Why Ruby On Rails?

+

Aaron Bartell
aaron@powerruby.com

PowerRuby

Copyright 2014 PowerRuby, Inc.

There's something special going on with Ruby and Rails. Never before has there been such coordinated community efforts to efficiently produce reusable code (aka gems) in such open and social fashions. Coding is fun again. We want to introduce that reality to the IBM i platform.

Ruby… a dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write.

Matz desired a language which he himself enjoyed using, by minimizing programmer work and possible confusion - enter Ruby.

Hello World sample program:

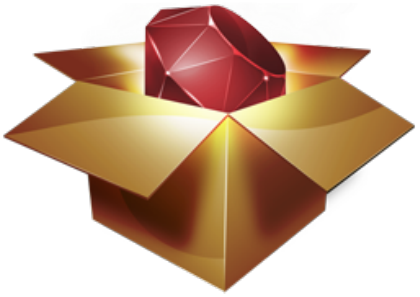
```
-bash-4.2$ cat hello_world.rb
puts "Hello World"
-bash-4.2$ ruby hello_world.rb
Hello World
-bash-4.2$
```

- Est 1995 by **Yukihiro "Matz" Matsumoto**
- Mass acceptance 2006
- Very active and well organized community

Features:

- variable declarations are unnecessary
- variables are not statically typed
- syntax is simple and consistent
- everything is an object
- classes, methods, inheritance, etc.

ruby-lang.org – Home website
ruby-doc.org/docs/ProgrammingRuby - Traditional learning
codecademy.com - Learn Ruby in the browser for free
amzn.to/1apcrse - Metaprogramming Ruby: Program Like the Ruby Pros

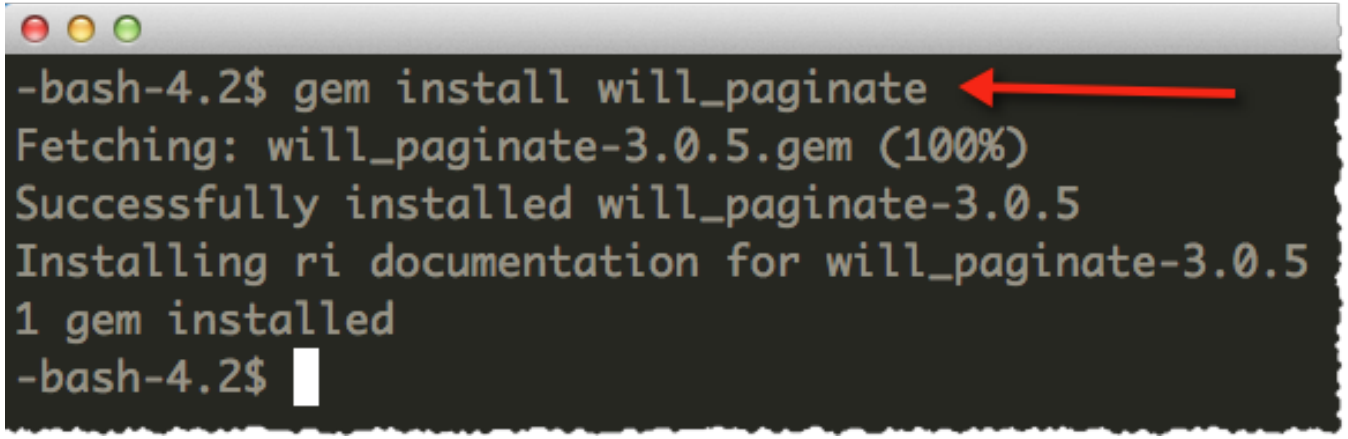


RubyGems… *simplify the process of installing, removing, updating and managing Ruby libraries and their dependencies.*

Ruby is a great language, but the Ruby community wanted to start modularizing and sharing code - enter RubyGems.

- Est 2001
- server for hosting and distribution of gems, rubygems.org
- manage gem dependencies
- manage multiple versions of the same library easily
- included in Ruby as of 1.9.3
- Rails and Bundler are gem themselves.

> *"There's a gem for that"*
> - every Ruby business programmer



rubygems.org - Home website
linuxjournal.com/article/8967 – RubyGem history

Bundler maintains a consistent environment for ruby applications. It tracks an application's code and the rubygems it needs to run, so that an application will always have the exact gems (and versions) that it needs to run.

RubyGems made sharing of code so easy that organizing them soon became hard to manage for an application maintainer - enter Bundler.

`myapp/Gemfile` contents:

```
source 'https://rubygems.org'

gem 'rails', '4.0.0'
gem 'ibm_db'
gem 'jquery-rails'
gem 'turbolinks'
gem 'jbuilder', '~> 1.2'
gem 'thin'
```

- Est 2008 by Yehuda Katz
- Application dependencies declared in *Gemfile*
- Command `bundle install` downloads all gems in Gemfile and also their dependencies.
- Included in Rails as of 3.0

bundler.io - Home website
railscasts.com/episodes/201-bundler-revised – Video tutorial
yehudakatz.com/2010/12/16/clarifying-the-roles-of-the-gemspec-and-gemfile

"...Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency."

Invented by Linus Torvalds (creator of Linux), in 2004, maintained at git-scm.com, intro video here.

Frictionless Context Switching. Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.

Feature Based Workflow. Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.

Disposable Experimentation. Create a branch to experiment in, realize it's not going to work, and just delete it - abandoning the work—with nobody else ever seeing it (even if you've pushed other branches in the meantime).

Try Git from the browser: try.github.io

Git is included with PowerRuby's IBM i install.

*GitHub is a web-based hosting service for software development projects that use the Git revision control system. Their **fork** and **pull request** features have revolutionized social coding.*

Fork - Desire to contribute to someone else's project, or would like to use someone's project as the starting point for your own.

Pull Request - Pull requests let you tell others about changes you've pushed to a GitHub repository. Once a pull request is sent, interested parties can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.

