



Innovation In Rails



<http://railsenvy.com/railsconf09>

Ruby/Rails IDE?





RubyMine 1.0 RELEASED

The most intelligent Ruby IDE


[Download](#)

[Buy Now](#)
[Overview](#)
[Features & Screenshots](#)
[Download](#)
[Docs & Demos](#)
[Buy](#)
[BOOKMARK](#)

The Most Intelligent Ruby IDE

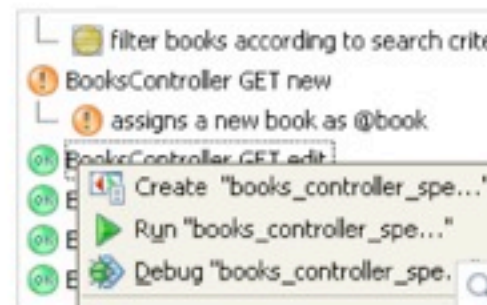
JetBrains RubyMine – Ruby and Rails IDE built on years proven IntelliJ platform. RubyMine brings the whole range of essential developers tools, all tightly integrated together for creating the convenient development environment for productive Ruby development and Web development with Ruby on Rails.

Key Features

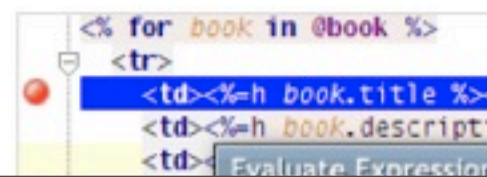
- **Intelligent Editor:** code completion, code snippets and automatic refactorings
- **Project Navigation:** wide range of views and one-click jumping between components
- **Error-Free Coding:** on-the-fly code analysis and type inference
- **Web Development with Ruby on Rails** (including best-of-breed HTML, CSS and JavaScript editing support)
- **RSpec and Test::Unit** support, with GUI-based test runner
- **Ruby Debugger:** full support for Rails applications debugging and easy-to-use interface
- **VCS Integrations:** Git, Perforce, Subversion and CVS with changelists and merge
- **Open API and Plugin Development SDK**, ensuring IDE



RHTML editing and web page preview



Test runner: develop, test and fix



RubyMine Roadmap



Apr'09 – RubyMine 1.0 Release

May'09 – RubyMine 1.1 (Rails 2.3)

Fall'09 – [RubyMine 1.5 Release](#)

JetBrains RubyMine Community:

[Early Access Program](#)

[Discussion Forum](#)

[Bug Tracker](#)

[IntelliJ IDEA Ruby Plugin](#)



RubyMine blog »

Tips & tricks and news

Using RubyMine allows us to develop using a common tool making it easier to pair and help each other out without having to lose productivity due to the context switching from one development environment (platform) to another.

Jeremy Anderson, agile software developer at Pillar Technology

[More testimonials »](#)

TeamCity

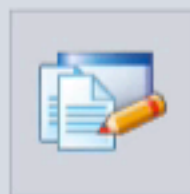
build server & continuous integration for Ruby

[DOWNLOAD FREE EDITION](#)



Develop with pleasure!

Quick Start



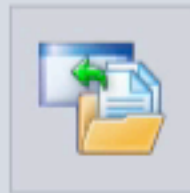
Create New Project

Create a new project



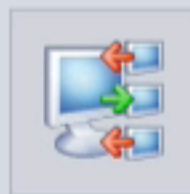
Open Directory

Start editing code in the specified directory



Reopen Recent Project

You can open one of the most recent projects you were working with. Click the icon or link to select a project from the list.



Check out from Version Control

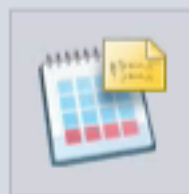
You can check out an entire project from a Version Control System. Click the icon or link to select your VCS.

Documentation



Read Help

Open JetBrains RubyMine "Help Topics" in a new window.



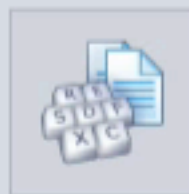
Tips of the Day

Learn how to increase your coding productivity.



Demos and Tutorials

View short live demos introducing features of RubyMine



Default Keymap Reference

Open PDF file with the default keymap reference card.

Plugins

[Open Plugin Man...](#)

My Plugins:

No plugins currently installed.

Bundled Plugins:

CSS Support

Provides syntax and error highlighting, completion navigation and other editing aids for CSS

CVS Integration

Provides integration with CVS version control system

Git Integration

Provides integration with Git version control system

HTML Preview

Provides preview for HTML files using Mozilla browser (not available on 64bit platforms)

Images

Provides image viewing and thumbnail browsing

Inspection-JS

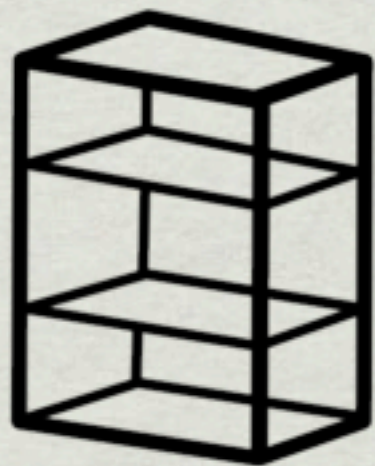
Adds over 100 new automated code inspections for auditing JavaScript.

JavaScript Intention Power Pack



“Rolling your own, tiny like, web thingoes”

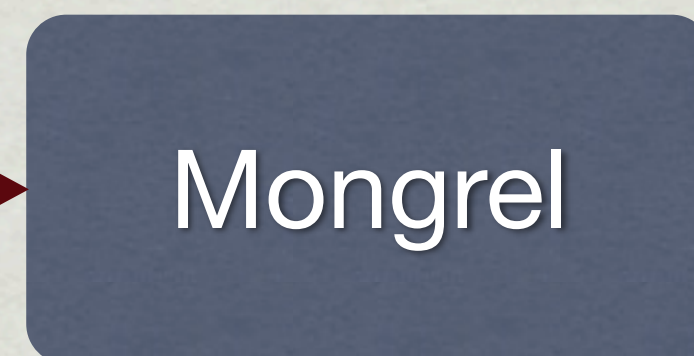
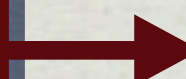
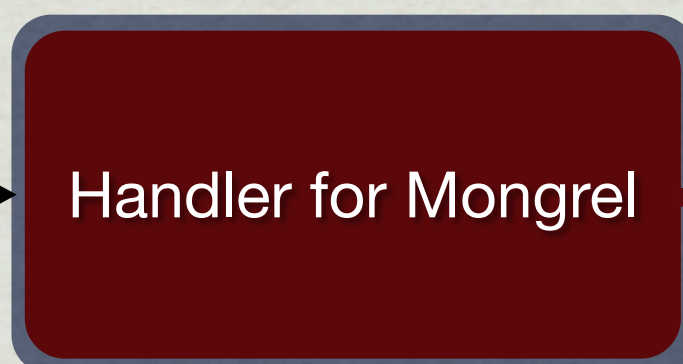
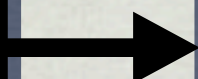
-Ryan Allen

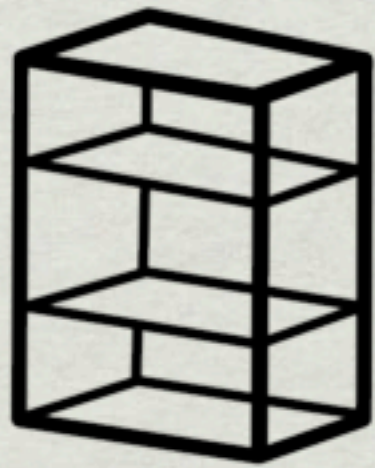


rack
powers web applications

Framework A

Rack





rack
powers web applications

Framework A

Core Code

Rack

- * MONGREL
- * EVENTEDMONGREL
- * SWIFTPLIEDMONGREL
- * WEBRICK
- * FCGI
- * CGI
- * SCGI
- * LITESPEED
- * THIN
- * EBB
- * FUZED
- * PHUSION PASSENGER

Mongrel

Rack Middleware



Telephone

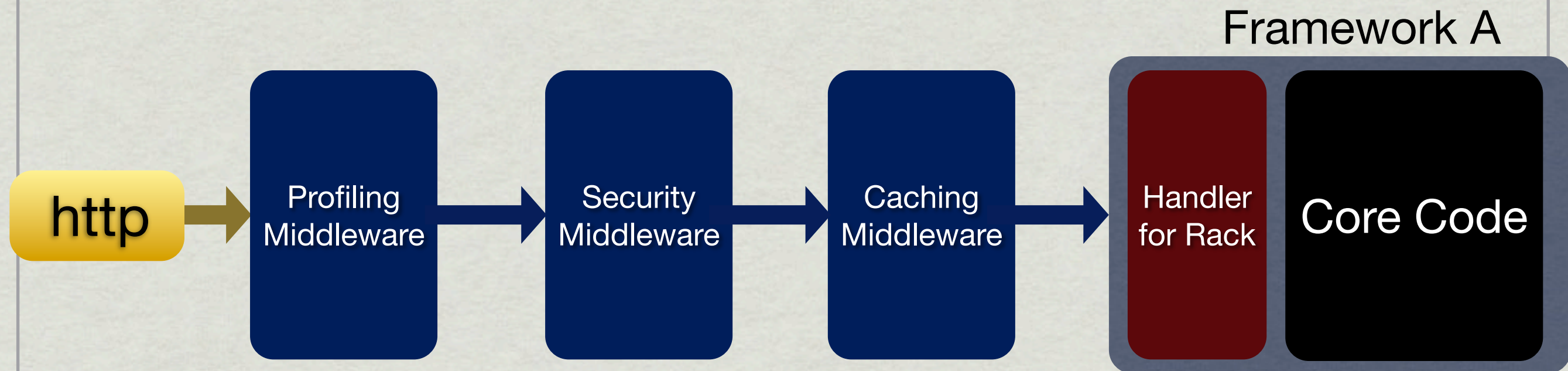
I love my Car

I loathe Sylar

I love Sylar

Brains are Yummy!





