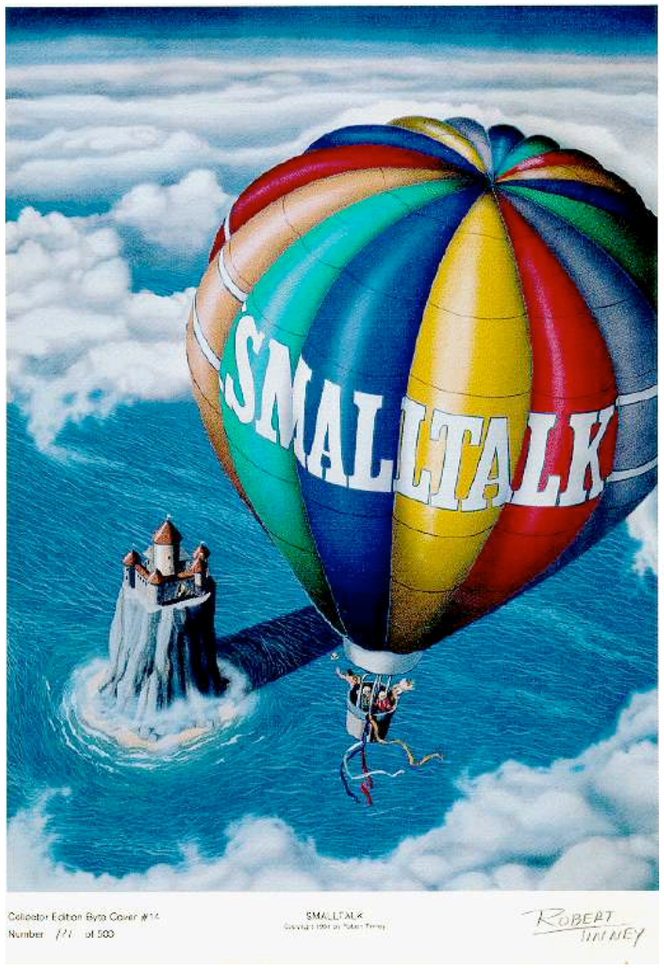
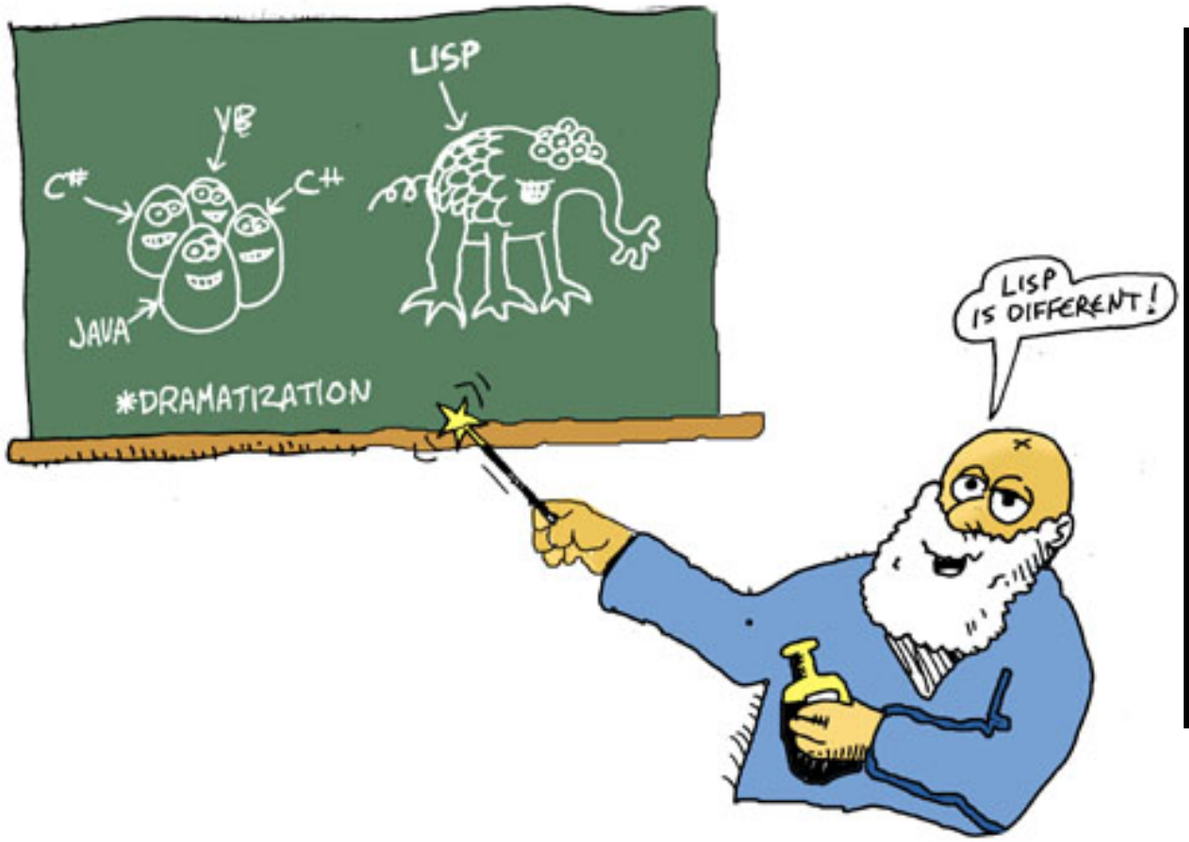
Ruby on Rails

Rails Released 1.0 Yesterday!