Social coding:

- invites more people to the table
- many developers providing input from many industries
- brings open source projects to fruition faster
- eliminate bugs sooner
- Ruby and Rails community makes extensive use of GitHub

github.com - Home site
youtube.com/watch?v=-CpIlxXPsvw - intro to Forks and Pull Requests
youtube.com/user/GitHubGuides - best place to learn

… an open-source web framework written in the Ruby language that's optimized for programmer happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration.

Programming 5250 green screens with RPG was very efficient. Ruby On Rails (aka Rails) accomplishes the same, but in the modern web world.

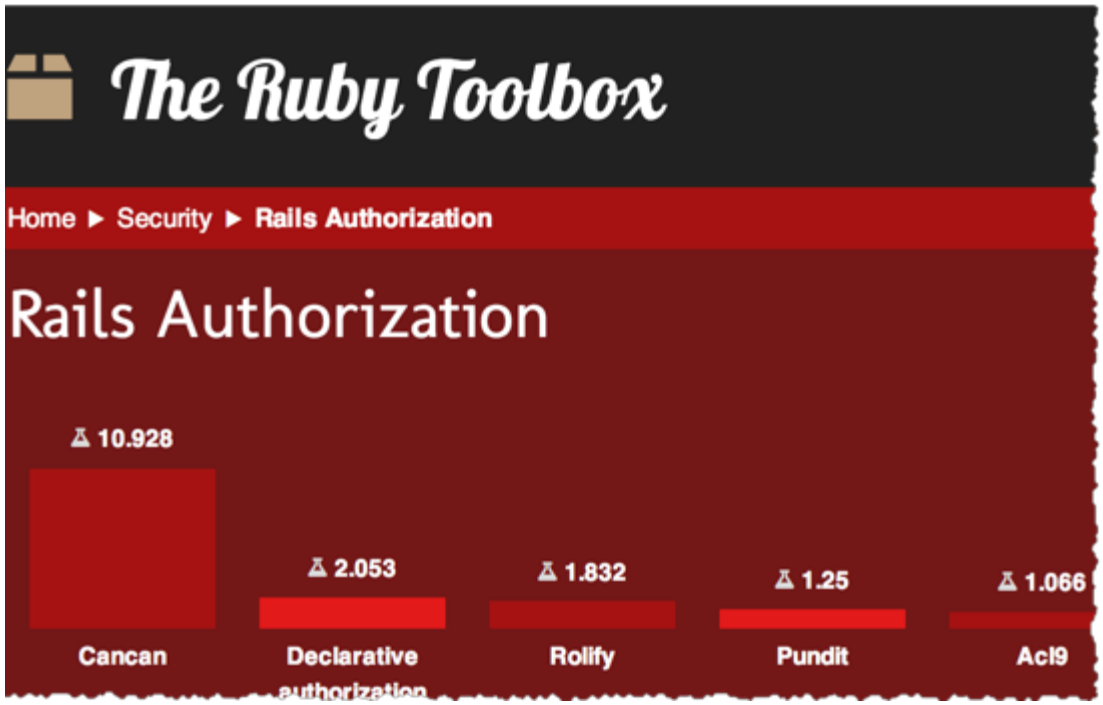
- **Est 2004 by David Heinemeier Hansson**, in 2006 Apple ships with OSX 10.5 "Leopard"
- **Separation of concern** (model-view-controller, "helpers", mailers, presenters)
- **Generate application components** with custom gems and configurations.
- **Multiple environments** (i.e. dev, test, prod, <custom>)
- **ActiveRecord** pattern
- i18n features (**internationalization**) by default when using `rails` command.
- Includes Rake, a **task management** tool (i.e. rake db:migrate)
- **Unit testing**
- **Asset pipeline**

Really, too many features and benefits to list.

RubyOnRails.org – Framework home
Guides.RubyOnRails.org/getting_started.html – Quick learning, formal documentation
RailsCasts.com - Excellent video resource for learning about all things Rails
TeamTreehouse.com - Comprehensive Ruby/Rails/HTML/CSS/Javascript training

"Rails", "Ruby on Rails", and the Rails logo are registered trademarks of David Heinemeier Hansson. All rights reserved.

Many open source choices



With the crazy amount of community participation to projects that might do the same or similar things, you might be wondering how to make the right pick.

Start at the ruby-toolbox.com

The Ruby Toolbox is a comprehensive catalog of Ruby and Rails plug-ins, gems, tools and resources for Ruby developers with popularity ratings based on Github watchers and Gem downloads

Notable people with quotes

"Rails is the most well thought-out web development framework I've ever used. And that's in a decade of doing web applications for a living. I've built my own frameworks, helped develop the Servlet API, and have created more than a few web servers from scratch. Nobody has done it like this before."

-James Duncan Davidson, Creator of Apache Tomcat and Ant

"After researching the market, Ruby on Rails stood out as the best choice. We have been very happy with that decision. We will continue building on Rails and consider it a key business advantage."

-Evan Williams, Creator of Blogger, ODEO, and Twitter

"Having scaled multiple rails sites, a couple to millions and another to billions of PV/month, the runtime performance of the interpreter is never the slowest. Scaling a site is about architecture, databases, caching, event queues, disk IO, CDNs, etc. With a few exceptions, the runtime or framework doesn't crack the top 5. Most important of all, Rails is the most efficient framework right now to build your app quickly and effectively. Stop worrying and start building. Less than 1% of sites on the internet get to enough traffic to even care about performance.... but if you have performance problems, rejoice! That means you built something people use!"

- Aaron Batalion, CTO & Co-Founder, LivingSocial

Why is RubyOnRails Hot? **BusinessInsider** explains well…
businessinsider.com/heres-why-ruby-on-rails-is-hot-2011-5

*

*

Who uses Ruby On Rails?**

**builtwith.com and rubyonrails.org used for references

Open Source Rails Websites

spree
(spreecommerce.com)

refinery
content management system

Discourse

opensourcerails.com
your resource for the most popular and
up to date open source rails based sites.

