4.0

Highlights in Rails 4.0

- At least Ruby 1.9.3
- Thread safe by default
- Strong Parameters
- Turbolinks
- Russian Doll Caching

Removed from master

- Rails.queue (ActiveSupport::Queue) => 4.1
- Asynchronous Action Mailer => 4.1

Extracted Gems

- Sprockets Rails
- ActiveResource
- AR Observer + AC Sweeper
- AR SessionStore
- AR deprecated_finders
- ...

Deprecation Policy

- 4.0 mostly just deprecates
- 4.1 removes those features
  So manually add the gems to Gemfile
- 5.0 removes support for some of those gems

Strong Parameters

Mass assignment protection on controllers

Rails 3

```
class Post < ActiveRecord::Base
  attr_accessible :title, :body
end

class PostsController < ApplicationController
  def create
    @post = Post.create(params[:post])
  end
end
```

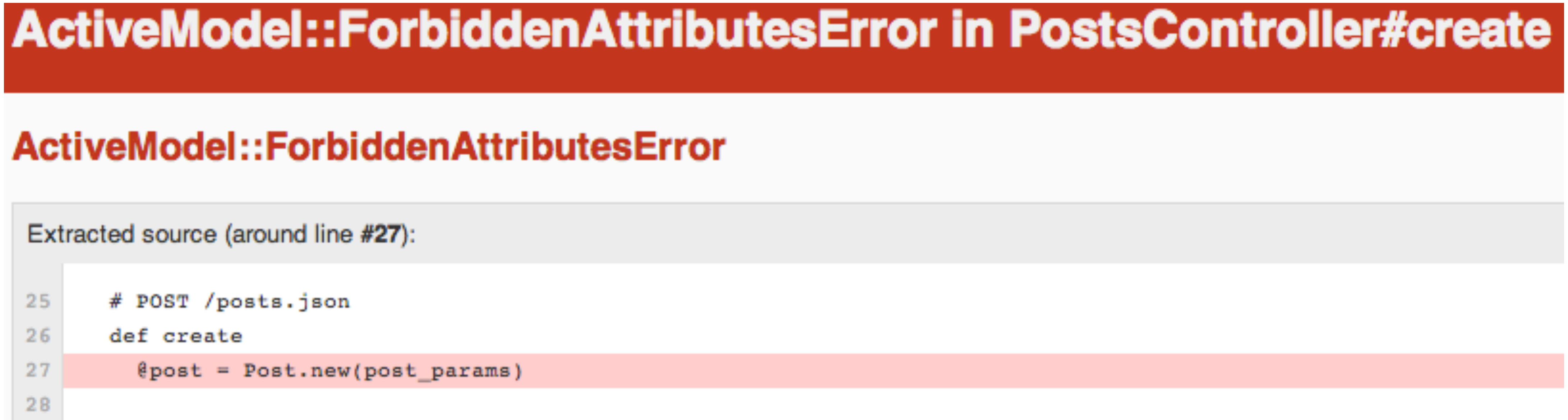
Rails 4

```
class Post < ActiveRecord::Base
end

class PostsController < ApplicationController
  def create
    @post = Post.create(post_params)
  end

  private

  def post_params
    params.require(:post).
      permit(:title, :body)
  end
end
```



ActiveModel::ForbiddenAttributesError in PostsController#create

ActiveModel::ForbiddenAttributesError

Extracted source (around line **#27**):

```
25    # POST /posts.json
26    def create
27      @post = Post.new(post_params)
28
```

Permitted Scalar Values

```
params.permit(:id)
```

- key id passes whitelisting if it appears in params and it has a permitted scalar value associated
- disallowed keys:
  development and test => log,
  other env => filter it out

Scalar Values

String, Symbol, NilClass, Numeric, TrueClass, FalseClass, Date, Time, DateTime, StringIO, IO and ActionDispatch::Http::UploadedFile

