

Rails 3.0 Overview

a quick primer for Rails 2.0 users

Mike Hagedorn

Rails 3.0 Overview

- Prerequisites
- Rails Commands
- Bundler
- Routes
- ActionController
- ActionMailer
- ActiveRecord
- ActiveRecord
- UJS
- Additional Resources

Rails 3.0 Beta 3

- ruby requirements
 - 1.8.7 or higher
 - support for previous versions dropped
 - lots of talk of requiring 1.9.1+
 - “Rails 3 will definitely prefer Ruby 1.9.2 — when it’s released — and require Ruby 1.8.7 or later” — Jeremy Kemper

Ruby 1.9.2

- Its pretty clear 1.9.2 is the right way to go
- Installation of newer versions of ruby a scary proposition
- so use rvm - Ruby Version Manager

rvm

- Allows you to switch between available Ruby versions without messing up your current install
- Allows you to have different versions of ruby running in different terminal sessions

Installing rvm

```
bash < <( curl http://rvm.beginrescueend.com/releases/rvm-install-head
```

```
rvm install 1.9.2-head
```

See <http://rvm.beginrescueend.com> for more details

Set rvm defaults

```
rvm --default ruby-1.9.2-head
```

```
rvm default
```

```
ruby -v
```

```
ruby 1.9.2dev (2010-05-06 trunk 27646) [x86_64-darwin10.3.0]
```

```
rvm system
```

```
ruby -v
```

```
ruby 1.8.7 (2009-06-12 patchlevel 174) [i686-darwin10.0.0]
```

Almost There...

```
gem install rails --pre  
gem install bundler
```


Rails 3.0 Overview

- Prerequisites
- **Rails Commands**
- Bundler
- Routes
- ActionController
- ActionMailer
- ActiveRecord
- ActiveRecord
- UJS
- Additional Resources

```
$ rails example_app
  create ....
$ cd example_app
$ rails
```

Usage: rails COMMAND [ARGS]

The most common rails commands are:

generate	Generate new code (short-cut alias: "g")
console	Start the Rails console (short-cut alias: "c")
server	Start the Rails server (short-cut alias: "s")

Is script at this point
shows only a rails script

Rails 2.0 to Rails 3.0 Commands

Version 2.0	Version 3.0
<code>script/generate</code>	<code>rails g</code>
<code>script/console</code>	<code>rails c</code>
<code>script/server</code>	<code>rails s</code>
<code>script/dbconsole</code>	<code>rails db</code>

Rails 3.0 Overview

- Prerequisites
- Rails Commands
- Bundler
- Routes
- ActionController
- ActionMailer
- ActiveRecord
- ActiveRecord
- UJS
- Additional Resources

Bundler

- Rails 2.0
 - specify gems inside enviroment.rb

```
config.gem "chronic"
```
 - \$ rake gems:install

Issues

- Coupled to Rails
- Dependency resolution marginal
- Conflicts not found until runtime
- Unpack created issues on some platforms

ActiveMerchant 1.4.2
activesupport >= 2.3.2

Rails 2.3.2
activesupport = 2.3.2

System Gems

Rails 2.3.2

Rails 2.3.3

ActiveMerchant 1.4.2

Rails App

ActiveMerchant 1.4.2
activesupport >= 2.3.2

Rails 2.3.2
activesupport = 2.3.2

System Gems

Rails 2.3.2

Rails App

Rails 2.3.3

ActiveMerchant 1.4.2

ActiveMerchant 1.4.2
activesupport >= 2.3.2

Rails 2.3.2
activesupport = 2.3.2

System Gems



Rails 2.3.2

Rails App

Rails 2.3.3

ActiveMerchant 1.4.2



already activated 2.3.3

silverchair solutions



Bundler Commands

- Gems are specified inside Gemfile

```
gem "haml"  
gem "chronic", '0.2.3'
```

- **\$ bundle install**

download and install gems into ~/.bundle directory, makes sure Gemfile dependencies are satisfied

- **\$ bundle package**

locks version and unpacks into /vendor/cache

With Bundler..