Brian McCallister

ApacheCon US 2005





Smalltalk + Lisp + Perl

Robert Tinney
www.tinney.net

Conrad Barski
www.lisperati.com

O'Reilly
www.perl.com

Without the...

- Weird GUI
- Parens
-

```
$~='`';$_=$:=$~|'%';$;=$^='/'|$~;$;++;$\=$~|"'";
$;++;$:.=++$;;$/=++$;;+$\++;$_.='#'|$~;
$,=++$/;$_."$\$^$\"";++$,;$_.='@'|'*'&~'!';$_."$,$;
$/$\"";$_.+='!.'|$~.$~;$_."$^$/$\$:$\"";
$_.='@'|':'&~'*';$_.=$:;$_.=$^&'|';$_.=$".$\;
$_.=+"$~$~$~"|'!#+';++$.;$.++;`$_$:,>&$.`;
```

```
["hello", "world"].each {|word| print word }
```

we love each!

```
printer_generator = lambda {|x| lambda { print x }}
hello = printer_generator.call "hello world"
hello.call
```

functions to generate functions to...

```
"world hello" =~ /(\w+)\s(\w+)/
print "#{$2} #{$1}"
```

Yeah, yeah, told you

+

+

John Long
www.wiseheartdesign.com

Sun
java.sun.com

PHP
www.php.net

Model-2 Web Framework

Object/Relational Mapping Library

SOAP Stack

SMTP/Email Library

Database Migration Tool

Deployment and Management Tool

Code Generator

Ecosystem

Even more! (watch out, Cocoon!)

Less Code

Convention over Configuration

Opinionated

Gifts

(Welcome to your running example)

But, first we need a project

```
brianm@kite:~/acon$
```

This Just Generated:

Project Hierarchy

Default Database Configs

with samples for major databases

Rakefile (Makefile)

System Test Harness

Unit Test Harness

Apache HTTPD Configs (.htaccess)

Additional Code Generation Scripts...

Final Screencast Goes Here

Active Record

You get the data from the database
and shake it all about...

Not Required!

One Row == One Instance

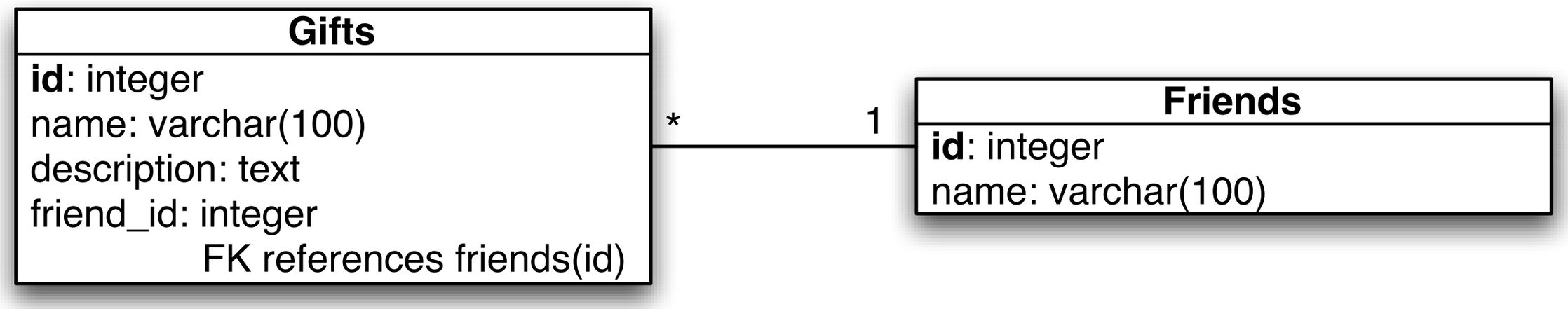
Dynamic Properties by Default

Embraces SQL

Including joins

PostgreSQL, MySQL, Oracle (OCI), DB2, Firebird, SQLite, SQLServer

I’ve been pestering the Derby folks ;-)



You want stuff.

Your friends check off what they will give you.

