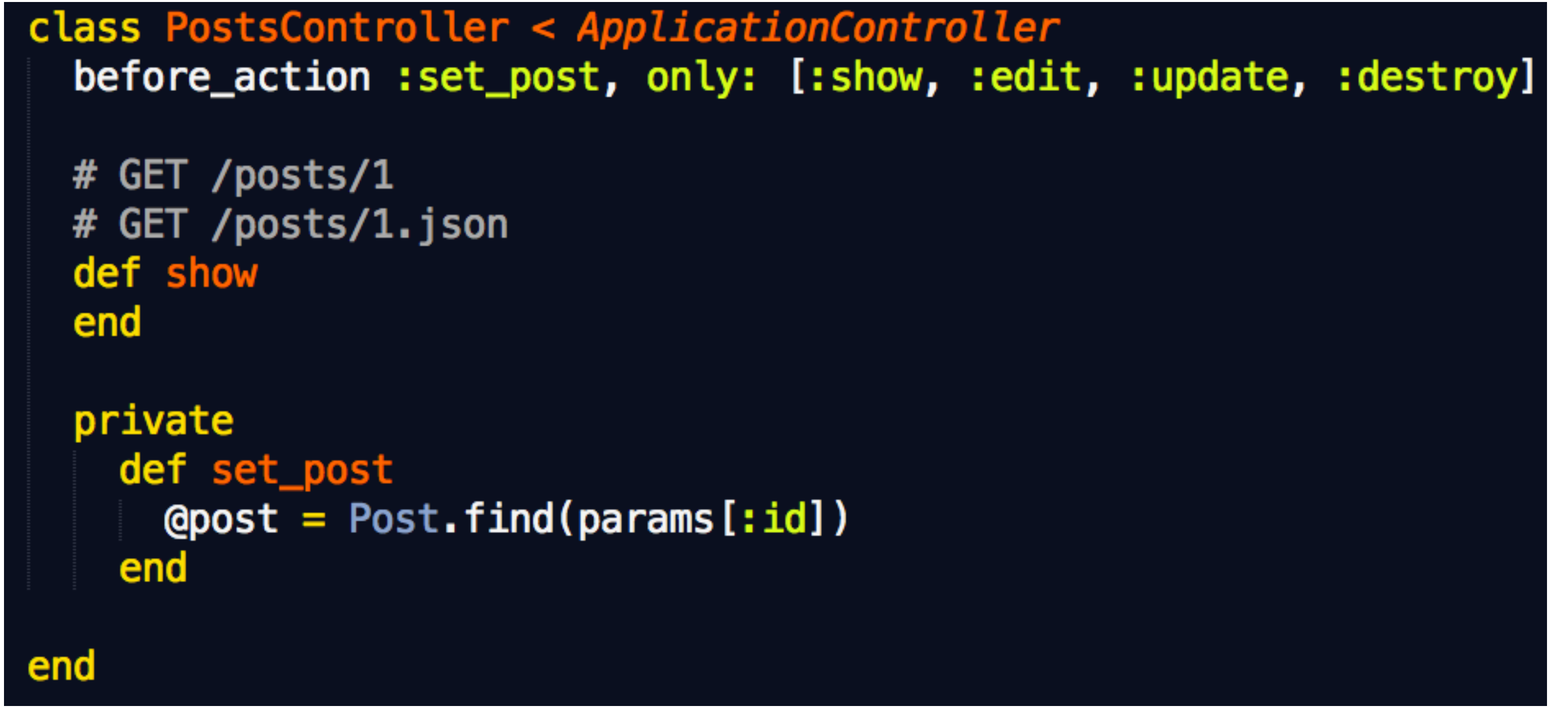
In this lecture, we will discuss…

- ✧ show RESTful action
- ✧ destroy RESTful action

Examining Seven Actions – show

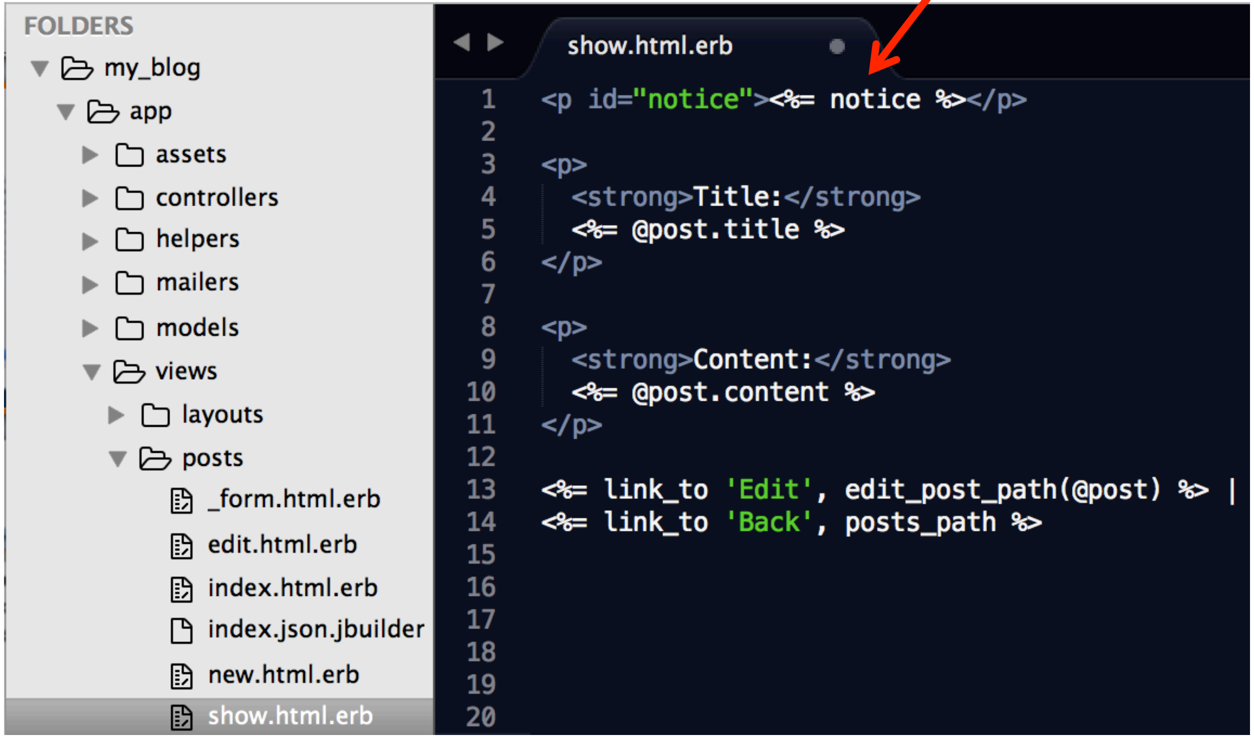
1. Retrieve specific post based on `id` parameter passed in (as part of URL)
2. (Implicit) Look for `show.html.erb` template to render response

Examining Seven Actions – show



```ruby
class PostsController < ApplicationController
  before_action :set_post, only: [:show, :edit, :update, :destroy]