Permit an Array

```
params.permit(:tags => [])
```

- must be an array of permitted scalar values

Nested Parameters

```
params.permit(
    :name,
    { :emails => []},
    :friends => [ :name, { :family => [ :name ] }]
)
```

ProtectedAttributes

- attr_accessible / attr_proteced removed
- https://github.com/rails/protected_attributes

Use it today

- https://github.com/rails/strong_parameters
- Rails 3.x
- include ActiveModel::ForbiddenAttributesProtection

Turbolinks

Makes following links faster

Live Demo

Turbolinks

- Activated by default
- Doesn’t reinterpret JavaScript and CSS each time the page loads
- Similar to pjax
- Use it today!
  https://github.com/rails/turbolinks

Russian Doll Caching

Cache Digests

‘Definition’

The technique of nesting fragment caches to maximize cache hits is known as russian doll caching.

Example Model

```
class Team < ActiveRecord::Base
  has_many :members
end

class Member < ActiveRecord::Base
  belongs_to :team, touch: true
end
```

Example Model

```
class Team < ActiveRecord::Base
  has_many :members
end

class Member < ActiveRecord::Base
  belongs_to :team, touch: true
end
```

Example Template

```
<!-- app/views/teams/show.html.erb -->
<% cache @team do%>
  <h1>Team: <%= @team.name %></h1>

  <%= render @team.members %>
<% end %>

<!-- app/views/members/_member.html.erb -->
<% cache member do %>
  <div class='member'>
    <%= member.name %>
    <p><%= member.bio %></p>
  </div>
<% end %>
```

Fragment Caches Example

- views/members/1-20121220141922
- views/members/2-20121220141922
- views/teams/2-20121220141922

Versioning

```
<!-- app/views/teams/show.html.erb -->
<% cache ["v1", @team] do%>
  <h1>Team: <%= @team.name %></h1>

  <%= render @team.members %>
<% end %>

<!-- app/views/members/_member.html.erb -->
<% cache ["v1", member] do %>
  <div class='member'>
    <span><%= member.name %></span>
    <p><%= member.bio %></p>
  </div>
<% end %>
```

Fragment Caches Example

- views/v1/members/1-20121220141922
- views/v1/members/2-20121220141922
- views/v1/teams/2-20121220141922

Cache Digests

- Rails 4 will suffix a digest of the template and its dependencies
- Don’t worry about fragment cache dependencies and versioning!

Fragment Caches Example

- views/members/1-20121220141922/74865fcb3e2752a0928fa4f89b3e4426
- views/members/2-20121220141922/74865fcb3e2752a0928fa4f89b3e4426
- views/teams/2-20121220141922/4277f85c137009873c093088ef609e60

Rake Tasks

```
$ rake cache_digests:dependencies \
                        TEMPLATE=projects/show

[
  “teams/show”,
  “members/member”
]

$ rake cache_digests:nested_dependencies \
                        TEMPLATE=projects/show
```

Caching Deprecations

- Action and Page caching has been extracted
- https://github.com/rails/actionpack-action_caching
- https://github.com/rails/actionpack-page_caching

HTTP PATCH Verb

for partial updates

RFC 5789

"In a PUT request, the enclosed entity is considered to be a modified version of the resource stored on the origin server, and the client is requesting that the stored version be replaced. With PATCH, however, the enclosed entity contains a set of instructions describing how a resource currently residing on the origin server should be modified to produce a new version."