REDMINE
Project Management

Clocking IT
TimeTracking 2.0

GitLab.org

Fat Free CRM

Amahi

BrowserCMS 2.0
BROWSER-BASED CONTENT MANAGEMENT SYSTEM

jekyll

- What is PowerRuby?
  - **Freely** available and **commercially supported** port of the Ruby language
  - Includes supporting infrastructure (i.e. Rails) for running Ruby web applications on IBM i
  - Available for download from PowerRuby.com
  - Includes native DB2 database driver – MySQL not necessary
  - Integrates with XMLSERVICE for access to IBM i programs and objects

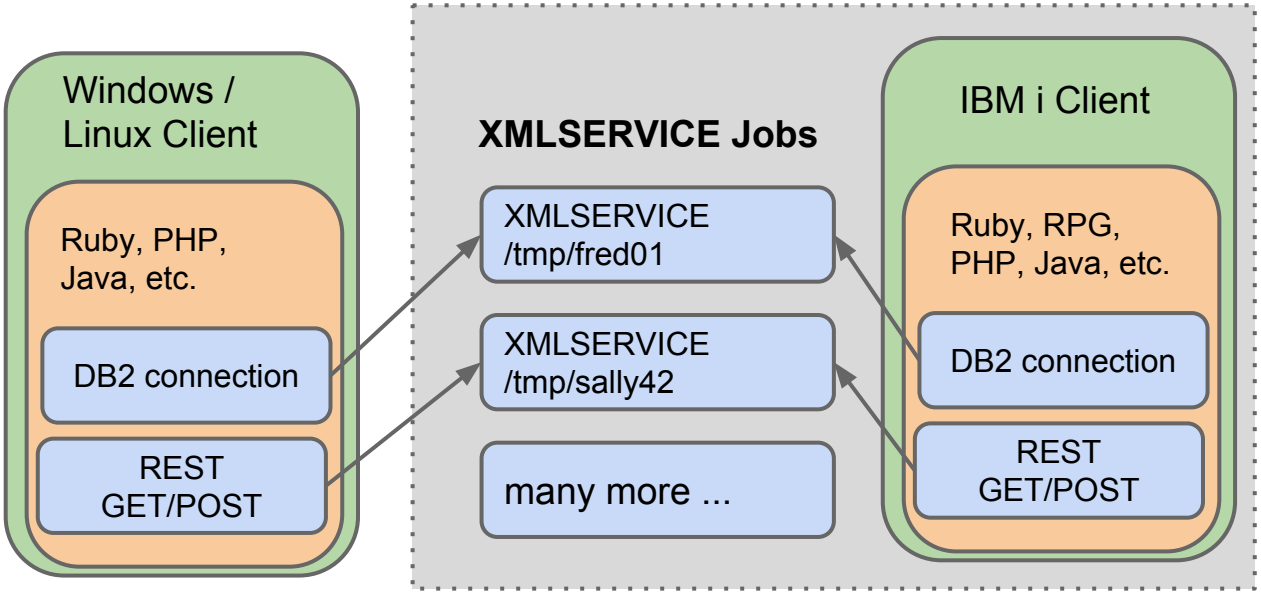
- Components
  - Ruby 2.1.0, 2.0.0, and 1.9.3 (MRI implementation, a.k.a CRuby)
  - ibm_db (IBM supported - rubyforge.org/projects/rubyibm/)
  - xmlservice (IBM supported - youngiprofessionals.com/wiki/index.php/XMLService)
  - Apache + Thin ← the web server stack (more Ruby app servers options coming)
  - Rails 3.2.x and 4.0.x (server-side asset compilation support in the works)

- Learn more
  - PowerRuby.com for downloads, updates, and news
  - twitter.com/rubyonpower

Ruby Integration with i - XMLSERVICE

- DB2 for i – SQL and Native
- Program call
- Procedure call
- Data Area
- Data Queue
- Message Queue
- Commands
- System values
- Spool files

- Delivered with OS as of IBM i 7.1: TR5 and also included with PowerRuby
- Similar to IBM ToolBox for Java, but better in many ways
- Great for re-use of existing RPG code
- Author: Tony Cairns, IBM Rochester



YoungiProfessionals.com/wiki/XMLSERVICE (a.k.a. YiPs)

More open source at YoungiProfessionals.com/**wiki**

*

XMLSERVICE - RPG program to call

XMLSERVICE/ZZCALL

```
D  INCHARA        S                1a
D  INDEC1         S                7p 4

C     *Entry        PLIST
C                   PARM                    INCHARA
C                   PARM                    INDEC1
 /free

   INCHARA = 'C';
   INDEC1 = 321.1234;
   return;

 /end-free
```

Much more complex program calls are also supported and even go beyond the PCML/Java Toolbox limitations with data structures, sub-procedure return values, and date/time data types.

*

XMLSERVICE - Ruby invoking RPG

```
require 'xmlservice'
module ZzcallHelper
  def self.callpgm()
    ActiveXMLService::Base.establish_connection connection: 'ActiveRecord'
    zzcall = XMLService::I_PGM.new("ZZCALL", 'XMLSERVICE') <<
       XMLService::I_a.new('inchara',1,'a') <<
       XMLService::I_p.new('indec1',7,4,11.1111)

     zzcall.xmlservice

     puts "zzcall.xml_PARM3.xml_indec1: #{zzcall.xml_PARM1.xml_indec1}"
  end
end
```

Invoke via PASE command line:

```
$ cd /your/rails/apps/root/dir
$ rails console
irb> include ZzCallHelper
irb> ZzCallHelper.callpgm
```

- The call to Ruby method `establish_connection` omits the use of parentheses - another feature of Ruby to lessen the amount of code and make it easier/quicker to visually scan code. The same is true for `puts`
- Ruby's left shift operator (i.e. `'<<'`) is used to append parameters to the call.
- Line zzcall.xmlservice makes the