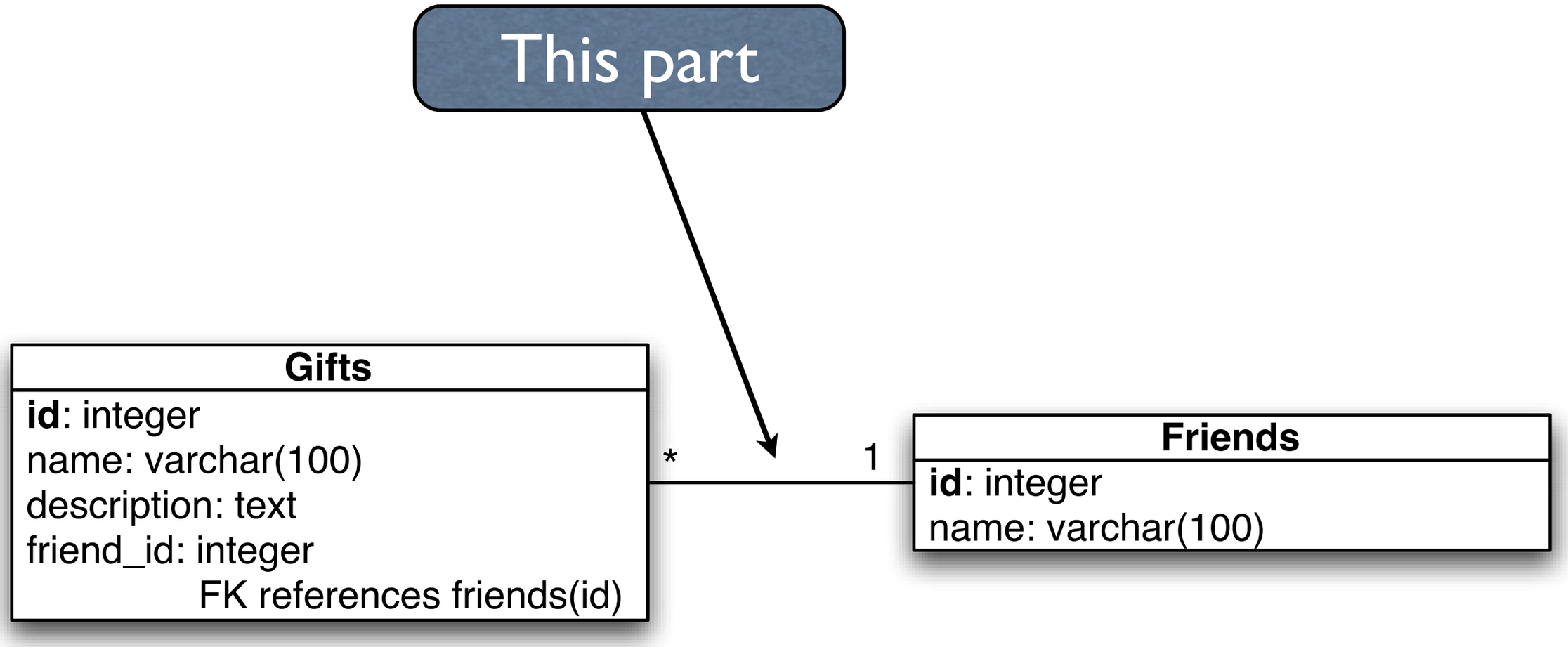
Your friends can see what they are giving you.

```
brianm@kite:~/acon/gifts$
```

```ruby
class Gift < ActiveRecord::Base
end




class Friend < ActiveRecord::Base
end
```

```ruby
class Gift < ActiveRecord::Base
  belongs_to :friend

  validates_presence_of :name
end



class Friend < ActiveRecord::Base
  has_many :gifts

  validates_presence_of :name
  validates_uniqueness_of :name
end
```

```ruby
class Gift < ActiveRecord::Base
  belongs_to :friend

  validates_presence_of :name
end



class Friend < ActiveRecord::Base
  has_many :gifts

  validates_presence_of :name
  validates_uniqueness_of :name
end
```


This part
Gifts
id: integer
name: varchar(100)
description: text
friend_id: integer
FK references friends(id)
*
1
Friends
id: integer
name: varchar(100)

```ruby
class Gift < ActiveRecord::Base
  belongs_to :friend

  validates_presence_of :name
end



class Friend < ActiveRecord::Base
  has_many :gifts

  validates_presence_of :name
  validates_uniqueness_of :name
end
```

```
brianm@kite:~/acon/gifts$
```