Changes

- PATCH is going to be the primary method for updates
- will be backward compatible
- PUT requests get routed to update as they are today
- Forms will set the hidden field _method to :patch on edits

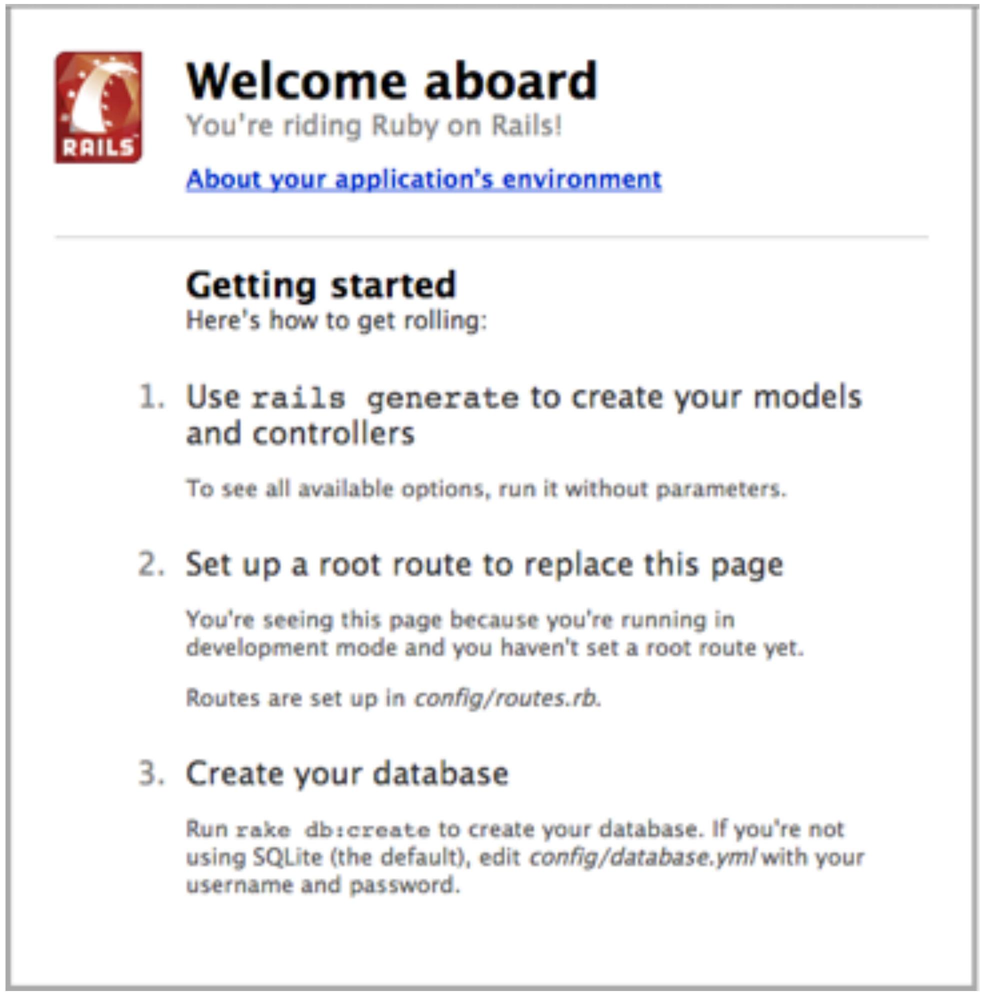
Railties

Gluing the Engine to the Rails

Rails::Plugin has gone

- vendor/plugins is no more
- use gems or
- bundler with path or git dependencies

Dynamic index.html



Welcome aboard

You're riding Ruby on Rails!

About your application's environment

Getting started

Here's how to get rolling:

1. Use `rails generate` to create your models and controllers

   To see all available options, run it without parameters.

2. Set up a root route to replace this page

   You're seeing this page because you're running in development mode and you haven't set a root route yet.