ActiveMerchant 1.4.2
activesupport \geq 2.3.2

Rails 2.3.2
activesupport = 2.3.2

System Gems

Rails 2.3.2

Rails 2.3.3

ActiveMerchant 1.4.2

Rails App

With Bundler..

ActiveMerchant 1.4.2
activesupport \geq 2.3.2

Rails 2.3.2
activesupport = 2.3.2

System Gems

Rails 2.3.3

dependencies evaled in a
global fashion

Rails App

Rails 2.3.2

ActiveMerchant 1.4.2

OK!



Rails 3.0 Overview

- Prerequisites
- Rails Commands
- Bundler
- **Routes**
- ActionController
- ActionMailer
- ActiveRecord
- ActiveRecord
- UJS
- Additional Resources

Routing

Rails 2

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

Rails 3

```
resources :posts do  
  resources :comments  
end
```

Routing

Rails 2

```
map.resources :posts, :member => { :confirm => :post, :notify => :post } do |post|
  post.resources :comments,
    :member => { :preview => :post },
    :collection => { :archived => :get }
end
```

Rails 3

```
resources :posts do
  member do
    post :confirm
    get :notify
  end
  resources :comments do
    member do
      post :preview
    end
    collection do
      get :archived
    end
  end
end
```

OR

```
resources :posts do
  member do
    post :confirm
    get :notify
  end
  resources :comments do
    post :preview, :on => :member
    get :archived, :on => :collection
  end
end
```

Regular Routes

Rails 2

```
map.connect 'parts/:id', :controller => 'inventory', :action => 'show'
```

Rails 3

```
match 'parts/:id', :to => 'inventory#show'
```


Named Routes

Rails 2

```
map.logout '/logout', :controller => 'sessions', :action => 'destroy'
```

Rails 3

```
match 'logout', :to => 'sessions#destroy', :as => "logout"
```

Root Route

Rails 2

```
map.root :controller => "welcome", :action => 'show'
```

Rails 3

```
root :to => 'welcome#show'
```

Http Verb Routes

Rails 2

```
map.connect '/articles/:year',  
  :controller => 'posts',  
  :action => 'index', :conditions => {:method => :get}
```

Rails 3

```
match '/articles/:year' => "posts#index", :via => :get
```

OR

```
get '/articles/:year' => "posts#index"
```

New Redirection

```
match "/foo/:id", :to => redirect("/bar/%{id}s")
```

redirects /foo/1 to /bar/1s

```
match 'account/proc/:name', :to => redirect { |params| "#{params[:name].pluralize}" }
```

redirects /account/proc/johns to /johns

Constraints

```
match('/:year' => "posts#index", :constraints => { :year => /\d{4}/ }
```

Rack

```
get 'hello' => proc { |env| [200, {}, "Hello Rack"] }
```

```
get 'rack_app' => CustomRackApp
```

rack apps return
code, headers, body

Rails 3.0 Overview

- Prerequisites
- Rails Commands
- Bundler
- Routes
- **ActionController**
- ActionMailer
- ActiveRecord
- ActiveRecord
- UJS
- Additional Resources

New Hierarchy

ActionController::Base

Normal Rails goodness you have come to love

ActionController::Metal

Exposes rack endpoints ala Sinatra

AbstractController::Base

Basic layer, delegates to concrete subclasses

New ActionController Syntax

```
class UsersController < ApplicationController
  def index
    @users = User.all
    respond_to do |format|
      format.html
      format.xml { render :xml => @users.to_xml }
    end
  end

  def show
    @user = User.find(params[:id])
    respond_to do |format|
      format.html # show.html.erb
      format.xml { render :xml => @user }
    end
  end
end
...
```

New ActionController Syntax

```
class UsersController < ApplicationController
  def index
    @users = User.all
    respond_to do |format|
      format.html
      format.xml { render :xml => @users.to_xml }
    end
  end

  def show
    @user = User.find(params[:id])
    respond_to do |format|
      format.html # show.html.erb
      format.xml { render :xml => @user }
    end
  end

  ...
end
```

New ActionController Syntax

```
class UsersController < ApplicationController
  respond_to :html, :xml, :json

  def index
    @users = User.all
    respond_with(@users)
  end

  def show
    @user = User.find(params[:id])
    respond_with(@user)
  end

  ...
end
```

scaffolds dont do this
because its too "magical"