PowerRuby Install

Download zip with *SAVF objects from PowerRuby.com

CRTSAVF FILE(QGPL/PRUBY_BASE)

CRTSAVF FILE(QGPL/PRUBY_0001)

CRTSAVF FILE(QGPL/PRUBY_0002)

CRTSAVF FILE(QGPL/PRUBY_0003)

RSTLICPGM LICPGM(1PRUBY1) DEV(*SAVF) LNG(2924) SAVF(QGPL/PRUBY_BASE)

RSTLICPGM LICPGM(1PRUBY1) DEV(*SAVF) LNG(2924) OPTION(1) SAVF(QGPL/PRUBY_0001)

RSTLICPGM LICPGM(1PRUBY1) DEV(*SAVF) LNG(2924) OPTION(2) SAVF(QGPL/PRUBY_0002)

RSTLICPGM LICPGM(1PRUBY1) DEV(*SAVF) LNG(2924) OPTION(3) SAVF(QGPL/PRUBY_0003)

---- QSYS.LIB ----

POWER_RUBY - Contains commands like RAILSNEW and RAILSSVR to create new projects and administer the Rails application server respectively.

---- IFS Root ----

`/PowerRuby/prV2R1M0` Various Ruby versions

`/PowerRuby/prV2R1M0/lib/ruby/gems` Precompiled gems including Rails, Thin, ibm_db

`/PowerRuby/oss/bin` Open source tools including bash, git, joe

New App From Scratch

The RAILSNEW command creates a new Rails application, including bundling necessary gems.

RAILSNEW

VRM(*V200) --version of PowerRuby

PLACE(*WWW) -- *WWW or *HOME

RAILSAPP(A2222) -- Name of Rails app. /www/A2222/htdocs/***A2222***

DBUSR(A2222) -- CRTUSRPRF A2222 with below DBPASS

DBPASS(A2222)

DBROOT(A2222) -- Run SQL: CREATE COLLECTION ***A2222_D;*** CREATE COLLECTION ***A2222_T***

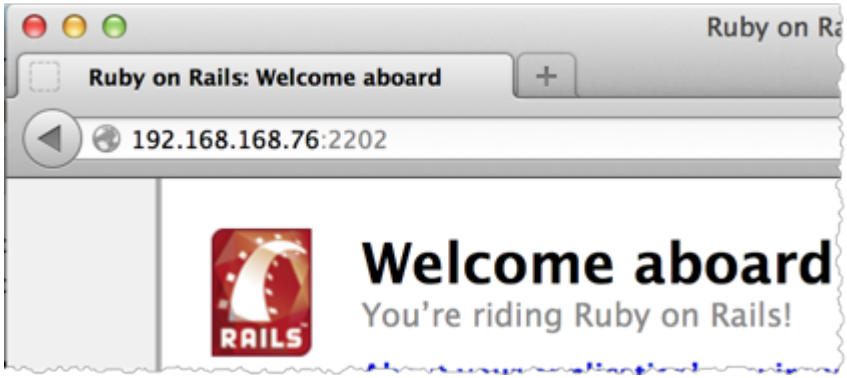
HTTPSRV(A2222) -- /www/***A2222***

EXTPORT(2222) -- /www/A2222/conf/httpd.conf ***Listen *:2222***

INTPORT(2202) -- thin start -p ***2202***

Command `bundle install --local` is run so all gems are resolved and the Gemfile.lock file is created.

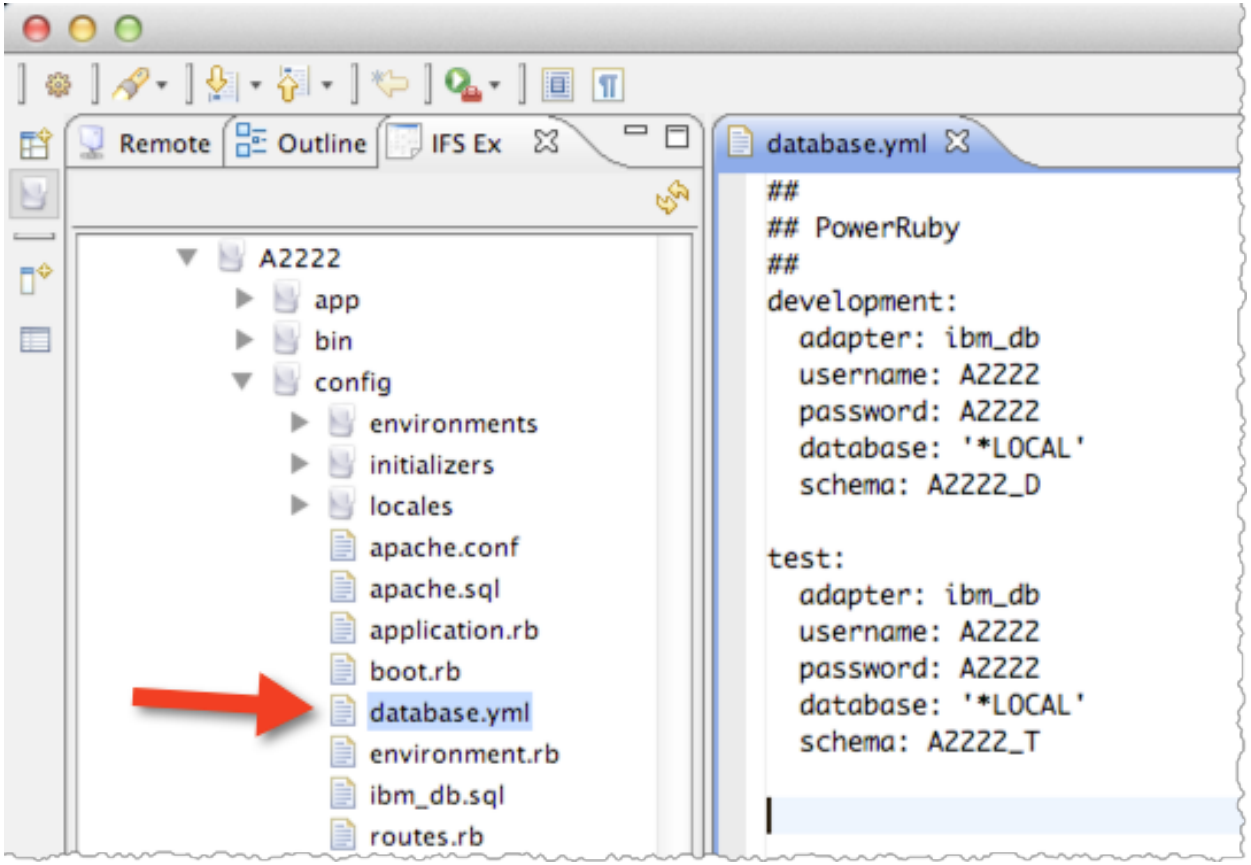
The corresponding Apache and Thin jobs are started as part of the RAILSNEW command.



