Rails 3 Ropes Course

Gregg Pollack
Nathaniel Bibler
Thomas Meeks
Jacob Swanner
Tyler Hunt
Mark Kendall
Caike Souza
<table>
<thead>
<tr>
<th>Workshop - Lab #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started &amp; Routes</td>
</tr>
<tr>
<td>Workshop - Lab #2</td>
</tr>
<tr>
<td>Bundler &amp; ActionController</td>
</tr>
<tr>
<td>Workshop - Lab #3</td>
</tr>
<tr>
<td>ActionMailer</td>
</tr>
<tr>
<td>Workshop - Lab #4</td>
</tr>
<tr>
<td>ActiveRelation &amp; ActiveModel</td>
</tr>
<tr>
<td>Workshop - Lab #5</td>
</tr>
<tr>
<td>XSS &amp; UJS</td>
</tr>
</tbody>
</table>
$ rails
Usage:
  rails APP_PATH [options]

Options:
- `-r, [--ruby=PATH]` # Path to the Ruby binary of your choice
- `-d, [--database=DATABASE]` # Preconfigure for selected database
- `-m, [--template=TEMPLATE]` # Path to an application template
  [---dev]
  [---edge]
- `[--skip-gemfile]` # Setup the application with
- `-O, [--skip-activerecord]` # Gemfile pointing to your Rails checkout
- `-T, [--skip-testunit]` # Setup the application with
- `-J, [--skip-prototype]` # Gemfile pointing to Rails repository
- `-G, [--skip-git]` # Don't create a Gemfile
# Skip ActiveRecord files
# Skip TestUnit files
# Skip Prototype files
# Skip Git ignores and keeps

.bundle
db/*.sqlite3
log/*.log
tmp/**/*
$ rails test_app
  create
  create  README
  create  .gitignore
  ...

$ cd test_app/
$ rails
Usage: rails COMMAND [ARGS]

The most common rails commands are:
generate  Generate new code (short-cut alias: "g")
console   Start the Rails console (short-cut alias: "c")
server    Start the Rails server (short-cut alias: "S")
dbconsole Start a console for the database specified in config/database.yml
           (short-cut alias: "db")

In addition to those, there are:
application Generate the Rails application code
destroy    Undo code generated with "generate"
benchmarker See how fast a piece of code runs
profiler    Get profile information from a piece of code
plugin      Install a plugin
runner      Run a piece of code in the application environment

All commands can be run with -h for more information.
script/generate

script/console

script/server

script/dbconsole

alias r='rails'
Rails::Initializer.run do |config|

    config.load_paths += %W( #{RAILS_ROOT}/extras )

    config.gem "bj"
    config.gem "sqlite3-ruby", :lib => "sqlite3"
    config.gem "aws-s3", :lib => "aws/s3"

    config.plugins = [ :exception_notification ]

    config.time_zone = 'UTC'

end
module TestApp
  class Application < Rails::Application

    config.load_paths += %W( #{RAILS_ROOT}/extras )
    config.plugins = [ :exception_notification ]
    config.time_zone = 'UTC'

  end
end

It's a Rack Application
New Router API
TestApp::Application.routes.draw do |
  map |
end

cfg/routes.rb

New Routing API

Rails 3
config/routes.rb

TestApp::Application.routes.draw do |map|
  map.resources :posts
end

Old routing syntax works
map.resources :posts do |post|
  post.resources :comments
end

resources :posts do
  resources :comments
end
post.resources :comments,  
    :member => { :preview => :post },  
    :collection => { :archived => :get }

resources :comments do  
    member do  
        post :preview
    end

    collection do  
        get :archived
    end
end

resources :comments do  
    post :preview, :on => :member  
    get :archived, :on => :collection
end
map.connect 'login', :controller => 'session', :action => 'new'

match 'login' => 'session#new'

Named Route login_path

Rails 2

Rails 3

map.login 'login', :controller => 'session', :action => 'new'

match 'login' => 'session#new', :as => :login

Rails 3

Rails 2
Rails 2

map.root :controller => 'users', :action => 'index'

Rails 3

root :to => 'users#index'

