Choose Your Battles and LetIt::REST

Hampton Catlin and Jeffrey Hardy

unspace

THE HAMPTONS

Obligatory Resume

- 50 years of Rails experience
- PHDs from MIT
- Founded the Dharma Initiative

Invented

Haml, Scribbish, Sass, ASCII, Ruby, Lisp,
The DaVinci Code, Unicorns, Rainbows,
Kittens, Cat Macros, Balloons, Prototype,
Rails, Babies, Altoids, SQL, Religion, and
Monkey Poop (not monkeys, though).

The Beauty of Resource-Based Controllers

```
class PostsController < ApplicationController

  def index
    @posts = Post.find(:all)

    respond_to do |format|
      format.html # index.rhtml
      format.xml  { render :xml => @posts.to_xml }
    end
  end

  def show
    @post = Post.find(params[:id])

    respond_to do |format|
      format.html # show.rhtml
      format.xml  { render :xml => @post.to_xml }
    end
  end

  def new
    @post = Post.new
  end

  def edit
    @post = Post.find(params[:id])
  end

  def create
    @post = Post.new(params[:post])

    respond_to do |format|
      if @posts.save
        flash[:notice] = 'Posts was successfully created.'
        format.html { redirect_to post_url(@post) }
        format.xml  { head :created, :location => post_url(@post) }
      else
        format.html { render :action => "new" }
        format.xml  { render :xml => @post.errors.to_xml }
      end
    end
  end

  def update
    @post = Post.find(params[:id])

    respond_to do |format|
      if @post.update_attributes(params[:post])
        flash[:notice] = 'Posts was successfully updated.'
        format.html { redirect_to post_url(@post) }
        format.xml  { head :ok }
      else
        format.html { render :action => "edit" }
        format.xml  { render :xml => @post.errors.to_xml }
      end
    end
  end

  def destroy
    @post = Post.find(params[:id])
    @post.destroy

    respond_to do |format|
      format.html { redirect_to posts_url }
      format.xml  { head :ok }
    end
  end
end
```

The Problem

The Goal

PRODUCTION

No Shit!

Exceptions are Manageable

Action Pattern

- load resource object(s)
- optional preloads
- perform modifying action (CUD)
- respond

```ruby
class PostsController < ApplicationController

  def index
    @posts = Post.find(:all)

    respond_to do |format|
      format.html # index.rhtml
      format.xml  { render :xml => @posts.to_xml }
    end
  end

  def show
    @post = Post.find(params[:id])

    respond_to do |format|
      format.html # show.rhtml
      format.xml  { render :xml => @post.to_xml }
    end
  end

  def new
    @post = Post.new
  end

  def edit
    @post = Post.find(params[:id])
  end

  def create
    @post = Post.new(params[:post])

    respond_to do |format|
      if @posts.save
        flash[:notice] = 'Posts was successfully created.'
        format.html { redirect_to post_url(@post) }
        format.xml  { head :created, :location => post_url(@post) }
      else
        format.html { render :action => "new" }
        format.xml  { render :xml => @post.errors.to_xml }
      end
    end
  end

  def update
    @post = Post.find(params[:id])

    respond_to do |format|
      if @post.update_attributes(params[:post])
        flash[:notice] = 'Posts was successfully updated.'
        format.html { redirect_to post_url(@post) }
        format.xml  { head :ok }
      else
        format.html { render :action => "edit" }
        format.xml  { render :xml => @post.errors.to_xml }
      end
    end
  end

  def destroy
    @post = Post.find(params[:id])
    @post.destroy

    respond_to do |format|
      format.html { redirect_to posts_url }
      format.xml  { head :ok }
    end
  end
end
```

```
class PostsController < ApplicationController

  make_resourceful do
    build :all
  end

end
```

```ruby
class CommentsController < ApplicationController

  make_resourceful do
    build :create, :destroy
  end

end
```

```
map.resources :posts, :has_many => :comments
```

```
class CommentsController < ApplicationController

  make_resourceful do
    build :create, :destroy
    belongs_to :post
  end

end
```

Association Proxy

(our saviour)

```
Post.find(params[:post_id]).comments.find(:all)
Post.find(params[:post_id]).comments.find(params[:id])
Post.find(params[:post_id]).comments.find(params[:id]).update_attributes(params[:comment])
Post.find(params[:post_id]).comments.find(params[:id]).destroy
Post.find(params[:post_id]).comments.build(params[:comment])
Post.find(params[:post_id]).comments.build(params[:comment]).save
```

notice something?

```
current_model
```

```
Comment.belongs_to :user
```

```
current_user
```

```
class CommentsController < ApplicationController

  make_resourceful do
    build :create, :destroy
    belongs_to :post
    associate_with :current_user
  end

end
```

More Helpers

```ruby
def current_param
  params[:id]
end

def current_object
  current_model.find(current_param)
end

def current_objects
  current_model.find(:all)
end

def load_object
  instance_variable_set("@#{model_name}", current_object)
end
```

Callbacks

```
class PostsController < ApplicationController

  make_resourceful do
    build :all

    before :show do
      @page_title = "Awesome"
    end
  end

end
```

```
class PostsController < ApplicationController

  make_resourceful do
    build :all

    before :show do
      @page_title = "Awesome"
    end

    response_for :show do |format|
      format.html { redirect_to current_object }
    end

    response_for :update do
      redirect_to root_url
    end
  end

end
```

No Blocking

Encourages Phat Models

Experimental: Publish

```
class PostsController < ApplicationController

  make_resourceful do
    build :all

    #...

    publish :formats    => [:xml, :yaml, :json],
            :only       =>  :show,
            :attributes => [:title, :body, { :comments => :body }]
  end

end
```

super alpha

(but we’re using it in production)

http://hamptoncatlin.com

hcatlin@gmail.com

unspace