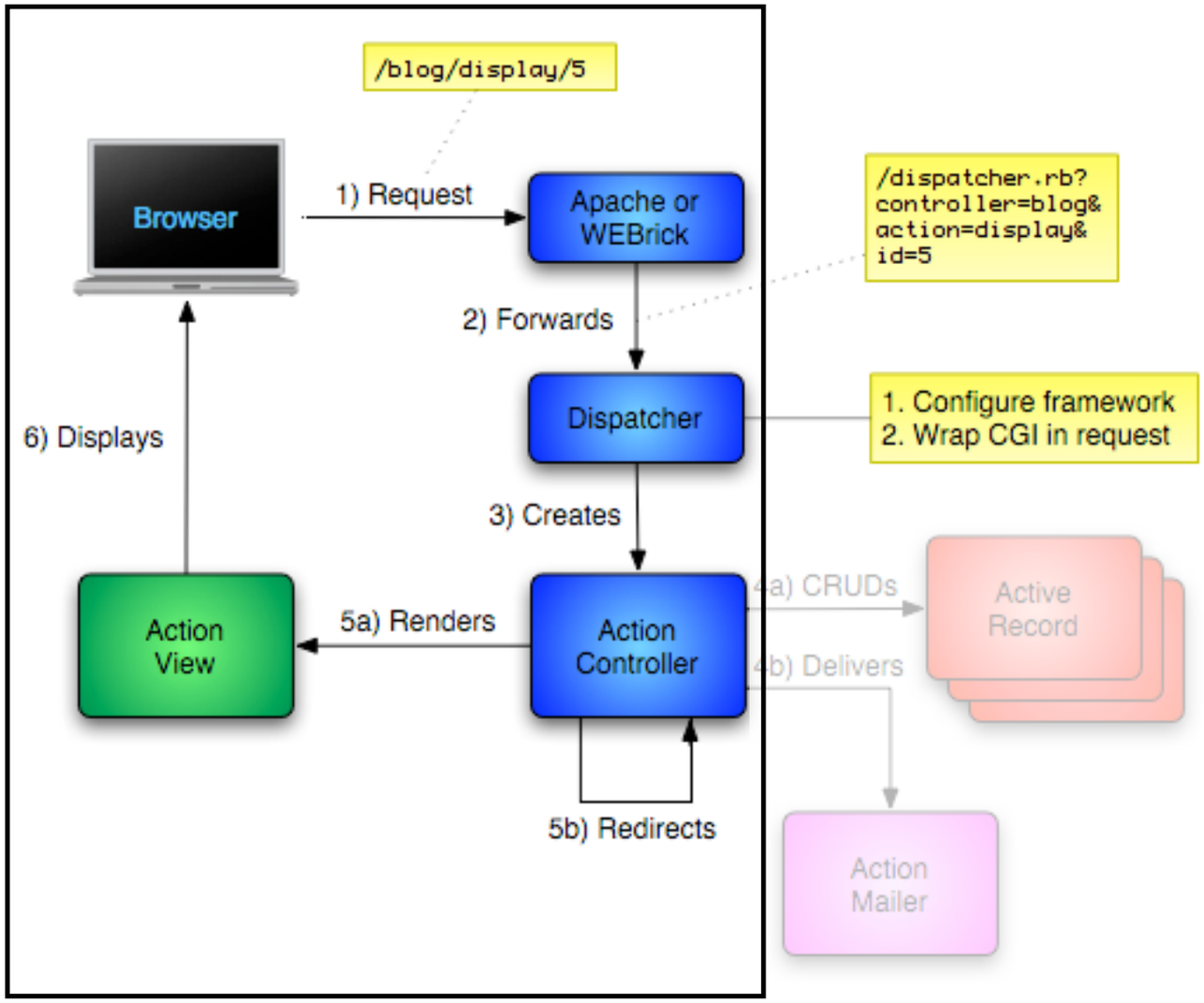
```
Gift.find :all,
          :order =>'name DESC',
          :limit => count,
          :include => [:friend]


Gift.find :first,
          :conditions => ['description like ?', '%me%'],
          :order => 'name DESC',
          :include => [:friend]


Gift.find_by_sql ["select g.* from gifts g where
                   g.friend_id < ?", 15]
```

```
Gift.find :all,
          :order =>'name DESC',
          :limit => count,
          :include => [:friend]


Gift.find :first,
          :conditions => ['description like ?', '%me%'],
          :order => 'name DESC',
          :include => [:friend]


Gift.find_by_sql ["select g.* from gifts g where
                   g.friend_id < ?", 15]
```

```
Gift.find :all,
          :order =>'name DESC',
          :limit => count,
          :include => [:friend]


Gift.find :first,
          :conditions => ['description like ?', '%me%'],
          :order => 'name DESC',
          :include => [:friend]


Gift.find_by_sql ["select g.* from gifts g where
                   g.friend_id < ?", 15]
```

Action Pack

That Web Stuff


/blog/display/5
Browser
1) Request
Apache or WEBrick
/dispatcher.rb?
controller=blog&
action=display&
id=5
2) Forwards
Dispatcher
1. Configure framework
2. Wrap CGI in request
6) Displays
3) Creates
Action View
5a) Renders
Action Controller
4a) CRUDs
Active Record
4b) Delivers
5b) Redirects
Action Mailer

```
brianm@kite:~/acon/gifts$
```

```
class PeopleController < ApplicationController

  def identify
  end
end
```

Controller

Action

```
class PeopleController < ApplicationController
  model :friend

  def identify
    if params[:friend]
      # if we have friend data in the request

      @friend = Friend.create_or_find params[:friend][:name]

      if @friend.errors.empty?
        # if there were no validation errors
        cookies[:name] = @friend.name
        redirect_to :controller => 'wishes',
                    :action => 'index'
      end
    end

    @friend ||= Friend.new
  end
end
```

