software studio

an overview of Rails

Daniel Jackson

what is Rails?

an application framework

› full stack: web server, actions, database

a programming environment

› eg, rake (like make), unit testing

an open-source community

› many plugins

history of Rails

genesis in Basecamp

› project management tool by 37signals

release

› open source in 2004

› shipped with OS X 10.5 in 2007

› Rails 3.1 in 2011, merging with Merb

Screenshot of David Heinemeier's bio removed due to copyright restrictions.
See his website for the image.

Rails's creator, describing himself with the typical modesty of the nouveau riche

convention over configuration

the key idea behind Rails

› database table: users
› model class: User
› file: /app/models/user.rb

what's the alternative?

› configuration files

technical features

Rails supports

- JQuery as standard JS library
- Sass for CSS templating
- ERB or HAML for HTML templating
- MySQL or PostgreSQL database

support for Rails from

- web servers such as Apache
- hosting services such as Heroku

model-driven development

if you type this...

```
$ rails generate scaffold Post name:string title:string content:text
```

Rails generates these...

File	Purpose
db/migrate/20100207214725_create_posts.rb	Migration to create the posts table in your database (your name will include a different timestamp)
app/models/post.rb	The Post model
test/unit/post_test.rb	Unit testing harness for the posts model
test/fixtures/posts.yml	Sample posts for use in testing
config/routes.rb	Edited to include routing information for posts
app/controllers/posts_controller.rb	The Posts controller
app/views/posts/index.html.erb	A view to display an index of all posts
app/views/posts/edit.html.erb	A view to edit an existing post
app/views/posts/show.html.erb	A view to display a single post
app/views/posts/new.html.erb	A view to create a new post
app/views/posts/_form.html.erb	A partial to control the overall look and feel of the form used in edit and new views
test/functional/posts_controller_test.rb	Functional testing harness for the posts controller
app/helpers/posts_helper.rb	Helper functions to be used from the post views
test/unit/helpers/posts_helper_test.rb	Unit testing harness for the posts helper
app/assets/javascripts/posts.js.coffee	CoffeeScript for the posts controller
app/assets/stylesheets/posts.css.scss	Cascading style sheet for the posts controller
app/assets/stylesheets/scaffolds.css.scss	Cascading style sheet to make the scaffolded views look better

Rails snags: reliance on strings

conventions rely on strings used for names

› pluralization: cute but ultimately painful
› name munging, eg for path helpers

alpha equivalence fails in Rails

› in lambda calculus, λx.x = λy.y
› "rename variable" refactoring

if you code this route:

```
resources :posts, :path => "/admin/posts"
```

you get these functions:

HTTP Verb	Path	action	named helper
GET	/admin/posts	index	posts_path
GET	/admin/posts/new	new	new_post_path

`pluralize``(word)` *public*

Returns the plural form of the word in the string.

Examples:

```
"post".pluralize             # => "posts"
"octopus".pluralize          # => "octopi"
"sheep".pluralize            # => "sheep"
"words".pluralize            # => "words"
"CamelOctopus".pluralize     # => "CamelOctopi"
```

Original question asked by Donald Hughes on Stack Overflow.

stackoverflow Questions Tags Users Badges Unanswered

Search

rails pluralization search

804 results

relevance newest votes active

10 votes

1 answer

Q: Ruby on Rails ActiveRecord: pluralization

I'm new to **Rails**, so forgive my ignorance of ActiveRecord. One of my models is named "campus". I ran the migration and it **pluralized** everything except "campus". I thought that was lame so I … to "campus" no longer work. I ran it through the console and noticed that I'm getting an uninitialized constant "Campu". So something still thinks "campus" is **plural**? Should I assume that config change will cause me nothing but trouble going forward? …

ruby-on-rails ruby activerecord model pluralize

asked feb 28 '10 by Donald Hughes

1

I looked around on Stack Overflow and Agile Development with Rails but couldn't find anything that answered all the parts of this I need.

