



# Rails 2.0

## Cheatsheet

compiled by Ryan Wood ✕ [weblog.sourcescape.com](http://weblog.sourcescape.com)

### 2.0.2 UPDATE

- SQLite3 is the new default database
- Rails doesn't check for template changes in production mode
- rake secret
- Added db:migrate:redo and db:migrate:reset for rerunning existing migrations

### ACTIVERECORD

#### QUERY CACHE

Queries are cached out of the box for each request. Huge performance advantage. You can disable it.

```
def force_find(*args)
  uncached { find(*args) }
end
```

#### SEXY MIGRATIONS

```
create_table :posts do |t|
  t.integer :user_id, :category_id
  t.text :body
  t.timestamps
end
```

#### FOXY FIXTURES

```
# sellers.yml
shopify:
  name: Shopify

# products.yml
pimp_cup:
  seller: shopify
  name: Pimp cup
```

### XML & JSON SERIALIZATION

```
Person.new.from_xml("David")
person.to_json
```

### REMOVAL

- acts\_as... moved to plugins
- Commercial db adaptors removed from core
- ActionWebService out, ActiveResource in
- with\_scope added syntactic vinegar

### ACTIONPACK

#### COOKIE-BASED SESSIONS

They are way faster, encrypted (using OpenSSL::HMAC), and the new default. You still need the following info in your initializer.

```
config.action_controller.session = {
  :session_key => '_myapp_session',
  :secret      => 'chooseasecretcode'
}
```

If you change the session key, all existing sessions are invalidated!

#### PARTIAL LAYOUTS

Finally a way to do nested layouts in Rails.

```
<% for post in @posts do %>
  <%= render :partial => 'post',
    :layout => 'excerpt',
    :locals => {:post => post} %>
<% end %>
```

#### HTTP AUTHENTICATION

Used increasingly for RSS and Web Service authentication. The method to use is `authenticate_or_request_with_http_basic`.

#### BETTER EXCEPTION HANDLING

You can use `rescue_from` to handle various errors in each controller. Just create a method to local method handle it.

```
class MyController <
  ApplicationController

  rescue_from
    ActiveRecord::RecordNotFound,
    :with => :bad_record
  ...
end
```

#### HELPER ACCESS

This lets you add a helper (normally only available in views) for use in a controller.

```
helper :other_helper_name
```

#### NEW VIEW NAMES

View templates should now be in the form of `action.format.renderer`. Such as:

- **show.erb**: same show template for all formats
- **index.atom.builder**: uses the Builder format, previously known as `rxml`, to render an index action for the application/atom+xml mime type
- **edit.iphone.haml**: uses the custom HAML template engine (not included by default) to render an edit action for the custom `Mime::IPHONE` format

#### FAKE MIME TYPES

```
Mime.register_alias "text/html", :iphone

# Then later in an action...
respond_to do |format|
  format.html # renders index.html.erb
  format.iphone # renders index.iphone.erb
end
```

#### SIMPLY HELPFUL

Converts objects to URLs on the fly

```
redirect_to(person)
link_to(person.name, person)
form_for(person)
```

## CACHING JAVASCRIPTS

“Using `javascript_include_tag(:all, :cache => true)` will turn `public/javascripts/.js` into a single `public/javascripts/all.js` file in production, while still keeping the files separate in development, so you can work iteratively without clearing the cache.”

## ASSET DISTRIBUTION

To distribute your asset calls (like `image_tag`) to servers `asset1` through `asset4`...

```
ActionController::Base.asset_host
  = "assets%d.example.com"
```

## SECURITY

By default, Rails 2.0 has a built-in mechanism for dealing with CSRF attacks. By including a special token in all forms and Ajax requests, you can guard from having requests made from outside of your application.

## REQUEST PROFILER

```
$ cat login_session.rb
get_with_redirect '/'
say "GET / => #{path}"
post_with_redirect '/sessions',
  :username => 'john', :password => 'doe'
say "POST /sessions => #{path}"
$ ./script/performance/request -n 10 /
login_session.rb
```

## ATOM FEED HELPER

Makes it even simpler to create Atom feeds using an enhanced Builder syntax.

## REST

*Technically this is still part of ActionController, but there is so much here that I gave it it's own space.*

## THE SEMICOLON IS GONE

This shouldn't create much havoc if you are using named routes.

```
/posts/:id;archive is now /posts/:id/archive
```

## AUTOMATIC NESTED ROUTE PREFIXING

```
map.resources :comments
map.resources :posts do |post|
  post.resources :comments
end
```

Pre-2.0 you would have to add a `:name_prefix` key to avoid name clashes, now these are automatically created:

```
post_comments(post)
  /posts/:post_id/comments

new_post_comment(post)
  /posts/:post_id/comments/new
```

## NAMESPACES

```
map.namespace(:admin) do |admin|
  admin.resources :posts
end
```

## HAS\_MANY & HAD\_ONE ROUTE HELPERS

```
map.resources :posts,
  :has_many => [:comments, :tags],
  :has_one => :user
```

## SCRIPT/GENERATE CHANGES

REST has taken over. The Rails 1.2 `scaffold_resource` generator has become the the `scaffold` generator.

## RAILTIES

### SIMPLIFIED FUNCTIONAL TESTS

You no longer have to setup functional tests with a request and response object. They now extend `ActionController::TestCase` rather than `Test::Unit::TestCase`

### COLLECTION FIXTURES

```
def test_post_find
  assert_equal posts(:rails2, :peepcode),
    Post.find(:all)
end
```

## ADDED TO RAKE

```
rake routes
rake db:create RAILS_ENV=test
rake db:create:all
rake db:drop RAILS_ENV=test
rake db:drop:all
rake db:reset
rake db:version
rake notes (TODO, FIXME and OPTIMIZE)
```

## DEBUGGING

Rails now have a real debugger (albeit from the command line). First, install the gem...

```
sudo gem install -y ruby-debug
```

Then put 'debugger' in your code where you want debugging to start. Start the rails server script with the '-u' or '--debugger' option.

```
script/server --debugger
```

To debug a test, run

```
rdebug -n test/unit/user_test.rb
```

Check out the `ruby-debug` project for all the available options and commands. There are a ton.

## CLEAN UP THE ENVIRONMENT

Use the new `config/initializers` directory to add custom configuration rather than dropping them directly into the `environment.rb` file.

## PLUGIN ORDER

If you have plugin dependencies...

```
config.plugins = [ :acts_as_list, :all ]
```

## DEPRECATIONS

- `params` not `@params`
- `find(:all)` not `find_all`
- don't use components (just don't)
- `form_for` not `form_tag`
- pagination is gone (use a plugin)
- `smtp_settings` not `server_settings`