Legacy Route

Rails 2

map.connect ':controller/:action/:id'
map.connect ':controller/:action/:id.:format'

Rails 3

match ':controller(/:action(/:id(/:format))))'
(commented out by default)
Optional Parameters

Rails 2

map.connect '/articles/:year/:month/:day', :controller => 'posts', :action => 'index'

Rails 3

match '/articles/:year/:month/:day' => "posts#index"

Rails 2

map.connect '/articles/:year/:month/:day', :controller => 'posts', :action => 'index'
map.connect '/articles/:year/:month', :controller => 'posts', :action => 'index'
map.connect '/articles/:year', :controller => 'posts', :action => 'index'

Rails 3

match '/articles(:year(/:month(/:day))))' => "posts#index"
Specifying the method

**Rails 2**

```ruby
done.connect '/articles/:year', :controller => 'posts', :action => 'index', :conditions => {:method => :get}
```

**Rails 3**

```ruby
match '/articles/:year' => "posts#index", :via => :get
```

```ruby
get '/articles/:year' => "posts#index"
```
Rails 3

match 'signin', :to => redirect("/login")

match 'users/:name', :to => redirect { |params| "#/#{params[:name]}" }

match 'google' => redirect('http://www.google.com/')}
Constraints

Rails 2

map.connect '/:year', :controller => 'posts', :action => 'index',
  :requirements => { :year => \d{4}/ }

Rails 3

match '/:year' => "posts#index", :constraints => { :year => \d{4}/ }

:constraints => { :user_agent => /iphone/ }

:constraints => { :ip => /192\.168\.1\./d{1,3}/ }

constraints(:host => /localhost/) do
  resources :posts
end

constraints IpRestrictor do
  get 'admin/accounts' => "queenbee#accounts"
end
Rails 3

get 'hello' => proc { |env| [200, {}, "Hello Rack"] }

get 'rack_endpoint' => PostsController.action(:index)

get 'rack_app' => CustomRackApp
Rails Routing from the Outside In
This guide covers the user-facing features of Rails routing. By referring to this guide, you will be able to:

- Understand the purpose of routing
- Decipher the code in routes.rb
- Construct your own routes, using either the classic hash style or the now-preferred RESTful style
- Identify how a route will map to a controller and action

1 The Dual Purpose of Routing
Rails routing is a two-way piece of machinery—rather as if you could turn trees into paper, and then turn paper back into trees. Specifically, it both routes incoming HTTP requests to the code in your application's controllers, and helps you generate URLs without having to hard-code them as strings.

1.1 Connecting URLs to Code
When your Rails application receives an incoming HTTP request, say

```
GET /patients/17
```

the routing engine within Rails is the piece of code that dispatches the request to the appropriate spot in your application. In this case, the application would most likely end up running the show action within the patients controller, displaying the details of the patient whose id is 17.

For more information

http://guides.rails.info/routing.html
Tutorial - Lab #1

Getting Started & Routes

Follow the directions in the README
# Rails 3 Ropes Course

<table>
<thead>
<tr>
<th>Getting Started &amp; Routes</th>
<th>Workshop - Lab #1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bundler &amp; ActionController</strong></td>
<td>Workshop - Lab #2</td>
</tr>
<tr>
<td><strong>ActionMailer</strong></td>
<td>Workshop - Lab #3</td>
</tr>
<tr>
<td><strong>ActiveRelation &amp; ActiveModel</strong></td>
<td>Workshop - Lab #4</td>
</tr>
<tr>
<td><strong>XSS &amp; UJS</strong></td>
<td>Workshop - Lab #5</td>
</tr>
</tbody>
</table>
Dependency Management
Typical Rails deployment

Specify your gems inside environment.rb

```ruby
config.gem "haml"
config.gem "chronic", :version => '0.2.3'
```

`$ rake gems:install`

fetch, download, and install/compile these gems into your system RubyGems directory

`$ rake gems:unpack:dependencies`

unpacks the gems into your application into your vendor/gems directory
Issues

- It's bound into Rails
- Not great dependency resolution
- Conflicts occur at runtime
Gem::LoadError: can't activate activesupport (= 2.3.2, runtime), already activated activesupport-2.3.3
Bundler Commands