Framework A

http

Profiling
Middleware

Security
Middleware

Caching
Middleware

Handler
for Rack

Core Code

Modify the incoming request

Manipulate the response

```
class StandardRack
```

```
  def initialize(app)
```

```
    @app = app
```

```
  end
```

```
  def call(env)
```



```
    status, headers, response = @app.call(env)
```

```
    [status, headers, response]
```

```
  end
```

```
end
```



http

Profiling
Middleware

```
def call(env) *  
  @app.call(env)  
end
```

call

Security
Middleware

```
def call(env) *  
  @app.call(env)  
end
```

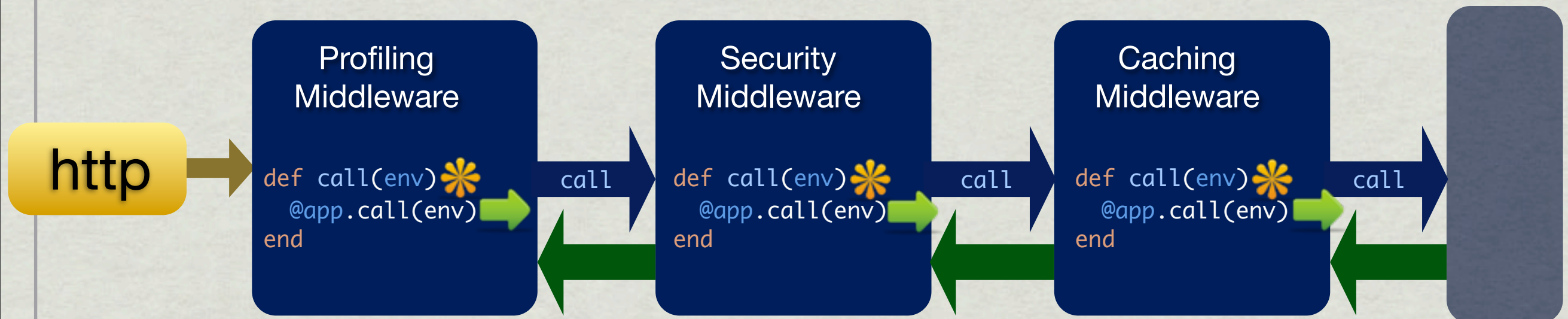
call

Caching
Middleware

```
def call(env) *  
  @app.call(env)  
end
```

call

Framework A





2.3

Released March 16, 2009

Now on Rack!

Rack::Cache

Reverse Proxy training wheels for your Rails app

```
gem install rack-cache
```

environment.rb

```
Rails::Initializer.run do |config|  
  ...  
  config.middleware.use("Rack::Cache")  
end
```



```
graph LR; Input(( )) --> RP[Reverse Proxy]; RP --> Rails[Rails];
```

Reverse Proxy

Rails

Contributed Rack Middleware and Utilities

This package includes a variety of add-on components for Rack, a Ruby web server interface:

- Rack::ETag - Automatically sets the ETag header on all String bodies.
- Rack::JSONP - Adds JSON-P support by stripping out the callback param and padding the response with the appropriate callback format.
- Rack::LighttpdScriptNameFix - Fixes how lighttpd sets the SCRIPT_NAME and PATH_INFO variables in certain configurations.
- Rack::Locale - Detects the client locale using the Accept-Language request header and sets a rack.locale variable in the environment.
- Rack::MailExceptions - Rescues exceptions raised from the app and sends a useful email with the exception, stacktrace, and contents of the environment.
- Rack::NestedParams - parses form params with subscripts (e.g., * "post[title]=Hello") into a nested/recursive Hash structure (based on Rails' implementation).
- Rack::PostBodyContentTypeParser - Adds support for JSON request bodies. The Rack parameter hash is populated by deserializing the JSON data provided in the request body when the Content-Type is application/json.
- Rack::ProcTitle - Displays request information in process title (\$0) for monitoring/inspection with ps(1).
- Rack::Profiler - Uses ruby-prof to measure request time.
- Rack::Sendfile - Enables X-Sendfile support for bodies that can be served from file.
- Rack::Signals - Installs signal handlers that are safely processed after a request
- Rack::TimeZone - Detects the clients timezone using JavaScript and sets a variable in Rack's environment with the offset from UTC.
- Rack::Evil - Lets the rack application return a response to the client from any place.
- Rack::Callbacks - Implements DLS for pure before/after filter like Middlewares.
- Rack::Config - Shared configuration for cooperative middleware.
- Rack::NotFound - A default 404 application.
- Rack::CSSHTTPRequest - Adds CSSHTTPRequest support by encoding responses as CSS for cross-site AJAX-style data loading
- Rack::Deflect - Helps protect against DoS attacks.
- Rack::ResponseCache - Caches responses to requests without query strings to Disk or a user provider Ruby object. Similar to Rails' page caching.
- Rack::RelativeRedirect - Transforms relative paths in redirects to absolute URLs.
- Rack::Backstage - Returns content of specified file if it exists, which makes it convenient for putting up maintenance pages.
- Rack::Format - Adds a format extension at the end of the URI when there is none, corresponding to the mime-type given in the Accept HTTP header.

[HTTP://GITHUB.COM/RACK/RACK-CONTRIB/TREE/MASTER](http://github.com/rack/rack-contrib/tree/master)



But where in the stack does it go?



```
$ rake middleware
```

```
use Rack::Lock
use ActionController::FailSafe
use ActionController::Session::CookieStore
use ActionController::ParamsParser
use Rack::MethodOverride
use Rack::Head
use ActiveRecord::ConnectionAdapters::ConnectionManagement
use ActiveRecord::QueryCache
run ActionController::Dispatcher.new
```

Rails has it's own Stack

ActionController::Dispatcher

```
middlewares = File.join(File.dirname(__FILE__), "middlewares.rb")  
middleware.instance_eval(File.read(middlewares))
```


middlewares.rb

```
use "Rack::Lock", :if => lambda {  
  !ActionController::Base.allow_concurrency  
}  
  
use "ActionController::Fail-safe"  
  
use lambda { ActionController::Base.session_store },  
   lambda { ActionController::Base.session_options }  
  
use "ActionController::ParamsParser"  
use "Rack::MethodOverride"  
use "Rack::Head"
```

But what does each bit do?

Rails Rack Bits

middlewares.rb

```
use "Rack::Lock", :if => lambda {  
  !ActionController::Base.allow_concurrency  
}
```

/rack-1.0.0/lib/lock.rb

```
module Rack  
  class Lock  
    def call(env) *  
      @lock.synchronize { @app.call(env) } ➡  
    end  
  end  
end
```

(this code has been simplified)

Rails Rack Bits

middlewares.rb

```
use "ActionController::Fail-safe"
```

/action_controller/action_dispatch/middleware/failsafe.rb

```
module ActionController
  class Fail-safe
    def call(env) ✨
      @app.call(env) ➡
    rescue Exception => exception
      failsafe_response(exception)
    end
  end
end
```

(this code has been simplified)

We're sorry, but something went wrong.