Check Params

Success

```
class PeopleController < ApplicationController
  model :friend

  def identify
    if params[:friend]
      # if we have friend data in the request

      @friend = Friend.create_or_find params[:friend][:name]

      if @friend.errors.empty?
        # if there were no validation errors
        cookies[:name] = @friend.name
        redirect_to :controller => 'wishes',
                    :action => 'index'
      end
    end

    @friend ||= Friend.new
  end
end
```

Let Model Think

```
class Friend < ActiveRecord::Base
  has_many :gifts

  validates_presence_of :name
  validates_uniqueness_of :name

  def Friend.create_or_find name
    friend = find_by_name name
    logger.debug "Friend.create_or_find(#{name}) \
                  found [#{name}]"
    unless friend
      logger.debug "Friend.create_or_find(#{name}) \
                    creating new"
      friend = Friend.create :name => name
    end
    return friend
  end
end
```

```
class Friend < ActiveRecord::Base
  has_many :gifts

  validates_presence_of :name
  validates_uniqueness_of :name

  def Friend.create_or_find name
    friend = find_by_name name
    logger.debug "Friend.create_or_find(#{name}) \
                  found [#{name}]"
    unless friend
      logger.debug "Friend.create_or_find(#{name}) \
                    creating new"
      friend = Friend.create :name => name
    end
    return friend
  end
end
```

```
<%= error_messages_for 'friend' %>
<%= start_form_tag :action => 'identify' %>
  <table class="form">
    <tr>
      <td class="label">
        <label for="friend[name]">Your Name</label>
      </td>
      <td class="value">
        <%= text_field 'friend', 'name' %>
      </td>
    </tr>
    <tr>
      <td class="label"> </td>
      <td class="value">
        <input type="submit" value="Enter!"/>
      </td>
    </tr>
  </table>
<%= end_form_tag %>
```

```
<%= error_messages_for 'friend' %>
<%= start_form_tag :action => 'identify' %>
  <table class="form">
    <tr>
      <td class="label">
        <label for="friend[name]">Your Name</label>
      </td>
      <td class="value">
        <%= text_field 'friend', 'name' %>
      </td>
    </tr>
    <tr>
      <td class="label"> </td>
      <td class="value">
        <input type="submit" value="Enter!"/>
      </td>
    </tr>
  </table>
<%= end_form_tag %>
```

```
ActionController::Routing::Routes.draw do |map|

  # Ask people to identify themselves at the /hello
  map.connect 'hello', :controller => 'people',
                       :action => 'identify'

  # Pretty url for the most common thing
  map.connect 'wish', :controller => 'wishes',
                      :action => 'index'

  # The Default
  map.connect ':controller/:action/:id'
end
```

```
ActionController::

  # Ask people to identify themselves at the /hello
  map.connect 'hello', :controller => 'people',
                       :action => 'identify'

  # Pretty url for the most common thing
  map.connect 'wish', :controller => 'wishes',
                      :action => 'index'

  # The Default
  map.connect ':controller/:action/:id'
end
```

http://localhost/hello

http://localhost/people/identify

Ruby on Rails: Welcome aboard

http://localhost:3000/

Welcome aboard

You're riding the Rails!

About your application's environment

Getting started

Here's how to get rolling:

1. Create your databases and edit `config/database.yml`

   Rails needs to know your login and password.

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

   To see all available options, run it without parameters.

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

   Routes are setup in config/routes.rb.

Search the Rails site

Join the community

Ruby on Rails
Official weblog
Mailing lists
IRC channel
Wiki
Bug tracker

Browse the documentation

Rails API
Ruby standard library
Ruby core

Not Bad for Slowest Server Option

Some Fancy Stuff

or something like it

```
<%= javascript_include_tag 'prototype' %>
<%= javascript_include_tag 'prototype', 'effects' %>
<h1>Stuff I Want!</h1>
<script lang="text/javascript">
    function highlight_last() {
        new Effect.Highlight($('wish_list').lastChild);
    }
</script>
<%= form_remote_tag :url => { :action => 'gimme' },
                    :update => 'wish_list',
                    :position => :bottom,
                    :complete => 'highlight_last();'
                     %>
  <label for="gift_idea">Gift Idea</label>
  <%= text_field_tag :gift_idea %>
  <%= submit_tag 'Gimme!' %>
<%= end_form_tag %>
<ul id="wish_list">
    <% for gift in current_gift_list %>
        <%= render_partial 'gift', gift %>
    <% end %>
</ul>
```