Database configuration

File `/www/A2222/htdocs/A2222/config/`**`database.yml`** stores the database connection information. Notice the multiple environments - `development` and `test`.

`*LOCAL` declares the Rails app is running on this machine.



NOTE: There are additional options to do encryption of the clear-text password shown below.

Create new model

Create a new model named ***Post*** with two properties, ***title*** and ***text***.

File app/models/***post.rb***

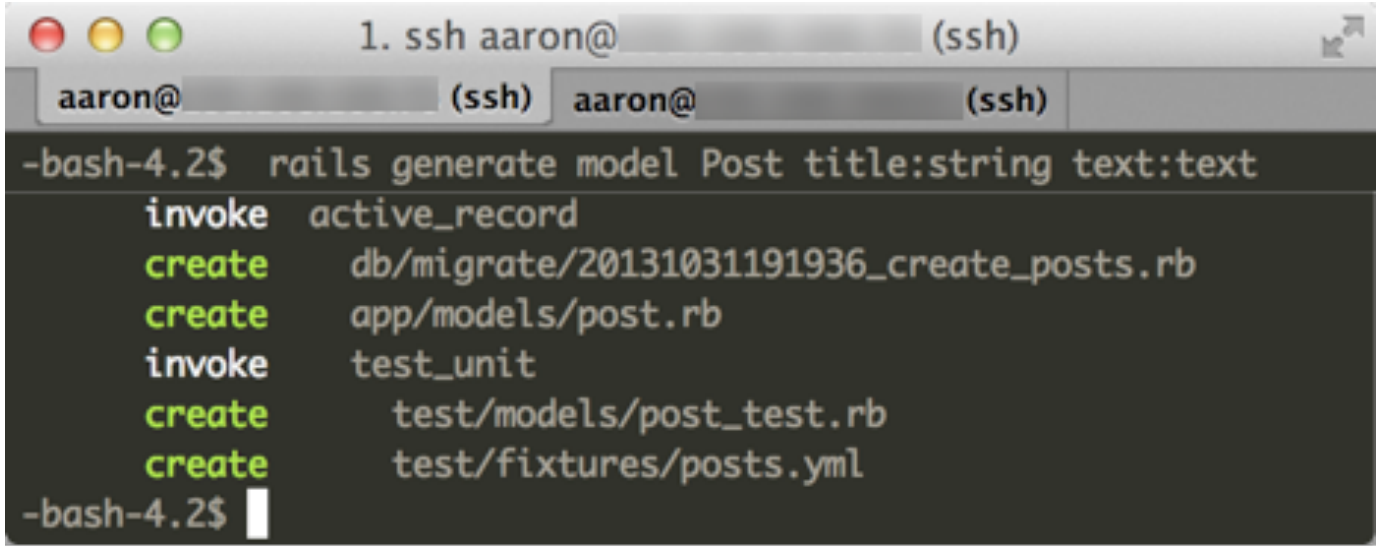
```
class Post < ActiveRecord::Base
end
```

File app/db/migrate/***20131031191936_create_posts.rb***

```
class CreatePosts < ActiveRecord::Migration
  def change
    create_table :posts do |t|
      t.string :title
      t.text :text
      t.timestamps
    end
  end
end
```

A Rails *model* object is used to interact with a DB2 table and store business logic.

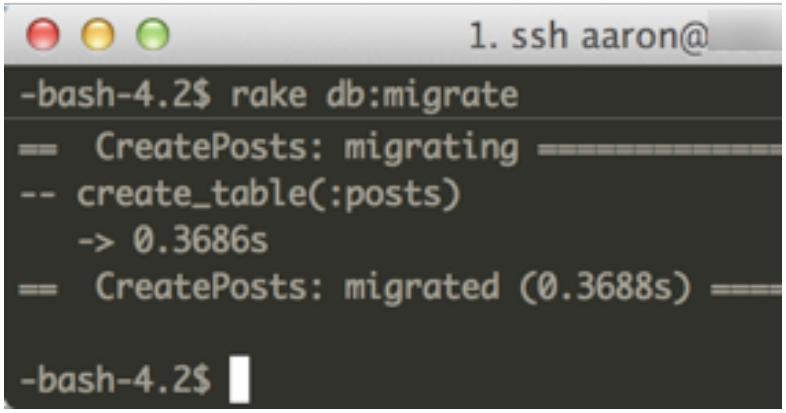
NOTE: The DB2 table does not yet exist. The `rake db:migrate` command will create it.

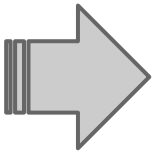


Database migrations

The `rake db:migrate` command queries table `A2222_D/SCHEMA_MIGRATIONS` (`SCHEM00001` below) to learn the most recent migration run against this schema (aka library).

