small, useful features from previous releases that aren't widely used.

A few of my favorites:

1. `query_attribute`
2. `polymorphic_path`
3. `debug`
4. `rake -T 'query'` (Not Rails specific, but still handy!)
5. `extract_options!`

1. ActiveRecord's `query_attribute`

Query methods are available for each of a record's attributes, providing for a cleaner way to check for the presence of an attribute.

```
1
2 # == Schema Information
3 # Schema version: 17
4 #
5 # Table name: users
6 #
7 # id              :integer(11)      not null, primary key
8 # first_name      :string(255)
9 # last_name       :string(255)
10
11 # Original
12 class User < ActiveRecord::Base
13   def named?
14     !first_name.blank? && !last_name.blank?
15   end
16 end
17
18 # Refactored to use query_attribute
19 class User < ActiveRecord::Base
20   def named?
21     first_name? && last_name?
22   end
23 end
```

2. Indifferent links with polymorphic paths.

Rails has polymorphic edit/new/formatted path routing available out of the box. Providing an array will namespace the path with those array parameters. Available Methods:

`(edit|new|formatted).polymorphic_path(record_or_hash_or_array)`

```
1
2 # Before:
3 <% if @record.is_a?(User) %>
4 <%= user_path(@record) %>
5 <% elsif @record.is_a?(Friend)
6 <%= friend_path(@record) %>
7 ... etc.
8
9 # After:
10 <%= polymorphic_path(@record) %>
```

Quite a few options are supported, and other Rails methods take advantage of polymorphic routing:

```

1
2 # Paths can be namespaced:
3 #=> admin/users/5/edit
4 edit_polymorphic_path([:admin, @record])
5
6 # Polymorphic urls are also internally used by helpers:
7 # redirects to store_path(@store)
8 redirect_to @store
9
10 # builds a form with an action to 'new_admin_stores_path'
11 #=> <form action="admin/stores/new" ... />
12 form_for([:admin, @store.new])
13
14 # <a href="/stores/5">A store in Minneapolis, MN</a>
15 link_to @store.name, @store
16

```

3. debug

Especially useful when starting out a project, this is quick way to understand what objects are being used in the view.

```

1
2
3 <%= debug @user %>
4
5 # Yields this in the view:
6 # - !ruby/object:User
7 #   attributes:
8 #     salt: 7be4287a1b27426fa6e5b6d733c707dd66425e82
9 #     updated_at: 2008-04-22 19:01:23
10 #     crypted_password: abb611def895dac923ba8ea59a78451f77473d5e
11 #     id: "1"
12 #     first_name: Eric
13 #     last_name: Chapweske
14 #     created_at: 2008-04-06 01:45:33
15 #   attributes_cache: {}

```

4. rake -T task

This is a handy Rake feature, and not limited to Rails. Can't remember the exact syntax for a particular rake task? Trim the results generated with `rake -T` with an optional search parameter.

```

1
2 $rake -T
3
4 rake annotate_models           # Add schema information (as comments)...
5 rake audit:purge              # Removes Audit records older than 2 m...
6 rake db:abort_if_pending_migrations # Raises an error if there are pending...
7 ... etc.
8 rake tmp:sockets:clear        # Clears all files in tmp/sockets
9
10 // Searching by the task's name
11
12 $rake -T db:migrate:r
13
14 rake db:migrate:redo          # Rollbacks the database one migration and re migrat...
15 rake db:migrate:reset         # Resets your database using your migrations for the...
16

```

5. extract_options!

While not needed very often, Rails comes bundled with a method to extract the options from methods that utilize the splat operator. This method removes the last object from an array if it's a Hash, otherwise an empty hash is returned.

```
1
2
3 class Story < ActiveRecord::Base
4
5   # Example:
6   # Story.published_and_tagged_with('deep', 'thoughts', :order => 'created_at desc')
7   # The generated options for this method look like this:
8   #=> { :include => :tags, :order => 'created_at desc' }
9   def self.published_and_tagged_with(*tag_names)
10    options = tag_names.extract_options!
11    options[:include] ||= :tags
12
13    ...
14  end
15 end
16
```

References

I wasn't able to find any write ups on the above methods, so reading the source code may be the best path if you're curious about their exact implementations.

Tags: [actionview](#) [activerecord](#) [activesupport](#) [polymorphic_path](#) [rails](#) [tips](#)
12 comments

Comments

[Leave a response](#)

grosser – April 25, 2008 @ 12:15 AM

I dunno where to put this, since there is no contact form...

i subscribed using google reader, and i get duplicate items A LOT, please have a look if you are sending valid RSS, or bookmark your feed yourself, it really starts to get annoying...

mg

Jon – April 25, 2008 @ 01:59 PM

Thanks, mg – we've investigated a bit. It looks like our feed is valid, but it could be a few other things (regarding ids/guids or edits). And it only happens in some RSS readers. We'll see what we can do.

Anyone have expertise with this sort of thing?

Dylan – April 25, 2008 @ 03:38 PM

polymorphic_path is a huge time saver. thanks for the heads up.

Jerod – April 25, 2008 @ 03:45 PM

Howdy!