We've been notified about this issue and we'll take a look at it shortly.

```
/?\ FAILSAFE /\ Fri Apr 24 23:32:36 0200 2009  
Status: 500 Internal Server Error
```

You only see this if rescue.rb missed

Rails Rack Bits

middlewares.rb

```
use lambda { ActionController::Base.session_store },  
    lambda { ActionController::Base.session_options }
```

/actionpack/lib/action_dispatch/middleware/session/cookie_store.rb

```
module ActionDispatch  
  module Session  
    class CookieStore  
      def call(env) *  
        env[ENV_SESSION_KEY] = AbstractStore::SessionHash.new(self, env)  
  
        status, headers, body = @app.call(env) ➡  
  
        cookie = build_cookie(@key, cookie.merge(options))  
        headers[HTTP_SET_COOKIE] = cookie  
  
        [status, headers, body]  
      end  
    end  
  end  
end
```

(this code has been simplified)

Rails Rack Bits

middlewares.rb

```
use "ActionController::ParamsParser"
```

/actionpack/lib/action_dispatch/middleware/params_parser.rb

```
module ActionDispatch
  class ParamsParser
    ActionController::Base.param_parsers[Mime::XML] = :xml_simple
    ActionController::Base.param_parsers[Mime::JSON] = :json

    def call(env) ✿
      if params = parse_formatted_parameters(env)
        env["action_controller.request.request_parameters"] = params
      end

      @app.call(env) ➡
    end
  end
end
```

(this code has been simplified)

Rails Rack Bits

middlewares.rb

```
use "Rack::MethodOverride"
```

/rack-1.0.0/lib/rack/methodoverride.rb

```
module Rack
  class MethodOverride
    HTTP_METHODS = %w(GET HEAD PUT POST DELETE OPTIONS)

    def call(env) *
      if env["REQUEST_METHOD"] == "POST"

        method = Request.new(env).POST["_method"]

        if HTTP_METHODS.include?(method)
          env["REQUEST_METHOD"] = method
        end
      end

      @app.call(env) ➡
    end
  end
end
```

(this code has been simplified)

Rails Rack Bits

middlewares.rb

```
use "Rack::Head"
```

/rack-1.0.0/lib/rack/head.rb

```
module Rack
  class Head
    def call(env) *
      status, headers, body = @app.call(env) ➡

      if env["REQUEST_METHOD"] == "HEAD"
        [status, headers, []]
      else
        [status, headers, body]
      end
    end
  end
end
```

(this code has been simplified)

middlewares.rb

```
use "Rack::Lock", :if => lambda {  
  !ActionController::Base.allow_concurrency  
}  
  
use "ActionController::FailSafe"  
  
use lambda { ActionController::Base.session_store },  
  lambda { ActionController::Base.session_options }  
  
use "ActionController::ParamsParser"  
use "Rack::MethodOverride"  
use "Rack::Head"
```

Creates a lock around the request

Last Resort 500 Error

Persist Session Information

Parse Params from XML/JSON

Set proper Method

Clear body with Head Requests

middlewares.rb

```
use "Rack::Lock", :if => lambda {  
  !ActionController::Base.allow_concurrency  
}
```

```
use "ActionController::Fail-safe"
```

```
use lambda { ActionController::Base.session_store },  
lambda { ActionController::Base.session_options }
```

Where did this come from?

```
use "ActionController::ParamsParser"
```

```
use "Rack::MethodOverride"
```

```
use "Rack::Head"
```

From command prompt

```
$ rake middleware
```

```
use Rack::Lock
```

```
use ActionController::Fail-safe
```

```
use ActionController::Reloader
```

```
use ActionController::Session::CookieStore
```

```
use ActionController::ParamsParser
```

```
use Rack::MethodOverride
```

```
use Rack::Head
```

```
use ActiveRecord::ConnectionAdapters::ConnectionManagement
```

```
use ActiveRecord::QueryCache
```

```
run ActionController::Dispatcher.new
```


Rails Rack Bits

from console

```
use ActiveRecord::ConnectionAdapters::ConnectionManagement
use ActiveRecord::QueryCache
```

/railties/lib/initializer.rb

```
def initialize_database_middleware
  configuration.middleware.insert_before "ActiveRecord::SessionStore",
    ActiveRecord::ConnectionAdapters::ConnectionManagement
  configuration.middleware.insert_before "ActiveRecord::SessionStore",
    ActiveRecord::QueryCache
end
```



(this code has been simplified)

Rails Rack Bits

from console

```
use ActiveRecord::ConnectionAdapters::ConnectionManagement
```

/activerecord/lib/active_record/connection_adapters/abstract/connection_pool.rb

```
module ActiveRecord
  class ConnectionManagement
    def call(env) *
      @app.call(env) 
    end
  end
end
```

(this code has been simplified)

Rails Rack Bits

from console

```
use ActiveRecord::QueryCache
```

`/activerecord/lib/active_record/query_cache.rb`

```
module ActiveRecord
  class QueryCache
    def call(env) ✱
      ActiveRecord::Base.cache do
        @app.call(env) ➡
      end
    end
  end
end
```

(this code has been simplified)



```
$ rake middleware
```

```
use Rack::Lock
use ActionController::FailSafe
use ActionController::Session::CookieStore
use ActionController::ParamsParser
use Rack::MethodOverride
use Rack::Head
use ActiveRecord::ConnectionAdapters::ConnectionManagement
use ActiveRecord::QueryCache
run ActionController::Dispatcher.new
```

`config.middleware.use("Rack::Cache")`

That's Your Rack Stack

Rails Rack Bits

ActionController::Dispatcher.middleware

Middleware Commands

```
.use MyMiddleware
```

```
.insert_after 'ActionController::Fail-safe', MyMiddleware
```

```
.middleware.swap 'Rails::Rack::Fail-safe', MyFailsafer
```




Metal

Metal is for optimizing actions
using rack end points



Welcome aboard

You're riding Ruby on Rails!

[About your application's environment](#)

Getting started

Here's how to get rolling:

1. Use `script/generate` to create your models and controllers

To see all available options, run it without parameters.

2. Set up a default route and remove or rename this file

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

Search

Join t

[Ruby on](#)
[Official](#)
[Wiki](#)

Browse
docur

[Rails A](#)
[Ruby st](#)
[Ruby co](#)
[Rails G](#)

httperf

1000 Requests

	Time to complete	requests / second	req duration
Controller	6.34 s	160 r/s	6.3ms



httperf

1000 Requests

	Time to complete	requets / second	req duration
Controller	6.34 s	160 r/s	6.3ms
Metal	3.8 s	262 r/s	3.8ms

	requets / second
With ActiveRecord	175 r/s
Without ActiveRecord	260 r/s

Even faster without
ActiveRecord

Rails Rack Bits

from prompt

```
script/generate metal poller
```

```
$ rake middleware
```

```
use Rack::Lock
use ActionController::FailSafe
use ActionController::Reloader
use ActionController::Session::CookieStore
use ActionController::Metal::ParamsParser
use Rack::MethodOverride
use Rack::Head
use ActiveRecord::ConnectionAdapters::ConnectionManagement
use ActiveRecord::QueryCache
run ActionController::Dispatcher.new
```

(this code has been simplified)

/railties/lib/initializer.rb

```
def initialize_metal
  configuration.middleware.insert_before(
    "ActionDispatch::ParamsParser",
    Rails::Rack::Metal, :if => Rails::Rack::Metal.metal?.any?)
end
```

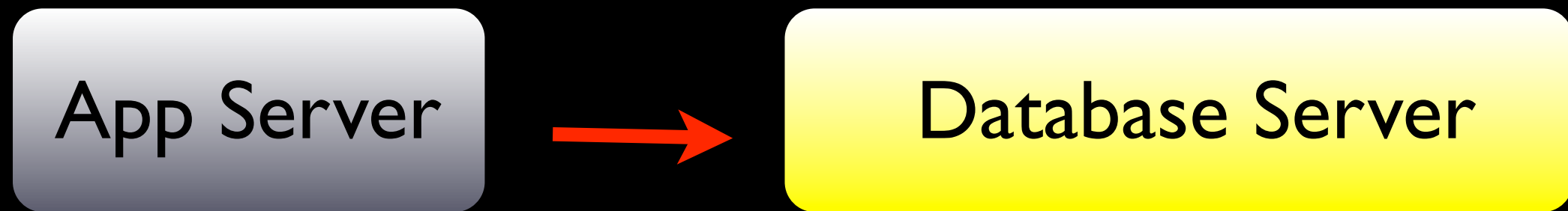
/railties/lib/rails/rack/metal.rb

```
module Rails
  module Rack
    class Metal
      def call(env) *
        @metals.keys.each do |app|
          result = app.call(env) →
          return result unless result[0].to_i == 404
        end
        @app.call(env) →
      end
    end
  end
end
```