Specify your gems inside Gemfile

```ruby
gem "haml"
gem "chronic", '0.2.3'
```

$ bundle

fetch, download, and install/compile these gems

$ bundle package

Moves gem source into /vendor/cache
With Bundler

ActiveMerchant 1.4.2
  activewire >= 2.3.2

Rails 2.3.2
  activewire = 2.3.2

System Gems
  Rails 2.3.3
  thor 0.13.1

Rails App
  Rails 2.3.2
  ActiveMerchant 1.4.2

Happy!
Dependency Resolution

Depth First Search

dependencies

paperclip

shoulda

sqlite3

aws-s3

activerecord

CONFLICT

mocha

rake-compiler
source "http://rubygems.org"

gem "hpricot", "0.6"
gem "sqlite3-ruby", :require => "sqlite3"
gem "rails", :git => "git://github.com/rails/rails.git"
gem "rails", :path => "~/Sites/rails"

git "git://github.com/rails/rails.git" do
  gem "railties"
  gem "active_model"
end

group :test do
  gem "webrat"
end

$bundle
Will ensure all gems are installed, including webrat (but it’s only included in test mode).

$bundle --without test
Will install everything except webrat.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$ bundle check</code></td>
<td>Check to see if all the dependencies are available for the Gemfile</td>
</tr>
<tr>
<td><code>$ bundle show</code></td>
<td>Shows all libraries which are included by the Gemfile and their dependencies</td>
</tr>
<tr>
<td><code>$ bundle show &lt;gem name&gt;</code></td>
<td>Shows where the gem is located in our filesystem</td>
</tr>
<tr>
<td><code>$ bundle open &lt;gem name&gt;</code></td>
<td>Will open the gem source in our default editor</td>
</tr>
</tbody>
</table>
When I download a new application

I locally run

$ bundle

When I deploy my application

I run on the server

$ bundle

When my application is going live

Gemfile.lock  Check into Source Control

Specifies exact gem versions
my application is running

on the server

$ bundle

Installs gems listed in Gemfile.lock rather than Gemfile
Gemfile vs Gemfile.lock

**Gemfile**

```ruby
source 'http://rubygems.org'
gem 'capistrano'
gem 'nokogiri', '>=1.4.0'
gem 'sqlite3-ruby', :require => 'sqlite3'
```

**Gemfile.lock**

```
remote: http://rubygems.org/
specs:
  capistrano (2.5.18)
  highline
  net-scp (>= 1.0.0)
  net-sftp (>= 2.0.0)
  net-ssh (>= 2.0.14)
  net-ssh-gateway (>= 1.0.0)
  nokogiri (1.4.0)
  sqlite3-ruby (1.2.5)
```

Recreates the Gemfile.lock and runs "bundle" to install new dependencies.

- `nokogiri 1.4.1 released!`
- `$ bundle`
- `$ bundle update nokogiri`
- `$ bundle update`
But I don’t want to rely on external servers for deployment.

Run locally:

$ bundle package

Copies all gems to your /vendor/cache directory

On the server:

$ bundle

Will install gems from the /vendor/cache directory
Bundler manages an application's dependencies through its entire life across many machines systematically and repeatably.

For more information

http://gembundler.com
includes

AbstractController Namespace
  - Assigns
  - Callbacks
  - Collector
  - Helpers
  - Layouts
  - Logger
  - Rendering
  - Translation
  - ViewPaths

ActionController Namespace
  - Cookies
  - Exceptions
  - Flash
  - Helpers
  - Redirecting
  - Rendering
  - Responder
  - UrlFor
  - ..... (lots more)

abstract

AbstractController::Base
abstract

ActionController::Base
abstract

ActionController::Metal
class UsersController < ApplicationController

def index
  @users = User.all

  respond_to do |format|
    format.html
    format.xml { render :xml => @users.to_xml }
  end
end

def show
  @user = User.find(params[:id])

  respond_to do |format|
    format.html # show.html.erb
    format.xml { render :xml => @user }
  end
end
...

class UsersController < ApplicationController

  respond_to :html, :xml, :json

  def index
    @users = User.all
    respond_with(@users)
  end

  def show
    @user = User.find(params[:id])
    respond_with(@user)
  end

  ...

