A First Post
by Chris Stearns - Wednesday, 1 March 2006, 12:16 PM

Anyone who wants a specific question, topic, or issue addressed regarding Stacks on the Tracks should feel free to post it, or email me, and I'll do my best to cover it here as well as in the live session.

Re: A First Post
by Lauren Fancher - Thursday, 16 March 2006, 09:28 AM

Do you have examples of how you are using this that we can look at before hand on your library website?

Re: A First Post
by Chris Stearns - Tuesday, 21 March 2006, 09:37 AM

Hi Lauren,

Sorry for the late reply...been out of the office.

We have a few small-medium public applications running on the public web at the moment:

1. A basic hours and info page... http://www.lib.auburn.edu/hours
2. A personnel directory... http://www.lib.auburn.edu/people
3. A professional training scheduler (unfortunately, most of this is password protected, for AU folks only, but you can get the gist of here ...
   http://www.lib.auburn.edu/training

   A recent university-wide redesign has caused some minor rendering glitches in IE, but nothing serious.

   We've also got some computer lab stats tracking software and a couple of little things running on an internal network. I'll be discussing some of that at the presentation on Friday.
Fri1B

0.133868 secs
Included 23 files
ticks: 13 user: 10 sys: 2 cuser: 0 csys: 0
Load average: 0.33
A few resources
by Chris Stearns - Wednesday, 22 March 2006, 08:15 AM

Here are a few resources that you may find interesting/useful:

This is an article/blog post describing the process of building a Rails app to run a website and serve a large amount of content. Being a newspaper site, I think it parallels (sometimes closely/sometimes not) the core services libraries tend to provide online.
http://brainspl.at/articles/2005/11/03/from-start-to-launch-http-yakimaherald-com

A video of an interesting Rails presentation by a Ruby expert.
http://www.rubynuby.org/pages/downloads

Some screencasts that mainly provide a look at the basics of getting started with a Rails app, and how they work.
http://rubyonrails.com/screencasts

Re: A few resources
by Lauren Fancher - Wednesday, 22 March 2006, 08:59 AM

Cool -- thanks!
Included 23 files
  ticks: 13 user: 11 sys: 2 cuser: 0 csys: 0
  Load average: 0.39
Stacks on the Tracks demo follow-up

Re: NoMethodError at the end of the demo

What happened:

The application choked when I added the following code to the projects/list view:

<%= project.teacher.name %>

The reason:

A silly one...if you remember, I created the database tables as distinct entities then tied them together using has_many in the Teacher model and belongs_to in the Project model.

That worked fine. But when I pulled up the projects/edit screen to show how the teacher we added (me) now appeared as a select option, I neglected to save it to the database, so when I added <%= project.teacher.name %> to projects/list, it attempted to find the teacher's name using the teacher_id value in projects.teacher_id, which happened to be NULL. So it choked on the NULL value in the teacher_id column in the projects table. And so it goes.

Thanks again.

Chris Stearns
AU Libraries
stearc@auburn.edu
334.844.1747
Fri1B

0.131447 secs
Included 23 files
ticks: 13 user: 14 sys: 0 cuser: 0 csys: 0
Load average: 0.41
Short List of Ruby/RubyOnRails resources

Books – Print & Electronic

*Programming Ruby*
by Dave Thomas with Chad Fowler and Andy Hunt
Pragmatic Programmers

*Agile Web Development with Rails*
by Dave Thomas and David Heinemeier Hansson
Pragmatic Programmers

*Ruby for Rails ( May 2006 )*
by David A. Black
Manning Publications

*Ruby in a Nutshell*
by Yukihiro "Matz" Matsumoto
O'Reilly

*Rails Cookbook ( August 2006 )*
by Rob Orsini
O'Reilly

*Programming Rails ( ??? )*
by Robby Russell
O'Reilly

*Rails Recipes ( May 2006 )*
by Chad Fowler
Pragmatic Programmers

*Ruby on Rails: Up and Running ( May 2006 )*
Bruce Tate and Curt Hibbs
O'Reilly

*Ruby Cookbook ( September 2006 )*
Leonard Richardson and Lucas Carlson
O'Reilly

Online Resources

Rails homepage
http://rubyonrails.com

Building and launching a large-scale content-driven application
http://brainspl.at/articles/2005/11/03/from-start-to-launch-http-yakimaherald-com

Software and Hardware setups at Robot Co-op
http://blog.segment7.net/articles/2006/03/20/robot-co-op-software
http://blog.segment7.net/articles/2006/03/15/robot-co-op-hardware

Scaling @ robot co-op
http://blog.segment7.net/articles/2006/03/06/2-5-million

Scaling @ eins.de (social networking site), 1st in series of articles
http://poocs.net/articles/2006/03/13/the-adventures-of-scaling-stage-1

compiled by Chris Stearns, Auburn University Libraries, stearcs@auburn.edu, 334.844.1747
Additional Resources
by Chris Stearns - Friday, 24 March 2006, 08:32 AM

If this works, you should find a .pdf attached with some books on Ruby/Rails and some additional info online that addresses real-world scaling, deployment, and hardware/software setups.

Re: Additional Resources
by Chris Stearns - Saturday, 25 March 2006, 07:51 PM

Here's one more great book for beefing up programming skills while learning Ruby:

Learn to Program
by Chris Pine
Pragmatic Programmers (2006)
Referential Integrity Question from the Presentation

by Chris Stearns - Friday, 7 April 2006, 09:06 AM

Toward the end of the discussion, someone asked a question about how Rails handles referential integrity among database tables. I mentioned that Rails did a pretty good job of handling this for you, but didn't go into detail (we were running late as it was).

I found myself in an unusual Friday lull here at work, so just in case anyone is still reading this stuff, I thought I'd go into a little more detail.

One simple way to handle this is to use the :dependent option in your model, which tells ActiveRecord (Rails' object-relational mapping module) that a child table's rows cannot exist if the row represented by the foreign key doesn't also exist in the parent table...so, if you delete the parent row, the corresponding child rows will also get deleted. Here's how that works:

To use our example from the presentation, we'll have two models...the Teacher model (parent) and the Project model(child), so a Project belongs_to:

class Project < ActiveRecord::Base
  belongs_to :teacher
end

AND

class Teacher < ActiveRecord::Base
  has_many :projects
end

To enforce simple referential integrity, we can add the :dependent => true option to the has_many declaration for a particular child table in the Teacher (parent) model like this:

class Teacher < ActiveRecord::Base
  has_many :projects, :dependent => 'true'
end

This tells ActiveRecord that projects table rows owe their very existence to their teacher row, that any child rows containing a foreign key reference to the teacher row (that belong_to: teacher) should be deleted if that teacher is deleted.

Anyhow, I hope that gives the person who asked this question a clearer picture of how ActiveRecord handles this sort of thing.
Fri1B

0.130565 secs
Included 23 files
ticks: 13 user: 10 sys: 3 cuser: 0 csys: 0
Load average: 0.18