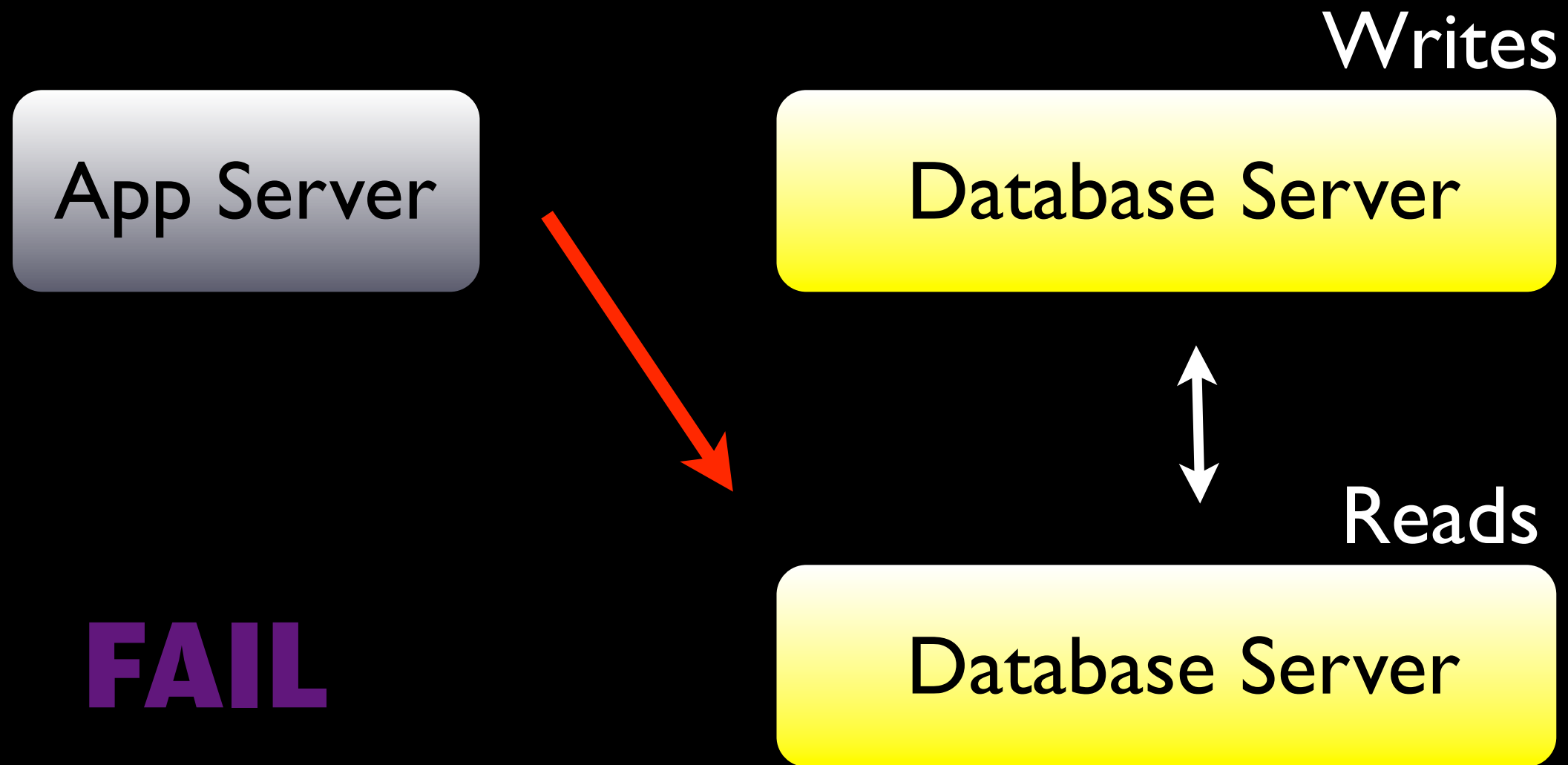

Metal is for optimizing actions
using rack end points

Write-Through Caching

User Updates Profile



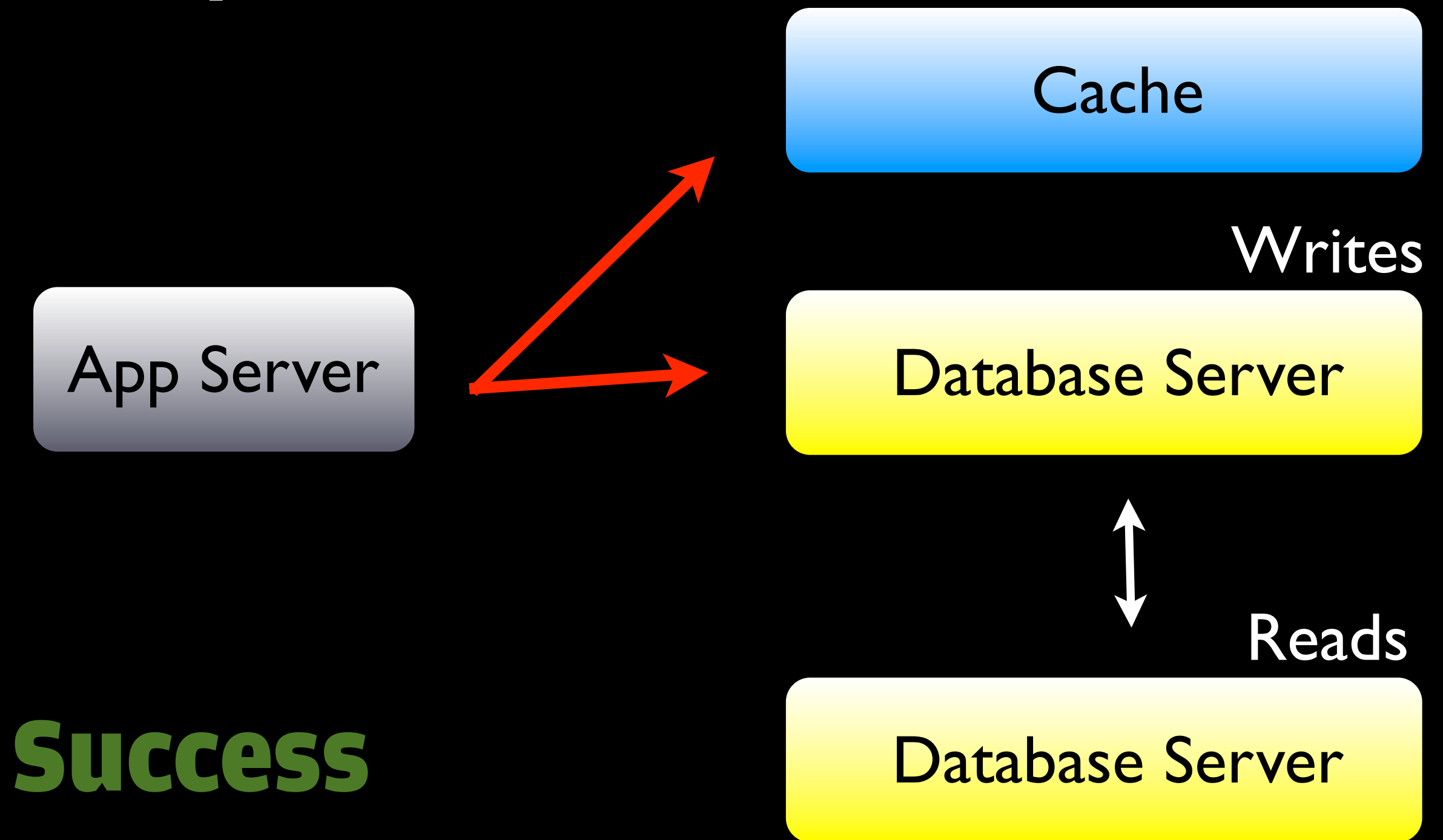
User Updates Profile



REPLICATION LAG



User Updates Profile





PARENTAL
ADVISORY
EXPLICIT CONTENT



<http://shouldiusescala.com>

environment.rb — cache-money-example

environment.rb

cache-money.rb

```
1 # Be sure to restart your server when you modify this file
2
3 # Specifies gem version of Rails to use when vendor/rails is not present
4 RAILS_GEM_VERSION = '2.3.2' unless defined? RAILS_GEM_VERSION
5
6 # Bootstrap the Rails environment, frameworks, and default configuration
7 require File.join(File.dirname(__FILE__), 'boot')
8
9 Rails::Initializer.run do |config|
10   # Settings in config/environments/* take precedence over those specified here.
11   # Application configuration should go into files in config/initializers
12   # -- all .rb files in that directory are automatically loaded.
13
14   # Add additional load paths for your own custom dirs
15   # config.load_paths += %W( #{RAILS_ROOT}/extras )
16   config.gem 'memcache-client', :lib => 'memcache', :version => ">= 1.7.1"
17   config.gem 'nkallen-cache-money', :lib => 'cache_money', :source => 'http://gems.github.com'
18   # Specify gems that this application depends on and have them installed with rake gems:install
19   # config.gem "bj"
20   # config.gem "hpricot", :version => '0.6', :source => "http://code.whytheluckystiff.net"
21   # config.gem "sqlite3-ruby", :lib => "sqlite3"
22   # config.gem "aws-s3", :lib => "aws/s3"
23
24   # Only load the plugins named here, in the order given (default is alphabetical).
25   # :all can be used as a placeholder for all plugins not explicitly named
26   # config.plugins = [ :exception_notification, :ssl_requirement, :all ]
27
28   # Skip frameworks you're not going to use. To use Rails without a database,
29   # you must remove the Active Record framework.
30   # config.frameworks -= [ :active_record, :active_resource, :action_mailer ]
31
32   # Activate observers that should always be running
33   # config.active_record.observers = :cacher, :garbage_collector, :forum_observer
34   config.action_controller.session = { :key => "myapp_session", :secret => "3456fa3a310cd61bc76056e67"
35   # Set Time.zone default to the specified zone and make Active Record auto-convert to this zone.
36   # Run "rake -D time" for a list of tasks for finding time zone names.
37   config.time_zone = 'UTC'
38
39   # The default locale is :en and all translations from config/locales/*.rb,yml are auto loaded.
40   # config.i18n.load_path += Dir[Rails.root.join('my', 'locales', '*.rb,yml')]
41   # config.i18n.default_locale = :de
```

cache-money-example

app

config

boot.rb

database.yml

environment.rb

environments

development.rb

production.rb

test.rb

initializers

backtrace_silencers.rb

cache-money.rb

inflections.rb

mime_types.rb

new_rails_defaults.rb

session_store.rb

locales

routes.rb

db

doc

lib

log

public

Rakefile

README

script

test

tmp

vendor

gems

nkallen-cache-money-0.2.5

specification

config

db

lib

README

spec

TODO

UNSUPPORTED_FEATURES

plugins

Line: 1 Column: 18 Ruby on Rails Soft Tabs: 2

Rails Templates



restful_authentication

will_paginate

rspec

role_requirement

paperclip

```
# template.rb
```

```
run "rm public/index.html"
```

```
generate(:scaffold, "person name:string")
```