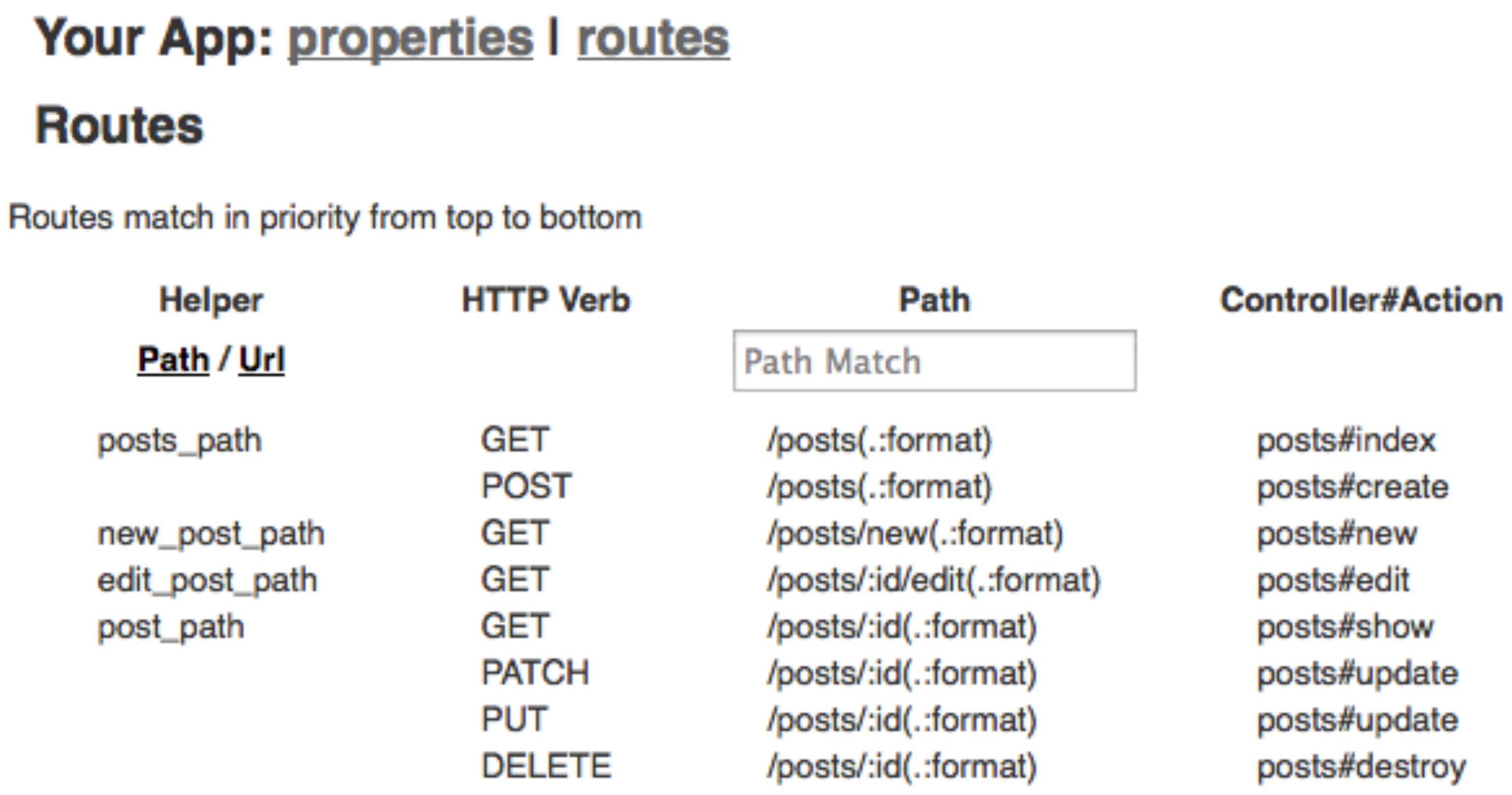
   Routes are set up in *config/routes.rb*.

3. Create your database

   Run `rake db:create` to create your database. If you're not using SQLite (the default), edit *config/database.yml* with your username and password.