Tutorial - Lab #2

Bundler & ActionController
# Rails 3 Ropes Course

<table>
<thead>
<tr>
<th>Topic</th>
<th>Workshop - Lab #1</th>
<th>Workshop - Lab #2</th>
<th>Workshop - Lab #3</th>
<th>Workshop - Lab #4</th>
<th>Workshop - Lab #5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started &amp; Routes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bundler &amp; ActionController</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ActionMailer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ActiveRelation &amp; ActiveModel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XSS &amp; UJS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ActionMailer

Mikel Lindsaar
New ActionMailer Syntax

**Rails 2**

```
$script/generate mailer UserMailer welcome forgot_password
create  app/models/user_mailer.rb
```

**Rails 3**

```
$rails g mailer UserMailer welcome forgot_password
create  app/mailers/user_mailer.rb
```
New ActionMailer Syntax

Rails 2

```ruby
def welcome(user, subdomain)
    subject 'Welcome to TestApp'
    recipients user.email
    from 'admin@testapp.com'

    body :user => user, :subdomain => subdomain
end
```

UserMailer.deliver_welcome(user, subdomain)

Rails 3

```ruby
def welcome(user, subdomain)
    @user = user
    @subdomain = subdomain

    mail(:from => "admin@testapp.com",
         :to => user.email,
         :subject => "Welcome to TestApp")
end
```

UserMailer.welcome(user, subdomain).deliver
class UserMailer < ActionMailer::Base

    def welcome(user, subdomain)
        @user = user
        @subdomain = subdomain

        attachments['test.pdf'] = File.read("#{Rails.root}/public/test.pdf")

        mail(:to => @user.email, :subject => "Welcome to TestApp") do |
            format.html { render 'other_html_welcome' }
            format.text { render 'other_text_welcome' }
        end
    end
end

Defaults

welcome.text.erb
welcome.html.erb
Tutorial - Step #3

ActionMailer
# Rails 3 Ropes Course

<table>
<thead>
<tr>
<th>Getting Started &amp; Routes</th>
<th>Workshop - Lab #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundler &amp; ActionController</td>
<td>Workshop - Lab #2</td>
</tr>
<tr>
<td>ActionMailer</td>
<td>Workshop - Lab #3</td>
</tr>
<tr>
<td>ActiveRelation &amp; ActiveModel</td>
<td>Workshop - Lab #4</td>
</tr>
<tr>
<td>XSS &amp; UJS</td>
<td>Workshop - Lab #5</td>
</tr>
</tbody>
</table>
replaces the internal ad-hoc query generation with query generation based on relational algebra.
Rails 2

@posts = Post.find(:all, :conditions => {:published => true})

immediately queries the db
returns an Array of Posts

Rails 3

@posts = Post.where(:published => true)

doesn’t query the db
returns an ActiveRecord::Relation
@posts = Post.where(:published => true)

if params[:order]
    @posts = @posts.order(params[:order])
end

@posts.each do |p|
    ...
end

Lazy Loading
@posts = Post.where(:published => true)

if params[:order]
  @posts = @posts.order(params[:order])
end

@posts = Post.where(:published => true)
@posts = @posts.order(params[:order])

@posts = Post.where(:published => true).order(params[:order])
@posts = Post.where(:published => true).order(params[:order])

posts = Post.order(params[:order])

@published = posts.where(:published => true)
@unpublished = posts.where(:published => false)

This is obviously a bad example

@published = Post.published
@unpublished = Post.unpublished
```ruby
class Post < ActiveRecord::Base
  default_scope :order => 'title'
  named_scope :published, :conditions => {:published => true}
  named_scope :unpublished, :conditions => {:published => false}
end
```

Rails 3

```ruby
class Post < ActiveRecord::Base
  default_scope order('title')
  scope :published, where(:published => true)
  scope :unpublished, where(:published => false)
end
```
New Finder Methods

where(:conditions)
having(:conditions)
select
group
order
limit
offset
joins
includes(:include)
lock
readonly
from
Post.find(:all, :conditions => {:author => "Joe"}, :includes => :comments, :order => "title", :limit => 10)