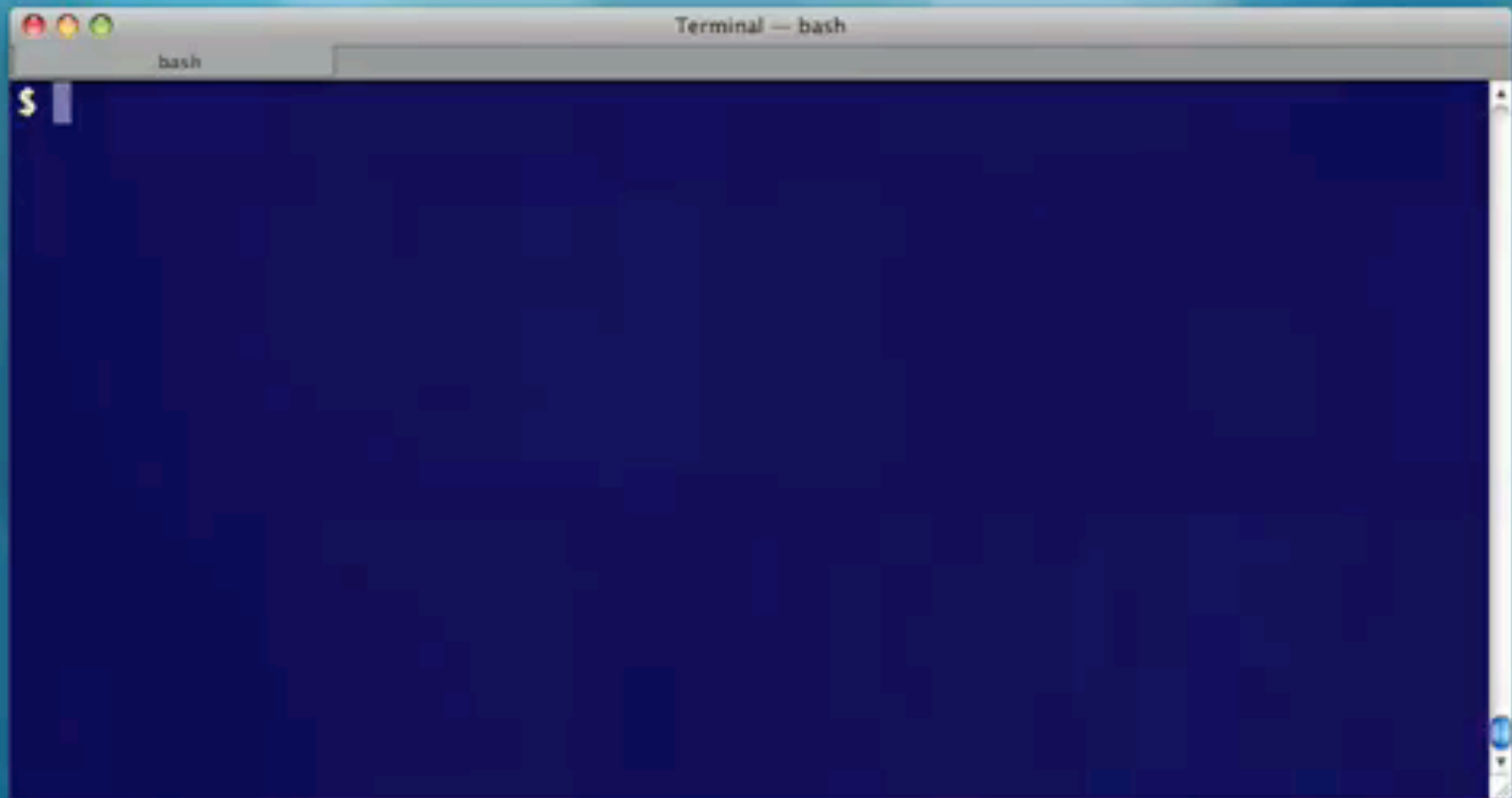
```
route "map.root :controller => 'people'"
```

```
rake("db:migrate")
```

```
git :init
```

```
git :add => "."
```

```
git :commit => "-a -m 'Initial commit'"
```

Rails Template Example

Template Functions

- Start a repository
- Commit to the repository
- Manipulate files
- Add gems and plugins
- Run rake tasks

<http://youvegotrails.com>

A Rails App in Seconds!

Plugins

☐ asset_packager

☐ attachment_fu

☐ authlogic

☐ exception_notifier

☐ hoptoad_notifier

☐ open_id_authentication

☐ paperclip

☐ restful_authentication

☐ role_requirement

☐ rspec

☐ ssl_requirement

▶ Gems

▶ Commands

Download

feedback

Rails Templates



<http://youvegotrails.com>

<http://github.com/jeremymcanally/rails-templates>

<http://www.railsboost.com>



BLAME

THE SECRET TO SUCCESS IS KNOWING WHO TO BLAME FOR YOUR FAILURES.

www.despair.com


```
if home_phone && home_phone.size > 0
  do_something
end
if work_phone && work_phone.size > 0
  do_something_else
end
if mobile_phone && mobile_phone.size > 0
  do_another_thing
end
```

Confessions of a Ruby Sadist

`sudo gem install flay`

Flay analyzes ruby code for structural similarities. Differences in literal values, names, whitespace, and programming style are all ignored.

Code that flay reports as similar is a good candidate for refactoring.

Try this:

```
% sudo gem install flay  
% flay lib/*.rb
```

and see what you find.



Flay Output

Matches found in :if (mass = 92)

file.rb:2

file.rb:5

file.rb:8

Flog

Confessions of a Ruby Sadist

`sudo gem install flog`

Flog shows you the most torturous code you wrote. The more painful the code, the higher the score. **The higher the score, the harder it is to test.**

Run it against your best stuff. I *double-dog* dare you.

Flog essentially scores an ABC metric: Assignments, Branches, Calls, with particular attention placed on calls.

Run flog on all your code. Try this:

```
find lib -name \*.rb | xargs flog
```



```

class Test
  def blah
    a = eval "1+1"
    if a == 2 then
      puts "yay"
    end
  end
end

```

Is seen by
flog as:

```

class Test
  def blah          # 11.2 =
    a = eval "1+1" # 1.2 + 6.0 +
    if a == 2 then # 1.2 + 1.2 + 0.4 +
      puts "yay"   # 1.2
    end
  end
end

```

and reported
as:

```

Test#blah: (11.2)
  6.0: eval
  1.2: branch
  1.2: ==
  1.2: puts
  1.2: assignment
  0.4: lit_fixnum

```


Roodi





Oh My Goth! ...the APDT's in town.

Roodi Results

[Roodi](#) parses your Ruby code and warns you about design issues you have based on the checks that it has configured.

File Path	Warning
app/controllers/tree_controller.rb:	Found = in conditional. It should probably be an ==
app/controllers/tree_controller.rb:	Found = in conditional. It should probably be an ==
app/controllers/tree_controller.rb:	Found = in conditional. It should probably be an ==
app/helpers/application_helper.rb:	Found = in conditional. It should probably be an ==
Found 4 errors.:	



```
class CsvWriter
  def write_line(fields)
    if (fields.length == 0)
      puts
    else
      write_field(fields[0])
      1.upto(fields.length-1) do |i|
        print ","
        write_field(fields[i])
      end
      puts
    end
  end
end

#...
end
```



```
$ reek csv_writer.rb
```

```
CsvWriter#write_line
```

calls fields.length multiple times
(Duplication)

```
CsvWriter#write_line
```

has approx 6 statements (Long
Method)

```
CsvWriter#write_line/block
```

has the variable name
'i' (Uncommunicative Name)

```
$ reek csv_writer.rb
```

```
CsvWriter#write_line
```

calls fields.length multiple times
(Duplication)

```
CsvWriter#write_line
```

has approx 6 statements (Long
Method)

```
CsvWriter#write_line/block
```

has the variable name
'i' (Uncommunicative Name)


```
$ reek csv_writer.rb
```

```
CsvWriter#write_line
```

calls fields.length multiple times
(Duplication)

```
CsvWriter#write_line
```

has approx 6 statements (Long
Method)

```
CsvWriter#write_line/block
```

has the variable name
'i' (Uncommunicative Name)

```
$ reek csv_writer.rb
```

```
CsvWriter#write_line
```

calls fields.length multiple times
(Duplication)

```
CsvWriter#write_line
```

has approx 6 statements (Long
Method)















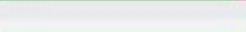
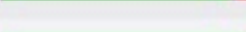




```
CsvWriter#write_line/block
```

has the variable name
'i' (Uncommunicative Name)

Metric FU

C0 code coverage information

Generated on Thu Mar 26 11:28:05 -0400 2009 with [rcov 0.8.1.2](#)

Name	Total lines	Lines of code	Total coverage	Code coverage
TOTAL	766	608	85.4% 	82.6% 
app/controllers/ssh_keys_controller.rb	87	61	33.3% 	14.8% 
app/helpers/commits_helper.rb	32	30	34.4% 	30.0% 
app/controllers/users_controller.rb	23	20	69.6% 	65.0% 
app/controllers/forks_controller.rb	22	19	77.3% 	73.7% 
app/helpers/repository_helper.rb	40	32	85.0% 	81.2% 
app/helpers/application_helper.rb	42	35	88.1% 	85.7% 
app/controllers/tree_controller.rb	61	52	90.2% 	88.5% 
app/models/project.rb	144	115	97.2% 	96.5% 
app/controllers/application_controller.rb	5	5	100.0% 	100.0% 

RCov

Source Control Churn Results

Files that change a lot in your project may be bad a sign. This task uses your source control log to identify those files.