- Rails::WelcomeController serves dynamic welcome screen
- Disable by defining root route in routes.rb

rails/info/routes

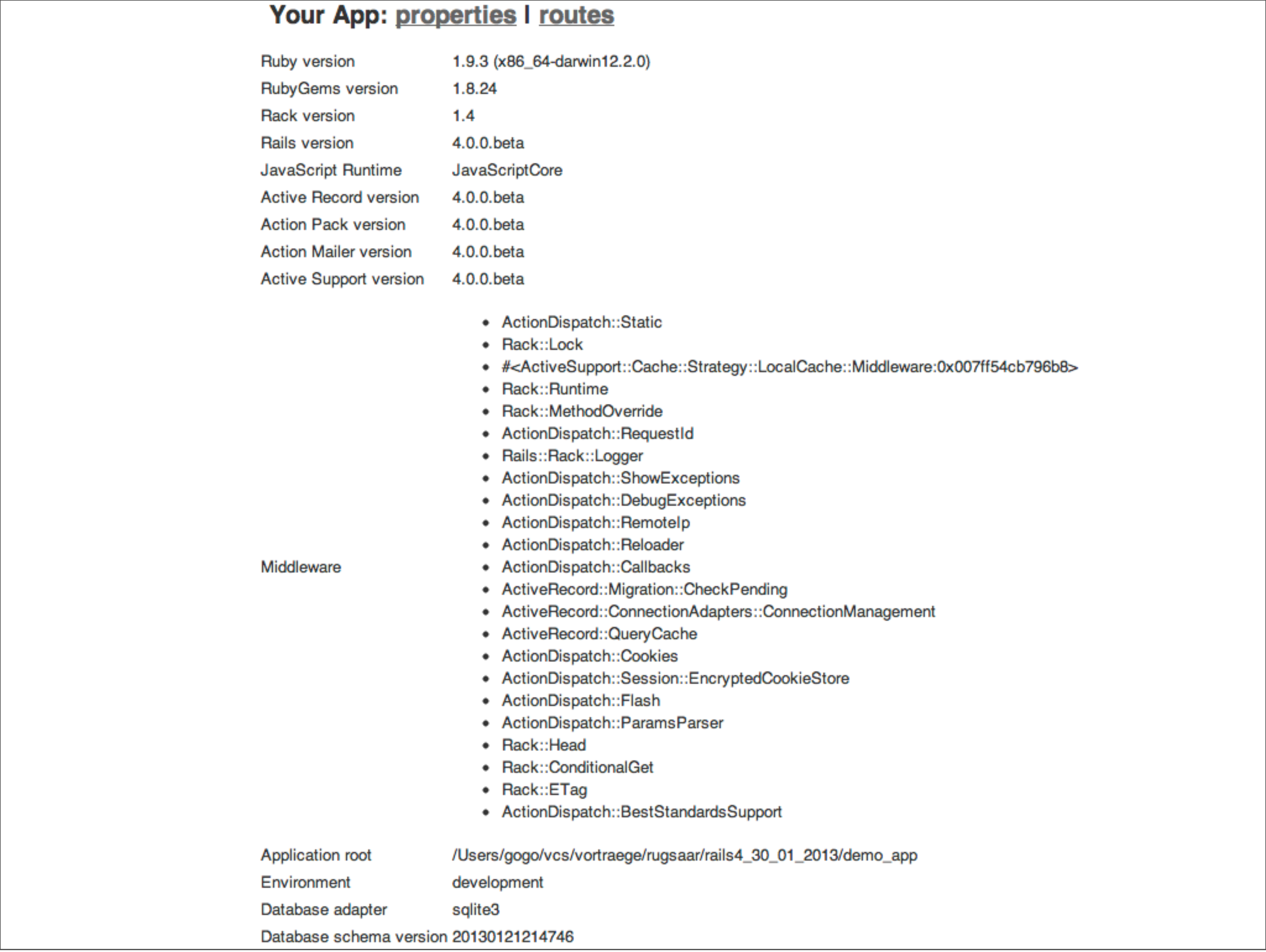