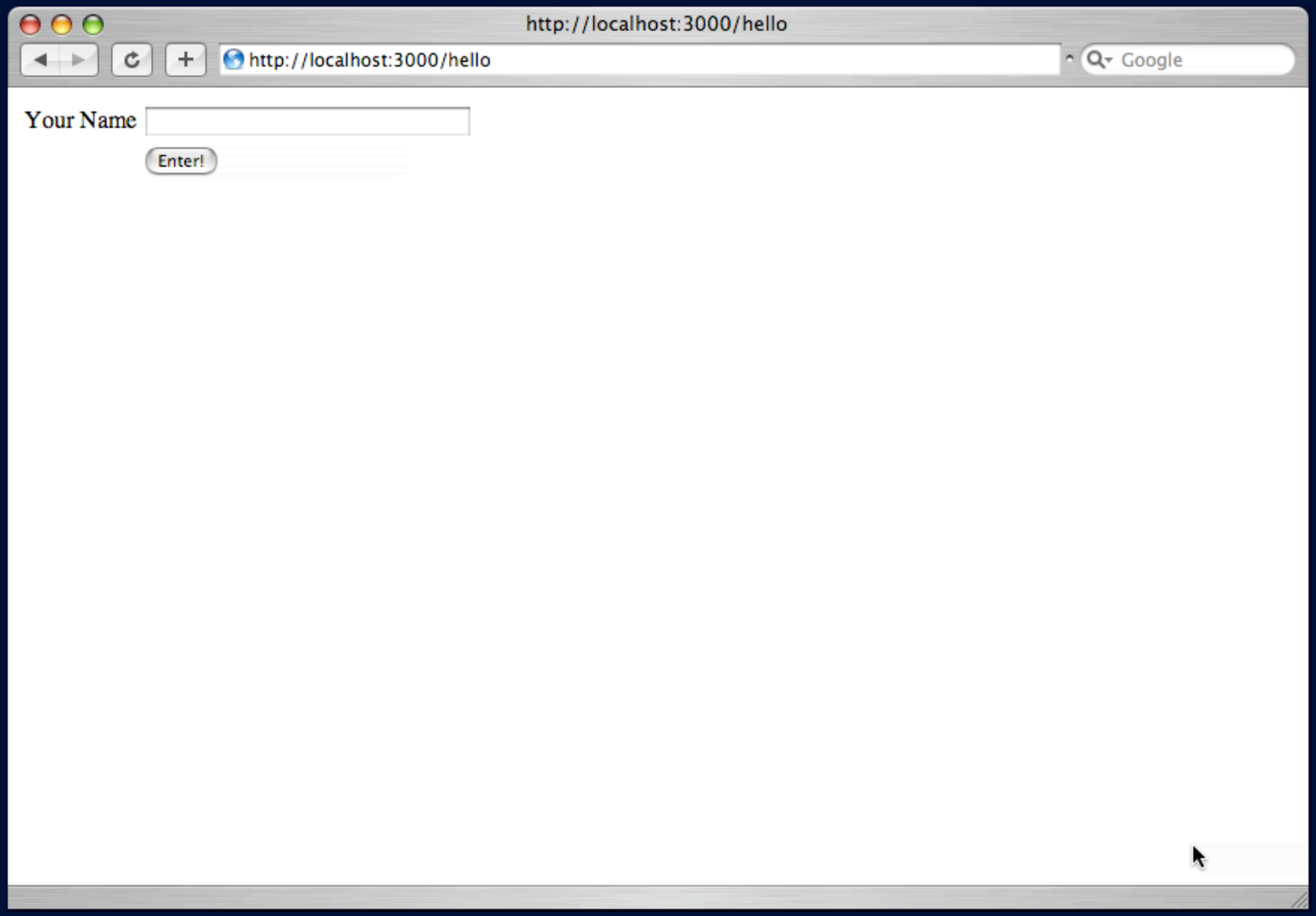
```
<%= javascript_include_tag 'prototype' %>
<%= javascript_include_tag 'prototype', 'effects' %>
<h1>Stuff I Want!</h1>
<script lang="text/javascript">
    function highlight_last() {
        new Effect.Highlight($('wish_list').lastChild);
    }
</script>
<%= form_remote_tag :url => { :action => 'gimme' },
                    :update => 'wish_list',
                    :position => :bottom,
                    :complete => 'highlight_last();'
                     %>
  <label for="gift_idea">Gift Idea</label>
  <%= text_field_tag :gift_idea %>
  <%= submit_tag 'Gimme!' %>
<%= end_form_tag %>
<ul id="wish_list">
    <% for gift in current_gift_list %>
        <%= render_partial 'gift', gift %>
    <% end %>
</ul>
```

```
<li><%= h @gift.name %></li>
```

```
class WishesController < ApplicationController
  model :gift

  def index
  end

  def gimme
    gift_idea = params[:gift_idea]
    if gift_idea
      f = Friend.find_by_name cookies[:name]
      gift = Gift.create :friend => f,
                         :name => gift_idea
      render :partial => 'gift', :object => gift
    else
      render :text => ''
    end
  end
end
```

```
class WishesController < ApplicationController
  model :gift

  def index
  end

  def gimme
    gift_idea = params[:gift_idea]
    if gift_idea
      f = Friend.find_by_name cookies[:name]
      gift = Gift.create :friend => f,
                         :name => gift_idea
      render :partial => 'gift', :object => gift
    else
      render :text => ''
    end
  end
end
```

```
class WishesController < ApplicationController
  model :gift

  def index
  end

  def gimme
    gift_idea = params[:gift_idea]
    if gift_idea
      f = Friend.find_by_name cookies[:name]
      gift = Gift.create :friend => f,
                         :name => gift_idea
      render :partial => 'gift', :object => gift
    else
      render :text => ''
    end
  end
end
```


http://localhost:3000/hello
http://localhost:3000/hello
Google
Your Name
Enter!