File Path	Times Changed
test/test_helper.rb	20
config/environment.rb	18
test/unit/project_test.rb	17
config/routes.rb	16
app/models/project.rb	16
test/integration/browse_repository_test.rb	15
app/views/tree/show.html.erb	15
test/integration/projects_test.rb	12
app/helpers/repository_helper.rb	11
app/views/projects/index.html.erb	9
db/schema.rb	9
app/controllers/tree_controller.rb	8
app/helpers/application_helper.rb	8
test/unit/user_test.rb	8
app/views/projects/show.html.erb	7
app/models/user.rb	7

Churn

Flog Results

Flog measures code complexity.

File	Total score	Methods	Average score	Highest score
repository_helper.rb	102.50	2	51.30	54.30
projects_controller.rb	113.00	4	19.10	28.80
forks_controller.rb	36.40	1	28.70	28.70
ssh_keys_controller.rb	100.20	3	22.17	27.30
users_controller.rb	31.10	1	26.10	26.10
commits_helper.rb	31.80	1	25.50	25.50
tree_controller.rb	94.00	3	20.63	25.30
project.rb	198.40	7	17.16	24.10
files_controller.rb	19.20	1	19.20	19.20
key_generator.rb	30.60	2	10.20	13.60
application_helper.rb	43.50	3	11.70	13.00

Flog

Flay Results

Flay analyzes ruby code for structural similarities.

Files	Matches
app/controllers/projects_controller.rb:53 app/controllers/projects_controller.rb:68 app/controllers/ssh_keys_controller.rb:53 app/controllers/ssh_keys_controller.rb:70 app/controllers/users_controller.rb:18	1) Similar code found in :block (mass = 115)
app/controllers/projects_controller.rb:29 app/controllers/ssh_keys_controller.rb:28	2) Similar code found in :defn (mass = 50)

Generated on Thu Mar 26 11:28:29 -0400 2009

Flay

Reek Results

[Reek](#) detects common code smells in ruby code.

File Path	Code Smell
"app/controllers/commits_controller.rb" -- 1 warnings:	[Nested Iterators] CommitsController#index has nested iterators
"app/controllers/files_controller.rb" -- 1 warnings:	[Long Method] FilesController#show has approx 6 statements
"app/controllers/forks_controller.rb" -- 1 warnings:	[Long Method] ForksController#create has approx 7 statements
"app/controllers/projects_controller.rb" -- 12 warnings:	[Duplication] ProjectsController#create calls format.html more than once [Duplication] ProjectsController#create calls format.xml more than once [Long Method] ProjectsController#create has approx 7 statements [Nested Iterators] ProjectsController#create has nested iterators [Nested Iterators] ProjectsController#destroy has nested iterators [Nested Iterators] ProjectsController#index has nested iterators [Nested Iterators] ProjectsController#new has nested iterators [Nested Iterators] ProjectsController#show has nested iterators [Duplication] ProjectsController#update calls format.html more than once [Duplication] ProjectsController#update calls format.xml more than once [Long Method] ProjectsController#update has approx 6 statements [Nested Iterators] ProjectsController#update has nested iterators

Reek

Rails . Cache



Read

<http://www.flickr.com/photos/swamibu/2868288357/>

A close-up photograph of a fountain pen with a clear barrel and a silver-colored nib, resting on a piece of lined paper. The paper has handwritten text in cursive. In the upper right corner, there is a dark gray rectangular box with a white border containing the word "Write" in white, bold, sans-serif font.

Write

<http://www.flickr.com/photos/churl/250235189/>

A close-up, macro photograph of a single computer keyboard key. The key is a light gray or off-white color with a slightly textured surface. It has a rectangular shape with rounded corners. The word "Del" is printed in a clean, white, sans-serif font in the center of the key. The background is dark and out of focus, showing other keys of the keyboard. The lighting is soft, highlighting the texture of the key and the clarity of the text.

Del

<http://www.flickr.com/photos/virgu/12496426/>

Rails.Cache

```
module ActiveSupport
  module Cache
    class Store
      cattr_accessor :logger

      def read(key, options = nil)
        log("read", key, options)
      end

      def write(key, value, options = nil)
        log("write", key, options)
      end

      def delete(key, options = nil)
        log("delete", key, options)
      end
    end
  end
end
```

Code deleted for brevity

Rails.cache

```
require 'memcache'
```


Rails.Cache

```
require 'memcache'
```

```
module ActiveSupport
```

```
  module Cache
```

```
    class MemCacheStore < Store
```

```
      module Response # :nodoc:
```

```
        STORED = "STORED\r\n"
```

```
        NOT_STORED = "NOT_STORED\r\n"
```

```
        EXISTS = "EXISTS\r\n"
```

```
        NOT_FOUND = "NOT_FOUND\r\n"
```

```
        DELETED = "DELETED\r\n"
```

```
      end
```

```
    def initialize(*addresses)
```

```
      addresses = addresses.flatten
```

```
      options = addresses.extract_options!
```

```
      addresses = ["localhost"] if addresses.empty?
```

```
      @addresses = addresses
```

```
      @data = MemCache.new(addresses, options)
```

```
      extend Strategy::LocalCache
```

```
    end
```

Code deleted for brevity

Rails.Cache

```
module ActiveSupport
  module Cache
    class MemCacheStore < Store
      ...

      def read(key, options = nil) # :nodoc:
        super
        @data.get(key, raw?(options))
      rescue MemCache::MemCacheError => e
        logger.error("MemCacheError (#{e}): #{e.message}")
        nil
      end
    end
  end
end
```

Code deleted for brevity

How to Cache Anything With ActiveSupport

Wherein our hero shows how to magically cache (almost) anything using ActiveSupport's built-in caching.

Posted April 27, 2009

We use a REST-like web service to communicate with a large Endeca index. This web service is stateless and uses JSON. Unfortunately, its performance leaves much to be desired and the server does not cache requests. Fortunately, even with its weak implementation of REST, the service's statelessness makes it trivial to add a caching layer to our client.

Rather than write our own caching solution, we looked to reuse one that already exists. No NIH for us. The caching provided by ActiveSupport::Cache turned out to be perfect for our needs and darn simple to implement. The end result is that web services requests that took from 200ms to 500ms and sometimes more (I told you it was slow) are now cache retrievals that take from 2ms to 5ms. That's **10,000%** faster. From 20 lines of code. With all the power and flexibility of the caching already built in to Rails. Winnage.

A Brief Review of ActiveSupport Caching

ActiveSupport caching is used by Rails to provide page, action, fragment and SQL caching. With SQL caching, the results of a SQL query are cached in memory for a given action so that repeated queries within an action hit the cache rather than the database. The Caching With Rails guide has a section on SQL Caching that provides a brief overview of the concept. For a more thorough introduction to caching in general, check out Ryan Tomayko's "Things Caches Do". It has sweet illustrations.


```
module Endeca
```

```
  class << self
```

```
    def perform_caching?; !@cache_store.nil? end
```

```
    def cache_store; @cache_store end
```

```
    def cache_store=(store_option)
```

```
      @cache_store = ActiveSupport::Cache.lookup_store(store_option)
```

```
    end
```

```
  end
```

```
  module Caching
```

```
    def self.included(base)
```

```
      base.alias_method_chain :get_response, :caching
```

```
    end
```

```
    private
```

```
    def get_response_with_caching
```

```
      return get_response_without_caching unless Endeca.perform_caching?
```

```
      fetch { get_response_without_caching }
```

```
    end
```

```
    def cache_key; Digest::SHA1.hexdigest uri.to_s end
```

```
    def fetch(&block); Endeca.cache_store.fetch(cache_key, &block) end
```

```
  end
```

```
end
```


Cucumber



Behaviour Driven Development
with elegance and joy
and vegetables

Scenario: Giving a presentation
at Railsconf

Given: I am a presenter at
Railsconf

And:

I don't want to
offend anyone

When:

I give my
presentation

Then:

It should not have
any pornographic
images

FAIL



When the owner's away, the fun begins for the furniture.

[BACK TO HOME](#)

[NEXT ==>](#)



"Is it hot in here, baby, or is it just you?"



**"Are you looking for loose change, baby, or are you just
happy to see me? Ooooh!"**



Somebody's gonna need re-upholstering after this hot time!