Your App: properties | routes

Routes

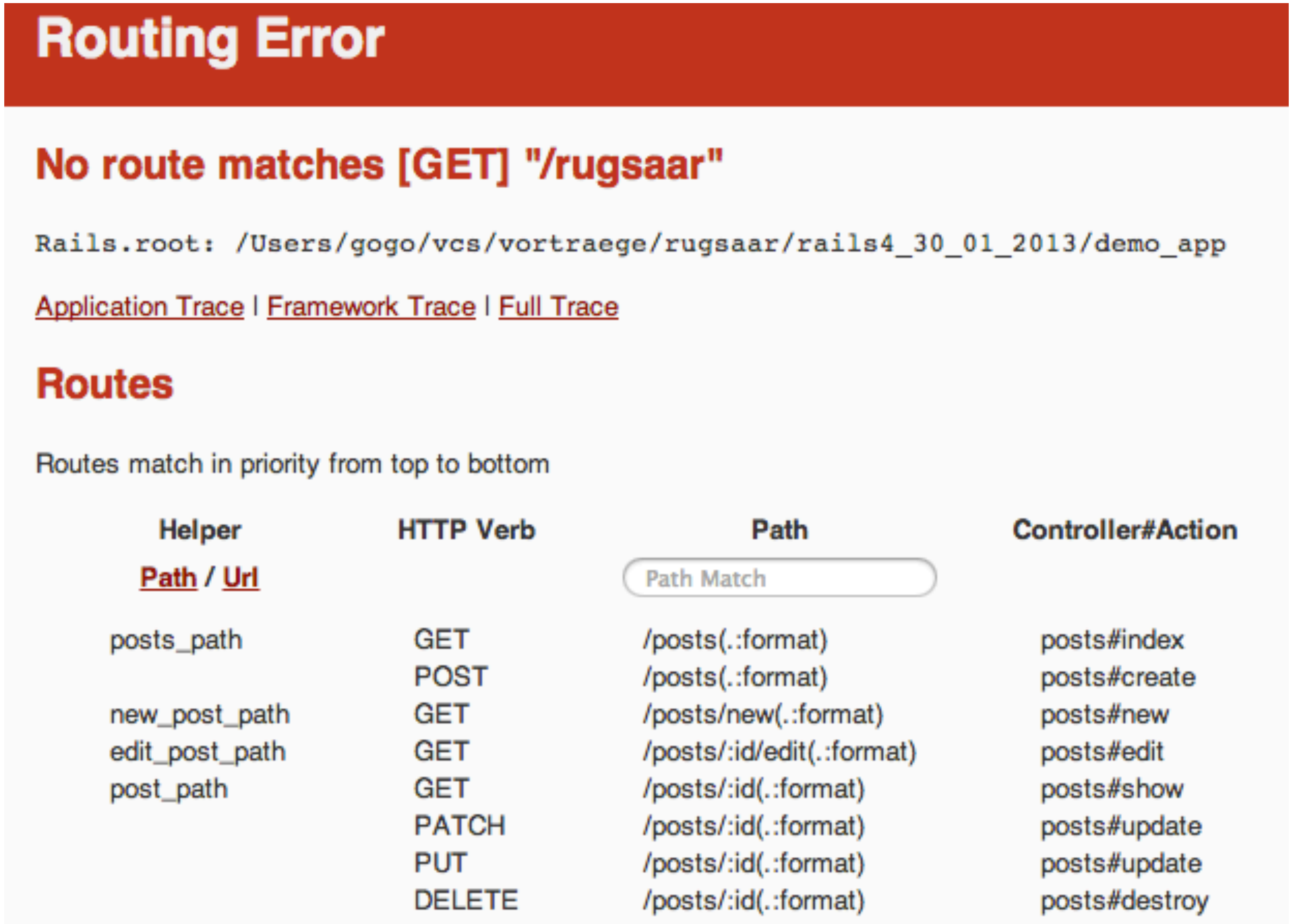
Routes match in priority from top to bottom