Post.where(:author => "Joe").include(:comments).order(:title).limit(10).all
ActiveModel

ActiveModel API

ActiveModel Helper Modules
Old ActiveRecord Stack

- Associations
- Calculations
- Migrations
- Schema
- Transactions
- Callbacks
- Dirty
- Serialization
- Named Scope
- Fixtures

ActiveModel

- Attribute Methods
- Callbacks
- Dirty
- Errors
- Naming
- Observing
- Serialization
- Translation
- Validations
ActiveModel

**Attribute Methods**

**Callbacks**

**Dirty**

**Errors**

**Naming**

**Observing**

**Serialization**

**Translation**

**Validations**

```ruby
before_create :authenticate
before_save :send_email
around_create :log

person.changed?
person.name_changed?
person.name_was
person.name_change

person = Person.new
person.serializable_hash
person.to_json
person.to_xml

validates_presence_of :email
validates_length_of :name, :within => 3..20
validates_inclusion_of :salary, :in => 50000..200000
```
ActiveModel

Attribute Methods
Callbacks
Dirty
Errors
Naming
Observing
Serialization
Translation
Validations
class Applicant
  include ActiveModel::Validations
  validates_presence_of :name, :email

  attr_accessor :name, :email
end

>> a = Applicant.new
   => #<Applicant:0x000001021b7198>
>> a.name = "Gregg"
   => "Gregg"
>> a.valid?
   => false
>> a.errors
   => {:email=>["can't be blank"]}
>> a.email = "Gregg@EnvyLabs.com"
   => "Gregg@EnvyLabs.com"
>> a.valid?
   => true
class Applicant
  include ActiveModel::Serializers::JSON

  attr_accessor :name

  def attributes
    @attributes ||= {:name => 'nil'}
  end
end

>> a = Applicant.new
   => #<Applicant:0x00000102186d68>
>> a.name = "Gregg"
   => "Gregg"
>> a.to_json
   => "{"name":"Gregg"}"

Serialization without db
Rails 3 Ropes Course

Getting Started & Routes
  Workshop - Lab #1

Bundler & ActionController
  Workshop - Lab #2

ActionMailer
  Workshop - Lab #3

ActiveRelation & ActiveModel
  Workshop - Lab #4

XSS & UJS
  Workshop - Lab #5
Cross Site Scripting
Cross-Site Scripting (XSS)

New post

Title

Body

<%=
@post.body %>

Rails 2

(unsafe)

<%= h @post.body %>

Rails 2

(safe)

<%= raw @post.body %>

Rails 3

(unsafe)

<%= @post.body %>

Rails 3

(safe)
Returns a String which is assumed to be safe and will not be escaped again.
Cross-Site Scripting (XSS)

Rails 2

<%= link_to "<span class='cart'>Cart</span>"#, cart_path %>

Rails 3

<%= link_to raw("<span class='cart'>Cart</span>"), cart_path %>

Rails 3

<%= link_to raw("<span class='cart'>#{@user_input}</span>"), cart_path %>

Vulnerable to attack!!

No escaping done on user input
Cross-Site Scripting (XSS)

<%= link_to raw("<span class='cart'>#{@user_input}</span>"), cart_path %>

<%= link_to raw("<span class='cart'>#{h @user_input}</span>"), cart_path %>

Safe!
UJS Unobtrusive Javascript
HTML 5 custom data attributes

Custom data attributes are intended to store custom data private to the page or application, for which there are no more appropriate attributes or elements.
Adopting Unobtrusive Javascript

Rails 2

<%= link_to_remote 'Show', :url => post %>

<a href="#" onclick="new Ajax.Request('/posts/1', {asynchronous:true, evalScripts:true, parameters:'authenticity_token=' + encodeURIComponent('9sk..44d')}); return false;">Show</a>

Rails 3

<%= link_to 'Show', post, :remote => true %>

<a href="/posts/1" data-remote="true">Show</a>
Adopting Unobtrusive Javascript

Rails 2

```ruby
<% remote_form_for(@post) do |f| %>
<%= form_for(@post, :remote => true) do |f| %>
<form action="/posts" class="new_post" id="new_post" method="post"
onsubmit="new Ajax.Request('/posts', {asynchronous:true, evalScripts:true, parameters:Form.serialize(this)}); return false;">
```