Lance and Debbi get into a Louis the LXIX position!

Slide made by Gregg Pollack



**Ever see two chairs goin' at it like this?! Ooh Yeah!
Somebody get the hose!**

Slide made by Gregg Pollack, officially supported by the Rails Activists

Scenario: Giving a presentation at Railsconf

Given: I am a presenter at Railsconf

And: I don't want to offend anyone

When: I give my presentation

Then: It should not have any pornographic images

Scenario:	Giving a keynote at Railsconf
Given:	I am incredibly awesome
And:	I want everyone to know
And:	I work 4 hours a week
And:	I have recorded every work-out since I was 18 years old
When:	I bench-press 450 pounds
Then:	Did I mention I was incredibly awesome?
Also:	I outsourced making this slide





FakeWeb

Files	Classes	Methods
CHANGELOG LICENSE.txt README.rdoc lib/fake_web.rb lib/fakeweb.rb	FakeWeb FakeWeb::NetConnectNotAllowedError	allow_net_connect= allow_net_connect? clean_registry (Fake register_uri (FakeW registered_uri? (Fak

README.rdoc

Path: README.rdoc

Last Update: Sat Apr 11 22:26:34 -0700 2009

FakeWeb

[FakeWeb](#) is a helper for faking web requests in Ruby. It works at a global level, without modify

Installation

The latest release of [FakeWeb](#) is once again available from your friendly RubyForge mirror. Just

```
sudo gem install fakeweb
```

Note: the gem was previously available as [FakeWeb](#) (capital letters), but now all versions are sin
have any old [FakeWeb](#) gems lying around, remove them: `sudo gem uninstall FakeWeb`

Help and discussion

RDocs for the current release are available at fakeweb.rubyforge.org.

There's a mailing list for questions and discussion at groups.google.com/group/fakeweb-users.

Spike

Request Details

URL

http://replay.it/users/matt/messages/36

Session

BAh7CDoMY3NyZi9pZCllM2Y2YzY4MTUzYjQyNmEzNGVhODdlYjBhMjBzTE11

Filter

Parameter

Value

user_id

matt

controller

messages

id

36

authenticity_token

e4a509255f1dea0e60b6e07ff2...

_method

delete

action

destroy

Renders

messages/destroy

(

"Processing MessagesController#destroy (for 84.12.60.139 at 2009-02-16 20:38:21) [DELETE]".

"Session ID: BAh7CDoMY3NyZi9pZCllM2Y2YzY4MTUzYjQyNmEzNGVhODdlYjBhMjBzTE11".

NzY6CXVzZXJpCCEIKZmxhc2hJQzonQWN0aW9uQ29udHJvbGxlcjo6Rmxhc2g6,

"OkZsYXNoSGFzaHsABjQKHVZzZWR7AA==--cfe20bd29eeb8d4fded284e04e6a92134f8ca18".

"Parameters: ({'action'=>'destroy', '_method'=>'delete', 'authenticity_token'=>'e4a509255f1dea0e60b6e07ff20f6fa23e49a81V', 'id'=>'36', 'controller'=>'messages', 'user_id'=>'matt'})".

"Rendering messages/destroy".

"Completed in 0.02000 (50 reqs/sec) | Rendering: 0.00800 (40%) | DB: 0.00800 (39%) | 200 OK [http://replay.it/users/matt/

)

Spike

Q- BAh7CDoMY3NyZi9pZCllM2Y2YzY4MTUzYjQyNmEzNGVhODdlYjBhMjBzTE11

DateTime	Filter	Status	Method	Client	Controller	Action	Req/s	Time/s	Render/s	DB/s
20:41:21 16/02		200	GET	84.12.60.139	PlaylistsController	show	4	0.216	0.184	0.008
20:41:17 16/02		200	GET	84.12.60.139	ContentsController	show	3	0.312	0.212	0.052
20:41:15 16/02		200	GET	84.12.60.139	PlaylistsController	index	22	0.044	0.02	0.016
20:41:13 16/02		200	GET	84.12.60.139	PlaylistsController	show	4	0.224	0.188	0.004
20:40:37 16/02		200	GET	84.12.60.139	ContentsController	show	1	0.528	0.156	0.056
20:40:31 16/02		200	GET	84.12.60.139	ContentsController	show	3	0.256	0.188	0.028
20:40:30 16/02		200	GET	84.12.60.139	PlaylistsController	index	20	0.048	0.024	0.02
20:40:29 16/02		200	GET	84.12.60.139	PlaylistsController	show	5	0.172	0.024	0.008
20:40:25 16/02		200	GET	84.12.60.139	ContentsController	show	9	0.1	0.016	0.04
20:40:24 16/02		200	GET	84.12.60.139	PlaylistsController	index	19	0.052	0.024	0.016
20:40:23 16/02		200	GET	84.12.60.139	PlaylistsController	show	7	0.14	0.024	0.012
20:40:18 16/02		200	GET	84.12.60.139	ContentsController	show	8	0.12	0.016	0.036
20:40:17 16/02		200	GET	84.12.60.139	PlaylistsController	index	10	0.092	0.052	0.032
20:40:16 16/02		200	GET	84.12.60.139	PlaylistsController	show	5	0.176	0.032	0.008
20:40:13 16/02		200	GET	84.12.60.139	ContentsController	show	3	0.316	0.24	0.048
20:40:12 16/02		200	GET	84.12.60.139	PlaylistsController	index	4	0.232	0.028	0.016
20:40:11 16/02		200	GET	84.12.60.139	PlaylistsController	show	3	0.252	0.156	0.032
20:40:07 16/02		200	GET	84.12.60.139	ContentsController	show	1	0.516	0.032	0.076
20:40:05 16/02		200	GET	84.12.60.139	PlaylistsController	index	19	0.052	0.032	0.016
20:40:03 16/02		200	GET	84.12.60.139	PlaylistsController	show	1	0.988	0.788	0.028
20:38:41 16/02		200	GET	84.12.60.139	ContentsController	show	1	0.724	0.032	0.032
20:38:32 16/02		200	GET	84.12.60.139	PlaylistsController	index	22	0.044	0.02	0.02
20:38:31 16/02		302	GET	84.12.60.139	HomepagesController	index	249	0.004	0	0
20:38:31 16/02		200	GET	84.12.60.139	DashboardsController	show	62	0.016	0.012	0
20:38:29 16/02		200	DELETE	84.12.60.139	MessagesController	destroy	49	0.02	0.008	0.004
20:38:24 16/02		200	GET	84.12.60.139	MessagesController	index	35	0.028	0.016	0
20:38:22 16/02		200	GET	84.12.60.139	MessagesController	index	2	0.388	0.376	0.008
20:38:21 16/02		200	DELETE	84.12.60.139	MessagesController	destroy	50	0.02	0.008	0.008
20:38:18 16/02		200	DELETE	84.12.60.139	MessagesController	destroy	49	0.02	0.008	0.004
20:38:15 16/02		200	DELETE	84.12.60.139	MessagesController	destroy	12	0.08	0.004	0.024
20:38:12 16/02		200	DELETE	84.12.60.139	MessagesController	destroy	35	0.028	0.008	0.004
20:38:09 16/02		200	DELETE	84.12.60.139	MessagesController	destroy	41	0.024	0.004	0.012
20:38:02 16/02		200	GET	84.12.60.139	MessagesController	index	1	0.64	0.584	0.16
20:37:57 16/02		200	GET	84.12.60.139	PlaylistsController	index	3	0.328	0.128	0.1
20:37:52 16/02		200	GET	84.12.60.139	DashboardsController	show	0	2.524	2.208	0.408
20:36:07 16/02		200	GET	84.12.60.139	PlaylistsController	index	19	0.052	0.032	0.012

87 requests

Ultrasphinx

- lib/ultrasphinx/search.rb
- lib/ultrasphinx/spell.rb
- ...ltrasphinx/ultrasphinx.rb

delta_index_present?
load_stored_procedure
log
options_for
say
verify_database_name

[illegible]