Helper	HTTP Verb	Path	Controller#Action
Path / Url		Path Match	
posts_path	GET	/posts(.:format)	posts#index
	POST	/posts(.:format)	posts#create
new_post_path	GET	/posts/new(.:format)	posts#new
edit_post_path	GET	/posts/:id/edit(.:format)	posts#edit
post_path	GET	/posts/:id(.:format)	posts#show
	PATCH	/posts/:id(.:format)	posts#update
	PUT	/posts/:id(.:format)	posts#update
	DELETE	/posts/:id(.:format)	posts#destroy



Your App: properties | routes

Ruby version	1.9.3 (x86_64-darwin12.2.0)
RubyGems version	1.8.24
Rack version	1.4
Rails version	4.0.0.beta
JavaScript Runtime	JavaScriptCore
Active Record version	4.0.0.beta
Action Pack version	4.0.0.beta
Action Mailer version	4.0.0.beta
Active Support version	4.0.0.beta
Middleware	• ActionDispatch::Static • Rack::Lock • #<ActiveSupport::Cache::Strategy::LocalCache::Middleware:0x007ff54cb796b8> • Rack::Runtime • Rack::MethodOverride • ActionDispatch::RequestId • Rails::Rack::Logger • ActionDispatch::ShowExceptions • ActionDispatch::DebugExceptions • ActionDispatch::RemoteIp • ActionDispatch::Reloader • ActionDispatch::Callbacks • ActiveRecord::Migration::CheckPending • ActiveRecord::ConnectionAdapters::ConnectionManagement • ActiveRecord::QueryCache • ActionDispatch::Cookies • ActionDispatch::Session::EncryptedCookieStore • ActionDispatch::Flash • ActionDispatch::ParamsParser • Rack::Head • Rack::ConditionalGet • Rack::ETag • ActionDispatch::BestStandardsSupport
Application root	/Users/gogo/vcs/vortraege/rugsaar/rails4_30_01_2013/demo_app
Environment	development
Database adapter	sqlite3
Database schema version	20130121214746



Routing Error

No route matches [GET] "/rugsaar"

```
Rails.root: /Users/gogo/vcs/vortraege/rugsaar/rails4_30_01_2013/demo_app
```

Application Trace | Framework Trace | Full Trace

Routes

Routes match in priority from top to bottom

Helper	HTTP Verb	Path	Controller#Action
Path / Url		Path Match	
posts_path	GET	/posts(.:format)	posts#index
	POST	/posts(.:format)	posts#create
new_post_path	GET	/posts/new(.:format)	posts#new
edit_post_path	GET	/posts/:id/edit(.:format)	posts#edit
post_path	GET	/posts/:id(.:format)	posts#show
	PATCH	/posts/:id(.:format)	posts#update
	PUT	/posts/:id(.:format)	posts#update
	DELETE	/posts/:id(.:format)	posts#destroy

Action Controller

Rename all action callbacks from *_filter to *_action

```
class PeopleController < ActionController::Base
  before_action :set_person, except: [ :index, :new, :create ]
  before_action :ensure_permission, :only: [ :edit, :update ]

  # ...

  private

  def set_person
    @person = current_account.people.find(params[:id])
  end

  def ensure_permission
    current_person.can_change?(@person)
  end
end
```

Register Your Own Flash Types

```
# Rails 3
class PostsController
  def create
    ...
    flash[:error] = “my error”
    redirect_to home_path
  end
end

# app/views/home/index
<%= flash[:error] %>
```

Register Your Own Flash Types

```
# Rails 4
class ApplicationController
  add_flash_types :error, :warning
end

class PostsController
  def create
    ...
    redirect_to home_path, error: “my error”
  end
end

# app/views/home/index
<%= error %>
```

Action Dispatch

Root

```
# config/routes.rb    Rails 3
root :to => ‘pages#main’

# config/routes.rb    Rails 4
root ‘pages#main’
```