I just generated a Cow model in rails. Apparently, Rails uses an antiquated plural of cow ("kine"), so when I created that model, it built a Kine migration:

```
class CreateKine < ActiveRecord::Migration
  def change
    create_table :kine do |t|
      t.string :name
      t.string :farm
      t.string :breed

      t.timestamps
    end
  end
end
```

I know I could go into the model's .rb file and set_table_name back to cow, but I'm worried about associated controllers. If I create a Cows controller, will it not sync up?

How do I get everything to be Cow/Cows? Thanks. This is one of my first apps, and I'm already way confused by managing controller-model associations, so this inflection issue doesn't help.

ruby-on-rails | ruby-on-rails-3 | controller | migration | inflection

Original question asked by Sasha on Stack Overflow.

2

1

I am struggling with the pluralization of the RESTful route generation in Rails 2.3.2.

Specifically, I have a resource called `sitestatus`. This resource really is uncountable (deer is deer, not deers). When I specify it as uncountable in an intializer, I get some helpers, but the `sitestatuses_path` is unavailable (which would make sense).

So, in a gesture to conformity, I have allowed `sitestatus` to be countable. So now, Rails pluralizes `sitestatus` to `sitestatuses` (not too horrible), but it insists on *also* singularizing it to `sitestatu` (missing the 's', hilarious and horrible at the same time).

So, I whipped out my bigger hammer and added this code to the intializer:

```
ActiveSupport::Inflector.inflections do |inflect|
  inflect.plural "sitestatus", "sitestatuses"
  inflect.singular "sitestatus", "sitestatus"
end
```

(Note: I tried using `irregular` and it didn't work right)

Doing this gives me the expected results in the console when I `"sitestatus".pluralize`, but when I attempt to make a call to `sitestatuses_path` in my view I get

```
undefined local variable or method 'sitestatuses_path'
```

When I load up ActionController::UrlHelper in the console and call `sitestatus_path(123)` I get `sitestatus/123` as I would expect. However, when I call `sitestatuses_path` I get

```
undefined method 'sitestatuses_path' for #<Object...
```

This name is the name of the model and the controller and it really is the only logical name for both as it lines up with the business name for the object perfectly.

What am I missing?

Rails snags: too much magic

implicit calls

› bad for non-experts
› bad for tools

missing specs

› not clear what's going on
› magic changes over time

an example

› which fields in forms are logged?
› next slide...

```ruby
def show
  @post = Post.find(params[:id])

  respond_to do |format|
    format.html  # show.html.erb
    format.json  { render :json => @post }
  end
end
```

PUBLIC rails / rails

Watch ▾ | Star 17,459 | Fork 4,923

Code | Network | Pull Requests 178 | Issues 433 | Graphs

Closed **jeyb** wants to merge 1 commit into `rails:master` from `jeyb:filter_passwo…` 4 #6988

Discussion | Commits 1 | Files Changed 2

jeyb opened this pull request 7 months ago

Filter password_confirmation param from being logged.

No one is assigned — No milestone

Most common Rails authentication libraries use password_confirmation field on registration which should be excluded by default alongside the password field.

Closed

+ 2 additions

- 2 deletions

2 participants

jeyb added a commit — 7 months ago

jeyb Filter password_confirmation param from being logged. … a5fe8ae

josevalim commented — 7 months ago

When you pass :password, it ignores all fields that has the word password in it, so you don't need :password_confirmation.

Closed  **josevalim** closed the pull request 7 months ago

Pull request by jeyb on GitHub.

in summary...

rich environment
many libraries
code generation
helpful community
friendly online guides

invisible magic
quirky conventions
no static checking
masking of failures

an easy life?

or a
deadly
cocktail?

actually, neither

every tool has benefits & limitations

› just need to recognize them
› & work around the limitations

also, realize that context matters

Rails is great for

› rapid development
› data intensive apps with rich UIs

not so suitable for

› critical systems (eg, banking)
› specialized data (eg, web searching)

6.170 Software Studio
Spring 2013