Recognizes `20131031191936_create_posts.rb` hasn't been run and invokes it.





```
CREATE TABLE A2222_D.posts(
 id int NOT NULL
        GENERATED ALWAYS AS IDENTITY
        (START WITH 1 INCREMENT BY 1),
 title varchar(255),
 text varchar(255),
 PRIMARY KEY(id)
)
```

WRKOBJ A2222_D/*ALL

Opt	Object	Type	Library	Att
	QSQJRN0001	*JRNRCV	A2222_D	
	QSQJRN	*JRN	A2222_D	
	POSTS	*FILE	A2222_D	PF
	SCHEM00001	*FILE	A2222_D	PF
	SYSCHKCST	*FILE	A2222_D	LF
	SYSCOLUMNS	*FILE	A2222_D	LF

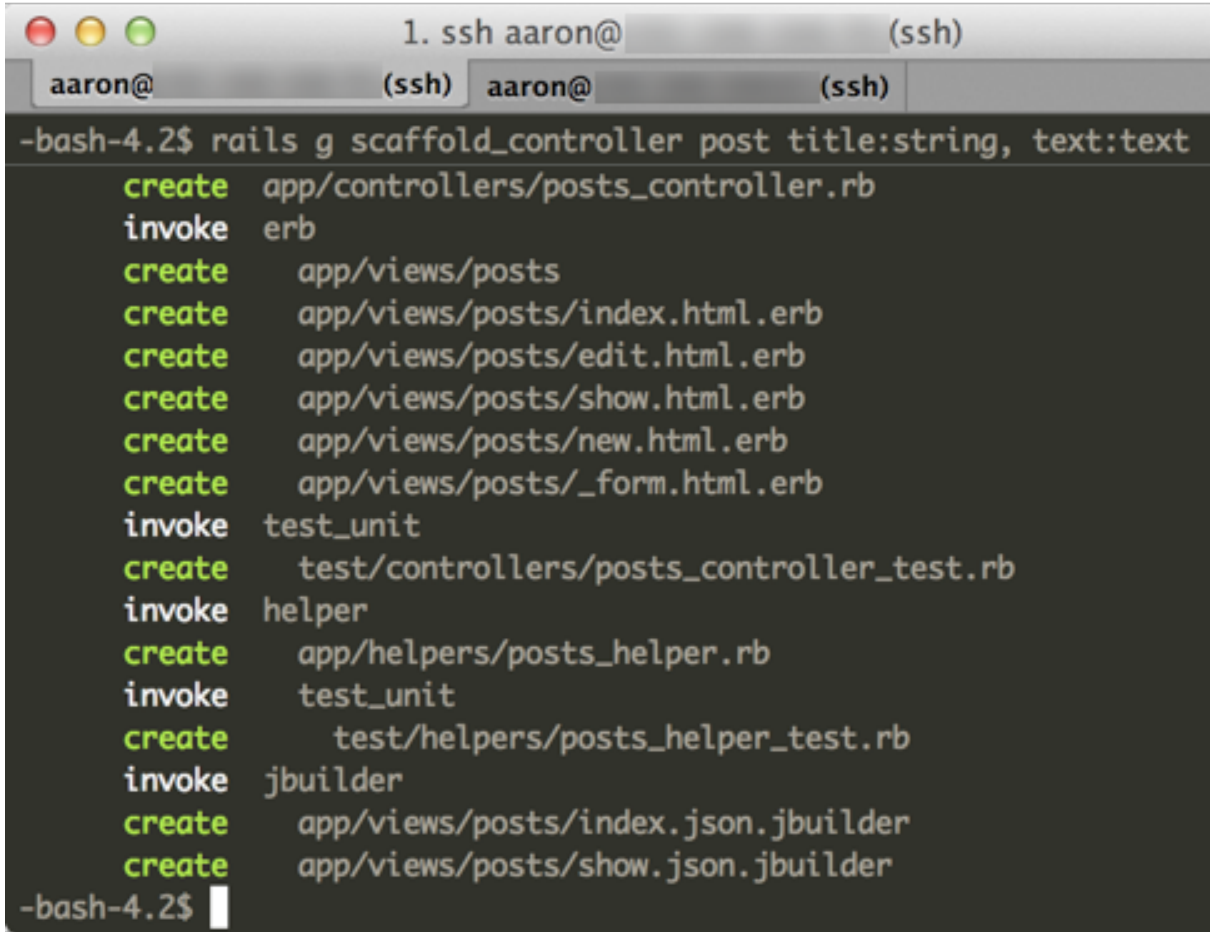
WHAT!?! Shouldn't it be ***POST*** and not ***POST<u>S</u>***?

It is a Rails convention to name tables the plural of the model they represent.

Generating controller and views

The `rails g scaffold_controller`command creates stubbed out files as shown below. Many assumptions are made in generating the controller *(i.e. that a model named Post exists)*.

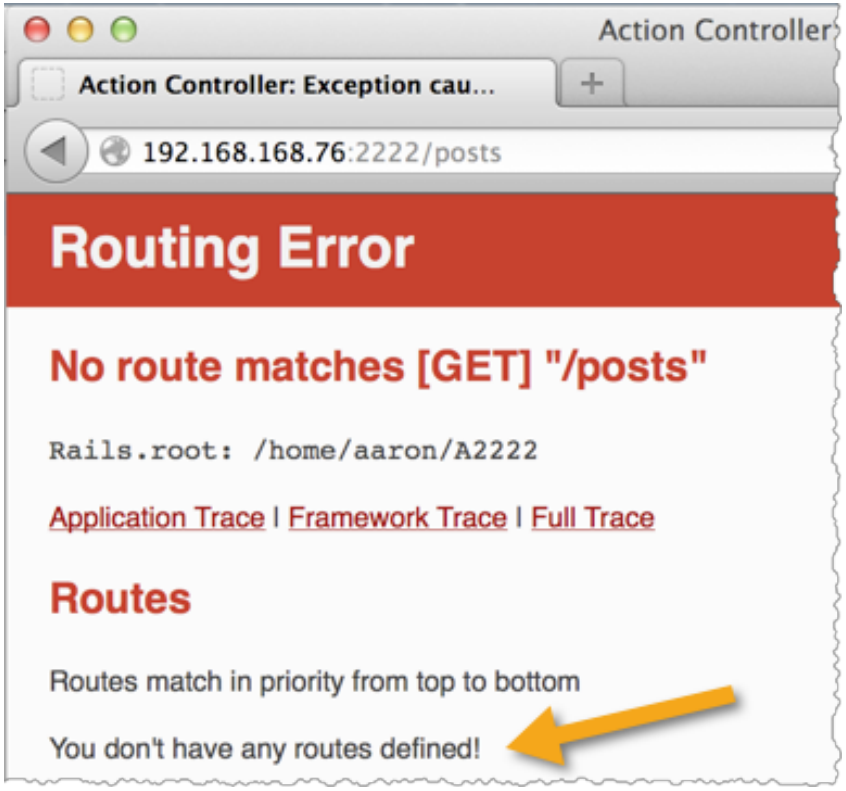
Once generation is completed you can modify to suit your needs. Easily saves hours of programming!