Unicode Characters in Routes

```
# config/routes.rb    Rails 3
get Rack::Utils.escape("Fußball") => 'sports#football'

# config/routes.rb    Rails 4
get ‘Fußball’ => ‘sports#football’
```

Routing Concerns

```
# config/routes.rb    Rails 3
resources :messages do
  resources :comments
end

resources :posts do
  resources :comments
  resources :images, only: index
end
```

Routing Concerns

```
# config/routes.rb    Rails 4
concern :commentable do
  resources :comments
end

concern :image_attachable do
  resources :images, only: index
end

resources :messages, concerns: :commentable
resources :posts, concerns:
                                  [:commentable, :image_attachable]
```

Match

```
match '/about' => 'generals#about', as: :about
```

- old DSL will match all verbs for the path to the specified endpoint, use :via => :all
- most of the time you actually mean +get+
- use HTTP verbs!

```
get '/about' => 'generals#about', as: :about
```

Active Record

No More Rails 2 Finder Syntax

```
User.find(:all, ...)
User.find(:first, ...)
User.find(:last, ...)
```

Model.all returns relation

```
# Rails 3
> Post.all.where(title: “RUGSAAR”)
=> NoMethodError: undefined method `where' for []:Array

# Rails 4
> Post.all.where(title: “RUGSAAR”).to_a
```

NOT

```
Article.where("title != ?", title)
```

```
Article.where.not(title: 'Rails 3')
# >> SELECT "articles".* FROM "articles" WHERE
("articles"."title" != 'Rails 3')

Article.where.not(title: ['Rails 3', 'Rails 5'])
# >> SELECT "articles".* FROM "articles" WHERE
("articles"."title" NOT IN ('Rails 3', 'Rails 5'))
```

Whiny nil

```
# Rails 3
> nil.id
=> RuntimeError: Called id for nil, which would mistakenly be
4 -- if you really wanted the id of nil, use object_id

# Rails 4
> nil.id
=> NoMethodError: undefined method ‘id’ for nil:nilClass
```

.first and .last ordering

```
# Rails 3
> Post.first
SELECT “posts”.* FROM “posts” LIMIT 1
```

```
# Rails 4
> Post.first
SELECT “posts”.* FROM “posts”
  ORDER BY “posts”.”id” ASC LIMIT 1
```

Eager-evaluated scopes

```
# Deprecated
scope :published,
      where(:published => true)

# Pitfall
scope :recent,
      where(“created_at > ?”, 1.week.ago)

# use a lamda (always, not just for time!)
scope :published,
      -> { where(:published => true) }
```

update_(attribute|column)s?

	Validations run?	Callbacks run?	updated_at touched?	Dirty attributes updated?
update_attributes(:name => “value”)	Yes	Yes	Yes	Yes
update_attribute(:name,“value”)	No	Yes	Yes	Yes
★ update_columns(:name => “values”)	No	No	No	No
update_column(:name,“value”)	No	No	No	No

update_attributes => update

Dynamic Finders

find_by_. . . find_by_....!	OK
find_all_by_. . . scoped_by_....	where(. . .)
find_last_by_. . .	where(. . .).last
find_or_initialize_by_. . .	where(. . .). first_or_initialize
find_or_create_by_. . .	where(. . .). first_or_create
find_or_create_by_. . .!	where(. . .). first_or_create!

find_by and find_by!

```
Post.find_by_name_and_rating('Spartacus', 4)
```

```
Post.find_by name: 'Spartacus', rating: 4
```

```
Post.find_by! name: 'Spartacus'
```

:-(

- Binstubs
- ActionView (new html5 and collection form helpers, ...)
- MiniTest
- PostgreSQL (hstore, array, date ranges, ...)
- ...

Danke

http://blog.wyeworks.com/2012/11/13/rails-4-compilation-links/

Danke

http://blog.wyeworks.com/2012/11/13/rails-4-compilation-links/