Rails 3.0 Overview

- Prerequisites
- Rails Commands
- Bundler
- Routes
- ActionController
- **ActionMailer**
- ActiveRecord
- ActiveRecord
- UJS
- Additional Resources

ActionMailer

```
$ rails g mailer UserMailer welcome forgot_password  
create app/mailers/user_mailer.rb
```

Mailer Syntax

Rails 2

```
def welcome(user, subdomain)
  subject 'Welcome to TestApp'
  recipients user.email from 'admin@testapp.com'
  body :user => user, :subdomain => subdomain end
end
```

```
UserMailer.deliver_welcome(user, subdomain)
```

Rails 3

```
def welcome(user, subdomain)
  @user = user
  @subdomain = subdomain
  mail(:from => "admin@testapp.com", :to => user.email,
       :subject => "Welcome to TestApp")
end
```

```
UserMailer.welcome(user, subdomain).deliver
:subject => "Welcome to TestApp")
```

Rails 3.0 Overview

- Prerequisites
- Rails Commands
- Bundler
- Routes
- ActionController
- ActionMailer
- **ActiveRelation**
- ActiveRecord
- UJS
- Additional Resources

Active Relation

Rails2

```
@posts = Post.find(:all, :conditions => {:published => true})
```

Immediate Query, returns [Posts]

Rails3

```
@posts = Post.where(:published => true)
```

Delayed until used,
returns ActiveRecord::Relation

Active Relation

```
@posts = Post.where(:published => true)
if params[:order]
  @posts = @posts.order(params[:order])
end
```

```
@posts.each do |p| Evaluation occurs here
  ...
end
```

Arel Chaining

```
@posts = Post.where(:published => true)
@posts = @posts.order(params[:order])
```

```
@posts = Post.where(:published => true).order(params[:order])
```

New Finders

where(:conditions)

having(:conditions)

select

group

order

limit

offset

joins

includes(:include)

lock

readonly

from

Rails 3.0 Overview

- Prerequisites
- Rails Commands
- Bundler
- Routes
- ActionController
- ActionMailer
- ActiveRecord
- **ActiveModel**
- UJS
- Additional Resources

ActiveModel

- Extracted from ActiveRecord
- Each feature is a ruby module
 - Validation
 - Serialization
 - Callbacks
 - Observers

ActiveModel

- Allows “AR” features in PORO objects just by including the module
 - `to_json`
 - `validates_presence_of`
 - `others....`

ActiveModel

```
class Applicant
  include ActiveModel::Validations
  validates_presence_of :name, :email
  attr_accessor :name, :email
end
```

Rails 3.0 Overview

- Prerequisites
- Rails Commands
- Bundler
- Routes
- ActionController
- ActionMailer
- ActiveRecord
- ActiveModel
- **UJS**
- Additional Resources

UJS

- Existing html emitted by Rails 2.0 is messy

```
<a href="/users/1" onclick="if (confirm('Are you sure?')) { var f =
document.createElement('form'); f.style.display = 'none';
this.parentNode.appendChild(f); f.method = 'POST'; f.action = this.href;var m =
document.createElement('input'); m.setAttribute('type', 'hidden'); m.setAttribute
('name', '_method'); m.setAttribute('value', 'delete'); f.appendChild(m);f.submit
(); };return false;">Delete</a>
```

- Rails 3 uses attributes and js binding instead

```
<a rel="nofollow" data-method="delete" data-confirm="Are you sure?" class="delete"
href="/user/1">Delete</a>
```

- JS drivers now use data-* to attach behavior

UJS

- Easy to plug in jQuery instead of prototype
- jQuery driver at <http://github.com/rails/jquery-ujs>
- Put it in your javascripts directory
- Remember to include “csrf_meta_tag” in your layout head or it just wont work

Resources

- <http://rails3.community-tracker.com/permalinks/5/notes-from-the-field-upgrading-to-rails-3>
- http://github.com/rails/rails_upgrade

Thanks!

Mike Hagedorn

mike@silverchairsolutions.com

[@mwhagedorn](#)