Learn more about the `rails generate` command here.

Did you discover your own commonly used pattern? Learn how to create your own generator here.

Routing requests

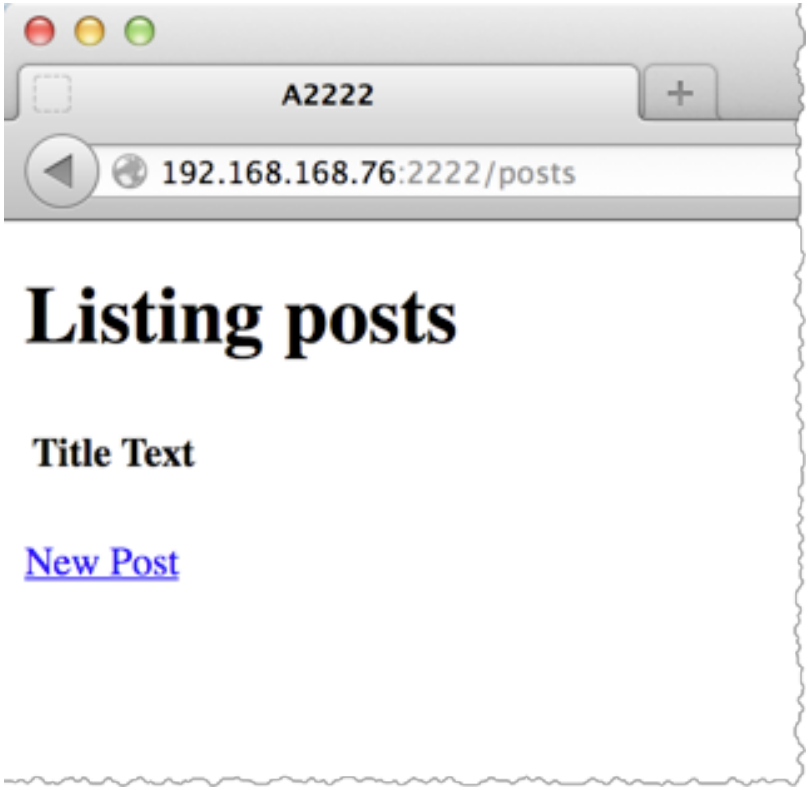


Before you can display the application there needs to be an entry added to the `routes.rb` file so the Rails application knows you want to expose all actions (index, edit, show, new).

Learn more about Rails routing here.

app/config/routes.rb

```
A2233::Application.routes.draw do
  resources :posts
end
```



What's involved?

app/views/posts/**index.html.erb**

```
<h1>Listing posts</h1>
<table>
  <thead>
    <tr><th>Title</th><th>Text</th><th></th><th></th><th></th><th></th></tr>
  </thead>
  <tbody>
    <% @posts.each do |post| %>
      <tr>
        <td><%= post.title %></td>
        <td><%= post.text %></td>
        <td><%= link_to 'Show', post %></td>
        <td><%= link_to 'Edit', edit_post_path(post) %></td>
        <td><%= link_to 'Destroy', post, method: :delete, data: {
confirm: 'Are you sure?' } %></td>
      </tr>
    <% end %>
  </tbody>
</table>
<br><%= link_to 'New Post', new_post_path %>
```

app/controllers/**posts_controller.rb**

```
class PostsController < ApplicationController
  def index
    @posts = Post.all
  end
.  .  .
end
```

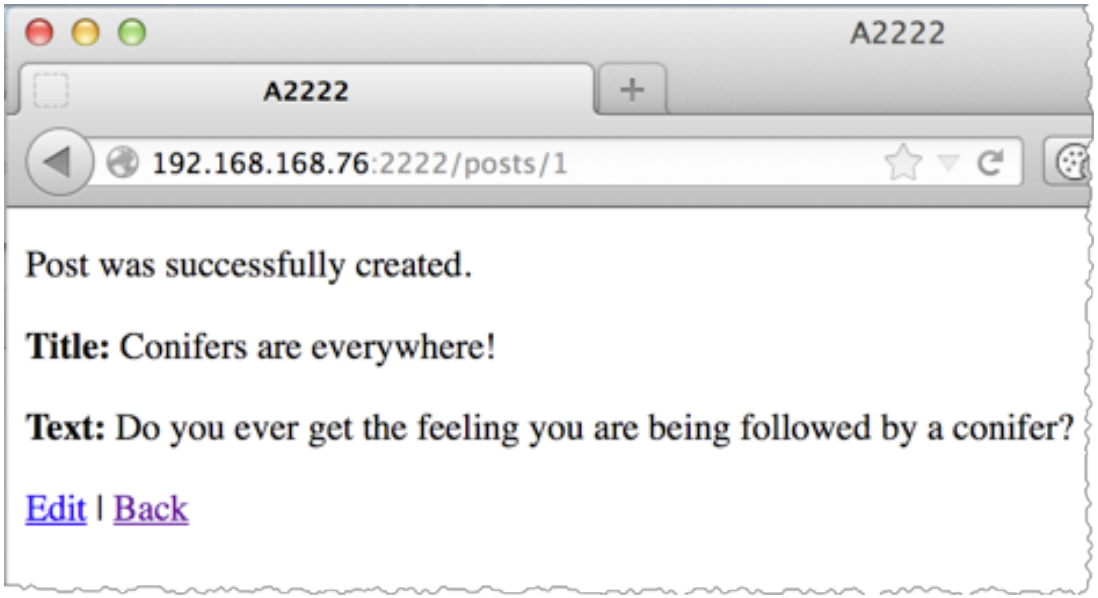
app/models/**post.rb**

```
class Post < ActiveRecord::Base
end
```