The first difference is that “enterprise software” costs more...

The second difference is that “enterprise software” doesn’t necessarily work...

--Kragen Sitaker

http://www.relevancellc.com/blogs/?p=36#comment-545

Rails generators create more code for testing than anything else.

```
$ find test/ -name *.rb -exec cat {} \; | wc -l
      84
$ find app/ -name *.rb -exec cat {} \; | wc -l
      52
$
```

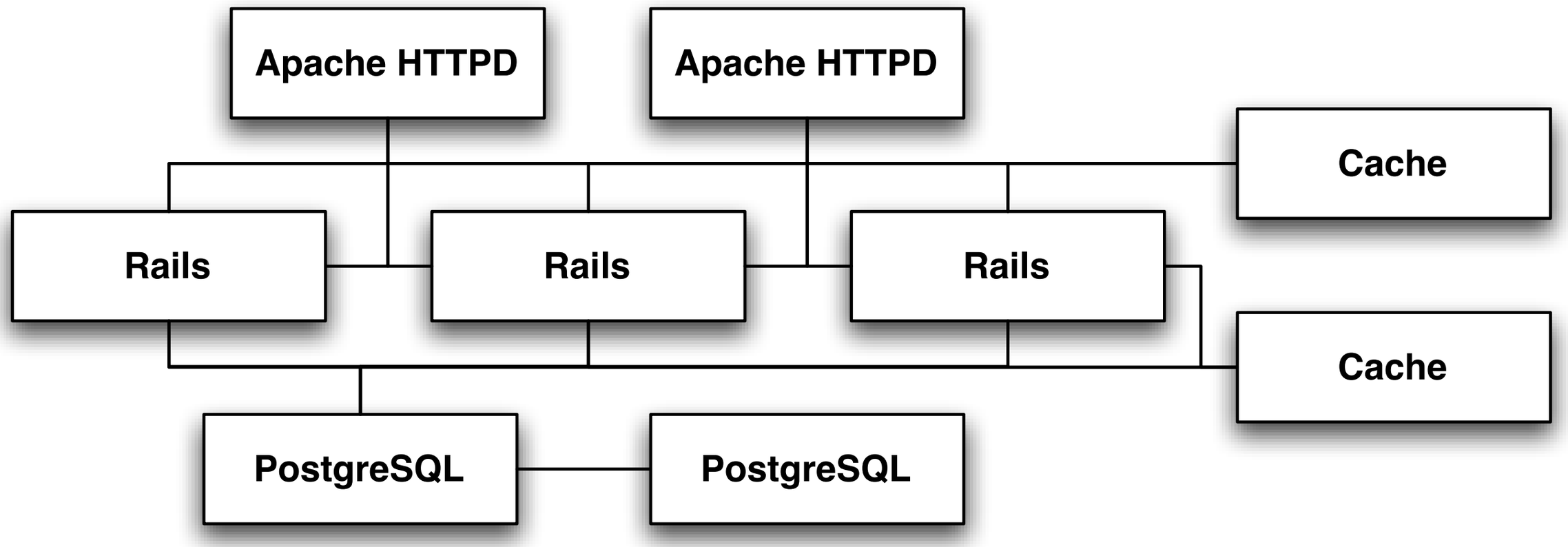
```
$ find test/ -name *.rb -exec cat {} \; | wc -l
      84
$ find app/ -name *.rb -exec cat {} \; | wc -l
      52
$
```

```
ActiveRecord::Schema.define() do

  create_table "friends", :force => true do |t|
    t.column "name", :string, :limit => 100, :null => false
  end

  add_index "friends", ["name"], :name =>"friends_name_key",
                                 :unique => true

  create_table "gifts", :force => true do |t|
    t.column "name", :string, :limit => 100, :null => false
    t.column "description", :text
    t.column "friend_id", :integer
  end

end
```

```
brianm@kite:~/acon/gifts$
```

```ruby
class Enemies < ActiveRecord::Migration
  def self.up
    create_table "enemies" do |t|
      t.column "name", :string
    end
  end

  def self.down
    drop_table "enemies"
  end
end
~
~
~
~
~
~
~
~
~
~
~
"db/migrate/001_enemies.rb" 11L, 185C
```

Switchtower


Apache HTTPD
Apache HTTPD
Cache
Rails
Rails
Rails
Cache
PostgreSQL
PostgreSQL

Apache HTTPD 1.3

- mod_fcgi

Apache HTTPD 2.0 (maybe 2.2)

- mod_scgi
- mod_proxy to Lighttpd

Lighttpd

- mod_fcgi

Apache D 3.0

- ???