ActiveRecord
ActiveRecord::Base
Ultrasphinx
Ultrasphinx::Search

Module: Ultrasphinx

Child modules and classes

Module **Ultrasphinx::Spell**
Class **Ultrasphinx::Search**

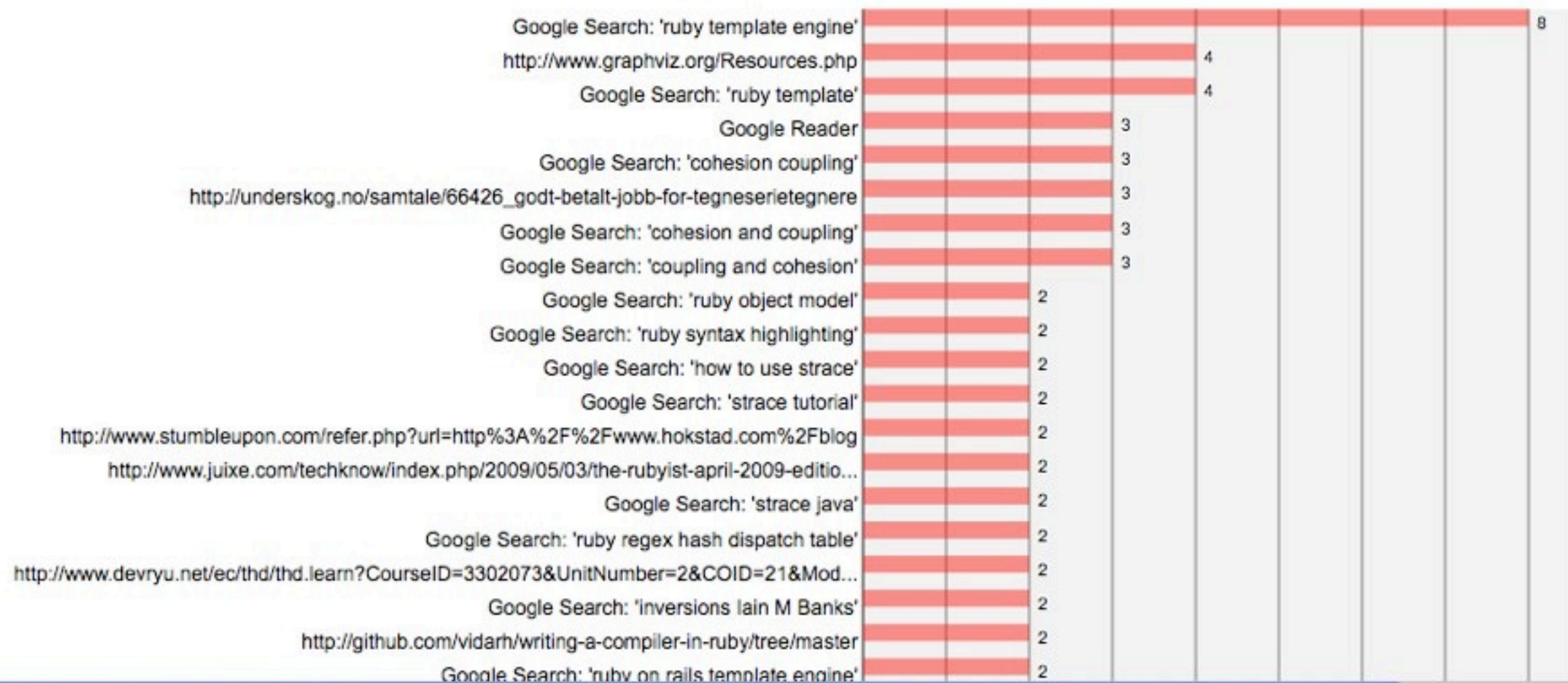
Constants

Name	Value
SUBDIR	"config/ultrasphinx"
DIR	"#{RAILS_ROOT}/#{SUBDIR}"
THIS_DIR	File.expand_path(File.dirname(__FILE__), DIR)
CONF_PATH	"#{DIR}/#{RAILS_ENV}.conf"
ENV_BASE_PATH	"#{DIR}/#{RAILS_ENV}_base"

Sliding Stats

Sliding Stats

Most recent referrers



Clearance



Sprinkle

Sprinkle Some Powder!

posted by crafterm, 27

Provisioning a brand new server or VPS slice can be quite tricky, tedious and time consuming, particularly if done manually with changing software versions and configurations.

In the Rails world, most of us are using virtual private servers which are instantiated from base operating system images, it takes only a few minutes to create a slice, however installing the rest of your server stack, be it Rails, Merb or another framework is where the work is. Provisioning in this sense, is installing all software required post operating system install.

Passenger stack, zero to hero in under five minutes

Scripts for [Sprinkle](#), the provisioning tool

[Watch the demo screen cast](#) of passenger-stack.

How to get your sprinkle on:

- Get a brand spanking new slice / host (Ubuntu please)
- Create yourself a user, add yourself to the /etc/sudoers file
- Set your slices url / ip address in deploy.rb (config/deploy.rb.example provided)
- Set username in config/deploy.rb if it isn't the same as your local machine (config/deploy.rb.e

From your local system (from the passenger-stack directory), run:

```
sprinkle -c -s config/install.rb
```


Spree

Shop by Brand

[Ruby](#)

[Ruby on Rails](#)

[Apache](#)

Shop by Category

[Clothing](#)

[Bags](#)

[Mugs](#)



Ruby on Rails Ringer
T-Shirt
\$17.99



Ruby on Rails Bag
\$22.99



Ruby on Rails Mug
\$13.99



Ruby on Rails Tote
\$15.99



Ruby on Rails Jr. Spaghetti
\$19.99



Apache Baseball Jersey
\$19.99

Webbrat

Example

```
class SignupTest < ActionController::IntegrationTest

  def test_trial_account_sign_up
    visit home_path
    click_link "Sign up"
    fill_in "Email", :with => "good@example.com"
    select "Free account"
    click_button "Register"
  end

end
```


Taps for Easy Database Transfers

TAPS DATABASES OPENSOURCE

WED FEB 11 16:19:28 -0800 2009

Migrating databases from one server to another is a pain: mysqldump on old server -> gzip -> scp big c -> gunzip -> mysql. It takes a long time, and is very manual and (and thus error-prone), and generally stink of “lame” hanging about it.

Ricardo Chimal, Blake Mizerany and I cooked up our attempt at a solution to this problem: Taps.

Taps is a temporary web service you run on a server that has access to the database you want to export. You can then run the client to connect to that service and pull data out of it in chunks. It works through firewalls, doesn't require a direct ssh connection, and - best of all - it's database independent. So you can export from a MySQL database




```
$ taps server mysql://root@localhost/mydb tmpuser tmppass  
= Sinatra/0.9.0.4 has taken the stage on 5000 for production with backup from Thin
```

(You'll need to either open port 5000 in your firewall, or use an Apache or Nginx to create a vhost for the temporary taps server.)

Now, on the ec2 instance, assuming you're using the raw port:

```
$ mysqladmin create mydb  
$ taps pull mysql://root@localhost/mydb http://tmpuser:tmppass@slicehost-box:5000  
Receiving schema from remote taps server  
Receiving data from remote taps server  
4 tables, 1,064 records  
widgets: 20% |=====| Time: 00:00:00
```


Photo Credits

Nine cents	Jonathan Pobre	http://www.flickr.com/photos/jonnny/255399662/
Impatience	mdezemery	http://www.flickr.com/photos/mdezemery/2190181769/
I will try fix you	Alfon...	http://www.flickr.com/photos/alfon18/2439293687/

Shameless
Self
Promotion

[home](#)[company](#)[portfolio](#)[speaking](#)[podcast](#)[hire us](#)

Rails Envy Podcast - Episode #077: 04/29/2009

by **Jason** on Apr 29, 2009

Episode 77.



Subscribe via iTunes - iTunes only link.

Download the podcast ~17:00 mins MP3.

Subscribe to feed via RSS by copying the link to your RSS Reader

- [Rack 1.0, a modular Ruby webserver interface](#)
- [Ruby Enterprise Edition 1.8.6-20090421](#)
- [Phusion Passenger 2.2.2](#)

HARVEST

Harvest is a web-based time tracking and invoicing application dedicated to helping freelancers and developers easily track hours and get paid faster. Simple and intuitive to use, it takes care of your numbers while you focus on delivering great work. Harvest's visual reporting shows you the health of your operation at-a-glance.

Use coupon code **RAILSENVY428** for \$10 off the first month of service with Harvest.

[Ads from Ruby Row](#)

Podcast

[RSS](#)[iTunes](#)[Archive](#)

Our Sponsor



RailsKits

The Rails Envy podcast is brought to you this week by Rails Kits. Rails Kits provides ready made and tested Rails code for you to use in your projects.. Check them out at RailsKits.com.

<http://railsenvy.com>

Screencasts

[Ruby 1.9 Package Deal](#)[Ruby 1.9 Part 1](#)[Ruby 1.9 Part 2](#)[Scaling Ruby](#)[Advanced ActiveRecord](#)[Ruby on Rails 2.2 Package Deal](#)[Ruby on Rails 2.2 Screencast](#)[Ruby on Rails 2.2 PDF](#)

Discussion/Support Mailing List

Email:

[Subscribe](#)[View Group](#)

Additional Rails Envy Media

- [Rails Envy Podcast](#)
- [Conference Coverage](#)



click image above for video preview

Ruby 1.9 Package Deal

What you need to know about Ruby 1.9

For just \$16 (best value) pick up both Part 1 and Part 2 of this Envycast by David Black. David teaches you all changes you need to know to migrate successfully to Ruby 1.9. See [Part 1](#) and [Part 2](#) for additional screenshots.

BONUS: Each purchase comes with a 40% off coupon for David's new book [The Well-Grounded Rubyist](#).

[Video Preview](#)**\$16.00**[Add to Cart](#)[See some Screenshots](#)

Ruby 1.9 Part 1

Taught by David A. Black

Covering what you need to know in Ruby 1.9 including:

- Changes with  and Symbols
- Working with
- Using the En
- New RubyGe

Running time: 41 minutes

QuickTime, Theora Ogg, and iPhone/iPod format

Get this Video with Part 2 for just \$8 dollars with the



Thank You

<http://railsenvy.com/railsconf09>