Add a record

Add record functionality generated by default.

Obviously you'd want to create some custom CSS to better the look and feel of the view layer - these are just defaults to get infrastructure in place.



Interactive Ruby (irb)

... IRB (Interactive Ruby Shell) is an interactive environment where you can enter Ruby syntax and have it immediately be run. (see below where "rails c" is starting the Rails Console and inherently invoking IRB)

Great for learning Ruby and Rails, testing ActiveRecord queries (as shown below), among many other things.

```
1. ssh aaron@192.168.168.76 (ssh)
-bash-4.2$ rails c
Loading development environment (Rails 4.0.0)
irb(main):001:0> Post.last
   (6.0ms)  SET SCHEMA A2222_D
   (6.9ms)  SET SCHEMA A2222_D
  Post Load (48.2ms)  SELECT posts.* FROM posts ORDER BY posts.id DESC FETCH FI
RST 1 ROWS ONLY
=> #<Post id: 1, title: "Conifers are everywhere!", text: "Do you ever get the
feeling you are being followed ...", created_at: "2013-10-31 00:00:00", updated
_at: "2013-10-31 00:00:00">
irb(main):002:0>
```

BIG TIME SAVER!

Normally, to run a quick bit of code you'd create a file, enter the code, and run it in the browser. This gets you right into testing the code!

Existing utilities like `STRSQL` still work on the newly created tables.

```
STRSQL: SELECT * FROM A2222.POSTS                    Display Data

Position to line  . . . . .

....+....1....+....2....+....3....+....4....+....5....+....6....+....7.
           ID   TITLE
            1   Conifers are everywhere!
********  End of data  ********
```

What's running?

- `A2222` entries are Apache jobs
- `RAILSSVR` is a Thin job

Work with Active Jobs

```
CPU %:    17.6     Elapsed time:   00:00:00     Active jobs:   196
                    Current
Opt  Subsystem/Job  User        Type  CPU %  Function        Status
     QHTTPSVR       QSYS        SBS      .0                  DEQW
       A2222        QTMHHTTP    BCH      .0  PGM-QZHBMAIN    SIGW
       A2222        QTMHHTTP    BCI      .0  PGM-QZSRLOG     SIGW
       A2222        QTMHHTTP    BCI     12.3  PGM-QZSRHTTP    SIGW
     QSYSWRK        QSYS        SBS      .0                  DEQW
       RAILSSVR     AARON       BCH      .0  PGM-ruby        SELW
```

Stopping and Starting

Start Apache

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(A2222)
```

End Apache

```
ENDTCPSVR SERVER(*HTTP) HTTPSVR(A2222)
```

Start Thin

```
RAILSSVR ACTION(*START) APP('/www/A2222/htdocs/A2222') VRM(*V200) PORT(2202)
```

End Thin

```
RAILSSVR ACTION(*END) APP('/www/A2222/htdocs/A2222')
```

Logs

Locations to look for logs and/or errors:

```
WRKSPLF <profile that used RAILSSVR>

WRKSPLF QTMHHTTP   <---- Apache job logs
```

/www/A2222/htdocs/A2222/logs/**development.log** <----- Rails log

```
Started GET "/posts" for  at 2014-01-20 22:22:57 +0000
Processing by PostsController#index as HTML
  [1m[35mPost Load (0.2ms)[0m  SELECT "posts".* FROM "posts"
  Rendered posts/index.html.erb within layouts/application (183.4ms)
Completed 200 OK in 280ms (Views: 230.7ms | ActiveRecord: 9.7ms)

Started GET "/postss" for  at 2014-01-21 00:36:49 +0000
ActionController::RoutingError (No route matches [GET] "/postss"):
  actionpack (4.0.0) lib/action_dispatch/middleware/debug_exceptions.rb:21:in `call'
  actionpack (4.0.0) lib/action_dispatch/middleware/show_exceptions.rb:30:in `call'
  railties (4.0.0) lib/rails/rack/logger.rb:38:in `call_app'
  railties (4.0.0) lib/rails/rack/logger.rb:21:in `block in call'
. . .
```

Rake

... a software task management tool similar to "make". It allows you to specify tasks and describe dependencies as well as to group tasks in a namespace.

- **Est 2005 by Jim Weirich**
- Syntax is completely defined in Ruby
- "Prerequisites" (i.e. `:environment`) allow for things like the Rails environment be loaded before the task is invoked.
- One could think of it as *"The CL of Ruby"*

my_app/lib/tasks/**set_pay_in_full.rake**

```
task :set_pay_in_full => :environment
do
  Enrollment.all.each do |e|
    e.pay_in_full = e.term.pay_in_full
    e.save!
  end
end
```

Invoke via PASE command line:

```
$ rake set_enrollment_pay_in_full_from_term
```

github.com/jimweirich/rake – Home website
railscasts.com/episodes/66-custom-rake-tasks – Video tutorial overview

We have reached the end!

PowerRuby

Go big.

Aaron Bartell

aaron@PowerRuby.com

www.PowerRuby.com

twitter.com/RubyOnPower