Rails 3

```ruby
<%= form_for(@post, :remote => true) do |f| %>
<form action="/posts" class="new_post" data-remote="true" id="new_post" method="post">
```
Adopting Unobtrusive Javascript

Rails 2

```<%= link_to 'Destroy', post, :method => :delete %>
```

Rails 3

```<a href="/posts/1" data-method="delete" rel="nofollow">Destroy</a>
```

```<a href="/posts/1" onclick="var f = document.createElement('form'); f.style.display = 'none'; this.parentNode.appendChild(f); f.method = 'POST'; f.action = this.href;var m = document.createElement('input'); m.setAttribute('type', 'hidden'); m.setAttribute('name', '_method'); m.setAttribute('value', 'delete'); f.appendChild(m);var s = document.createElement('input'); s.setAttribute('type', 'hidden');s.setAttribute('name', 'authenticity_token'); s.setAttribute('value', '9skdJ0k+l9/q3PWToz6MtfyiB2gcyhnKubeGV6WFL44='); f.appendChild(s);f.submit();return false;">Destroy</a>
```

```<a href="/posts/1" data-method="delete" rel="nofollow">Destroy</a>
```
Adopting Unobtrusive Javascript

Rails 2   Rails 3

\ `<%= link_to 'Destroy', post, :confirm => 'Are you sure?', :method => :delete %>`

\ `<a href="/posts/1" onclick="if (confirm('Are you sure?')) { var f = document.createElement('form'); f.style.display = 'none'; this.parentNode.appendChild(f); f.method = 'POST'; f.action = this.href; var m = document.createElement('input'); m.setAttribute('type', 'hidden'); m.setAttribute('name', '_method'); m.setAttribute('value', 'delete'); f.appendChild(m); var s = document.createElement('input'); s.setAttribute('type', 'hidden'); s.setAttribute('name', 'authenticity_token'); s.setAttribute('value', '9skdJ0k+l9/q3PWToz6MfFyiB2gcynKubeGV6WFL44='); f.appendChild(s); f.submit(); } ;return false;">Destroy</a>`

\ `<a href="/posts/1" data-confirm="Are you sure?" data-method="delete" rel="nofollow">Destroy</a>`
Adopting Unobtrusive Javascript

HTML 5 custom data attributes

- data-remote
- data-method
- data-confirm
- data-disable-with
document.observe("dom:loaded", function() {

  $(document.body).observe("click", function(event) {

    var message = event.element().readAttribute('data-confirm');
    if (message) {
      // ... Do a confirm box
    }

    var element = event.findElement("a[data-remote]");
    if (element) {
      // ... Do the AJAX call
    }

    var element = event.findElement("a[data-method]");
    if (element) {
      // ... Create a form
    }
  });
});
jQuery in Rails?

http://github.com/rails/jquery-ujs

```javascript
$('a[data-confirm],input[data-confirm]').live('click', function () {
  // ... Do a confirm box
});

$('form[data-remote]').live('submit', function (e) {
  // ... Do an AJAX call
});
```
Deprecated Methods

- link_to_remote
- remote_form_for
- observe_field
- observe_form
- form_remote_tag
- button_to_remote
- submit_to_remote
- link_to_function
- periodically_call_remote

[prototype_legacy_helper](http://github.com/rails/prototype_legacy_helper)
Cross-site Request Forgery

Hacker site

<form target="http://yoursite.com">

Your site

Rails 2

```ruby
class ApplicationController < ActionController::Base
  protect_from_forgery
end
```

Rails 3

Rails 2

```
<% form_for(@post) do |f| %>
  <input name="authenticity_token" type="hidden" value="vg..4=" />
<% end %>
```

Rails 3

```
<%= csrf_meta_tag %>
```

(in your layout)