Apache HTTPD 1.3
mod_fcgi

Apache HTTPD 2.0 (maybe 2.2)

- mod_scgi
- mod_proxy to Lighttpd

Lighttpd

- mod_fcgi

Apache D 3.0

- ???

We want to use Apache 2.x!

- Multi-processing model (like prefork)
- DB Connection per FCGI Process
- Remote FCGI instances
- Static and Dynamic Caching
- Easy to interface with C
  - (Almost as easy to interface with OCaml)

Ruby

http://www.ruby-lang.org/

Ruby on Rails

http://www.rubyonrails.org/

Try Ruby Online!

http://tryruby.hobix.com/

Why's Poignant Guide to Ruby

http://poignantguide.net/

Ruby Documentation

http://www.ruby-doc.org/

That's (almost) all folks!

Brian McCallister
brianm@apache.org
http://kasparov.skife.org/ac-us-04-ror.pdf
http://kasparov.skife.org/ac-us-04-ror.swf

Thank You:
Sun, Black Hat, Intel, Covalent, JCP, Azul Systems, CafeSoft, IBM, LogicBlaze, Rogue Wave, Thawte, Chariot Solutions, Byte.com, and C/C++ User's Journal

Interfacing to C

Not rails, but is important to know about!

```
require 'dl/import'

module XSLT
  extend DL::Importer
  dlload "libxslt.so"
  extern "void *xsltParseStylesheetFile(char*)"
  extern "void *xsltApplyStylesheet(void*, void*, void[])"
  extern "void xsltSaveResultToFd(int, void*, void*)"
  extern "void xsltSaveResultToFile(void*, void*, void*)"
  extern "void xsltFreeStylesheet(void*)"
end

module XML
  extend DL::Importer
  dlload "libxml2.so"
  extern "void *xmlParseFile(char*)"
  extern "void xmlFreeDoc(void*)"
end

cur = XSLT.xsltParseStylesheetFile("sample.xsl")
doc = XML.xmlParseFile("sample.xml")
res = XSLT.xsltApplyStylesheet(cur, doc, nil)

cur.free = XSLT["xsltFreeStylesheet"]
doc.free = XML["xmlFreeDoc"]
res.free = XML["xmlFreeDoc"]

XSLT.xsltSaveResultToFile(DL::CPtr[$stdout], res, cur)
```

```
require 'dl/import'

module XSLT
  extend DL::Importer
  dlload "libxslt.so"
  extern "void *xsltParseStylesheetFile(char*)"
  extern "void *xsltApplyStylesheet(void*, void*, void[])"
  extern "void xsltSaveResultToFd(int, void*, void*)"
  extern "void xsltSaveResultToFile(void*, void*, void*)"
  extern "void xsltFreeStylesheet(void*)"
end

module XML
  extend DL::Importer
  dlload "libxml2.so"
  extern "void *xmlParseFile(char*)"
  extern "void xmlFreeDoc(void*)"
end

cur = XSLT.xsltParseStylesheetFile("sample.xsl")
doc = XML.xmlParseFile("sample.xml")
res = XSLT.xsltApplyStylesheet(cur, doc, nil)

cur.free = XSLT["xsltFreeStylesheet"]
doc.free = XML["xmlFreeDoc"]
res.free = XML["xmlFreeDoc"]

XSLT.xsltSaveResultToFile(DL::CPtr[$stdout], res, cur)
```

```
require 'dl/import'

module XSLT
  extend DL::Importer
  dlload "libxslt.so"
  extern "void *xsltParseStylesheetFile(char*)"
  extern "void *xsltApplyStylesheet(void*, void*, void[])"
  extern "void xsltSaveResultToFd(int, void*, void*)"
  extern "void xsltSaveResultToFile(void*, void*, void*)"
  extern "void xsltFreeStylesheet(void*)"
end

module XML
  extend DL::Importer
  dlload "libxml2.so"
  extern "void *xmlParseFile(char*)"
  extern "void xmlFreeDoc(void*)"
end

cur = XSLT.xsltParseStylesheetFile("sample.xsl")
doc = XML.xmlParseFile("sample.xml")
res = XSLT.xsltApplyStylesheet(cur, doc, nil)

cur.free = XSLT["xsltFreeStylesheet"]
doc.free = XML["xmlFreeDoc"]
res.free = XML["xmlFreeDoc"]

XSLT.xsltSaveResultToFile(DL::CPtr[$stdout], res, cur)
```

That's (really) all folks!

Brian McCallister
brianm@apache.org
http://kasparov.skife.org/ac-us-04-ror.pdf
http://kasparov.skife.org/ac-us-04-ror.swf

Thank You:
Sun, Black Hat, Intel, Covalent, JCP, Azul Systems, CafeSoft, IBM, LogicBlaze, Rogue Wave, Thawte, Chariot Solutions, Byte.com, and C/C++ User's Journal