Rails.js unobtrusively adds the token
Tutorial - Lab #5

XSS & UJS
<table>
<thead>
<tr>
<th>Workshop - Lab #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started &amp; Routes</td>
</tr>
<tr>
<td>Workshop - Lab #2</td>
</tr>
<tr>
<td>Bundler &amp; ActionController</td>
</tr>
<tr>
<td>Workshop - Lab #3</td>
</tr>
<tr>
<td>ActionMailer</td>
</tr>
<tr>
<td>Workshop - Lab #4</td>
</tr>
<tr>
<td>ActiveRelation &amp; ActiveModel</td>
</tr>
<tr>
<td>Workshop - Lab #5</td>
</tr>
<tr>
<td>XSS &amp; UJS</td>
</tr>
</tbody>
</table>
Internal APIs
(that we didn’t cover)

Generators
ActiveModel API
Custom Validations
ActionController Modularity
Railties
<table>
<thead>
<tr>
<th>name</th>
<th>author</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>rainbow of 80s toys</td>
<td>merwing little dear</td>
<td><a href="http://www.flickr.com/photos/merwing/2152164258/">http://www.flickr.com/photos/merwing/2152164258/</a></td>
</tr>
<tr>
<td>Notting Hill Gate</td>
<td>Eole</td>
<td><a href="http://www.flickr.com/photos/eole/942309733/">http://www.flickr.com/photos/eole/942309733/</a></td>
</tr>
<tr>
<td>Das Licht</td>
<td>Small</td>
<td><a href="http://www.flickr.com/photos/small/62713023/">http://www.flickr.com/photos/small/62713023/</a></td>
</tr>
<tr>
<td>Metro Genova</td>
<td>opti mystic</td>
<td><a href="http://www.flickr.com/photos/miillio/2503634282/">http://www.flickr.com/photos/miillio/2503634282/</a></td>
</tr>
<tr>
<td>Immobility Dilemna</td>
<td>gilderic</td>
<td><a href="http://www.flickr.com/photos/gilderic/3528157964/">http://www.flickr.com/photos/gilderic/3528157964/</a></td>
</tr>
<tr>
<td>train station</td>
<td>nolifebeforecoffee</td>
<td><a href="http://www.flickr.com/photos/nolifebeforecoffee/1803584805/">http://www.flickr.com/photos/nolifebeforecoffee/1803584805/</a></td>
</tr>
<tr>
<td>Mystical station</td>
<td>Jsome1</td>
<td><a href="http://www.flickr.com/photos/jsome1/2226394415/">http://www.flickr.com/photos/jsome1/2226394415/</a></td>
</tr>
<tr>
<td>Railswaystation</td>
<td>Pieter Musterd</td>
<td><a href="http://www.flickr.com/photos/piet_musterd/2233025691/">http://www.flickr.com/photos/piet_musterd/2233025691/</a></td>
</tr>
<tr>
<td>The Handover</td>
<td>MarkyBon</td>
<td><a href="http://www.flickr.com/photos/markybon/152769885/">http://www.flickr.com/photos/markybon/152769885/</a></td>
</tr>
<tr>
<td>EN57</td>
<td>magro_kr</td>
<td><a href="http://www.flickr.com/photos/iks_berto/1328682171/">http://www.flickr.com/photos/iks_berto/1328682171/</a></td>
</tr>
<tr>
<td>Adirondack Extreme</td>
<td>Mikey Roach</td>
<td><a href="http://www.flickr.com/photos/mikeroach/4576888456/">http://www.flickr.com/photos/mikeroach/4576888456/</a></td>
</tr>
<tr>
<td>IMG_1242</td>
<td>khoogheem</td>
<td><a href="http://www.flickr.com/photos/khoogheem/3534078991/">http://www.flickr.com/photos/khoogheem/3534078991/</a></td>
</tr>
</tbody>
</table>
Presentation by:

Gregg Pollack
Gregg@EnvyLabs.com

Nathaniel Bibler
Nate@EnvyLabs.com

Thomas Meeks
Thomas@EnvyLabs.com

Jacob Swanner
Jacob@EnvyLabs.com

Tyler Hunt
Tyler@EnvyLabs.com

Mark Kendall
Mark@EnvyLabs.com

Caike Souza
Caike@EnvyLabs.com

If you need help with a Rails 3 project, feel free to give us a call