  # GET /posts/1
  # GET /posts/1.json
  def show
  end

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

end
```

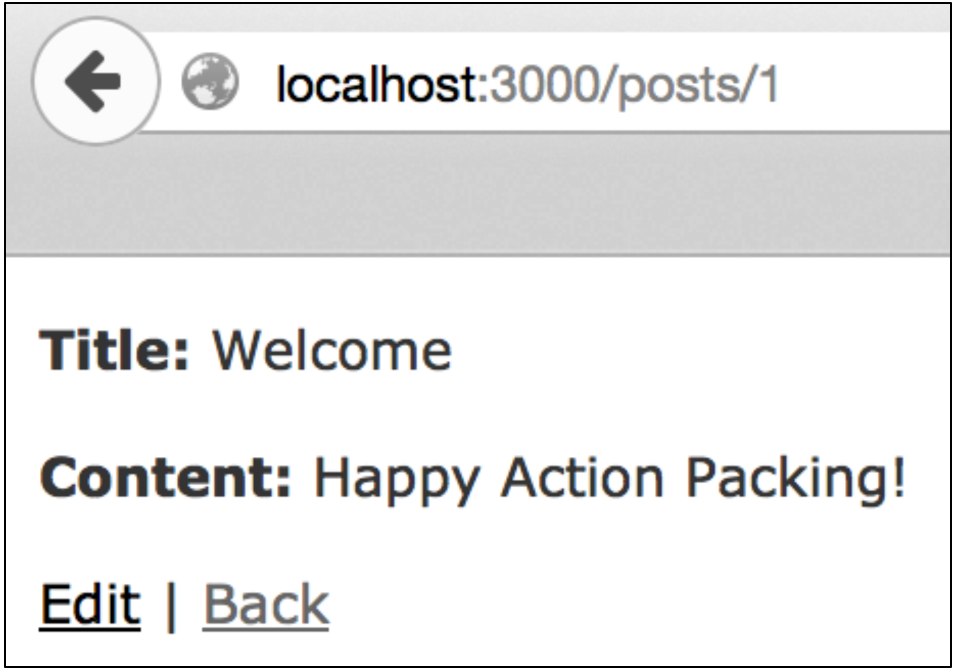
show.html.erb



```
<p id="notice"><%= notice %></p>

<p>
  <strong>Title:</strong>
  <%= @post.title %>
</p>

<p>
  <strong>Content:</strong>
  <%= @post.content %>
</p>

<%= link_to 'Edit', edit_post_path(@post) %> |
<%= link_to 'Back', posts_path %>
```

Explained later

show.html.erb



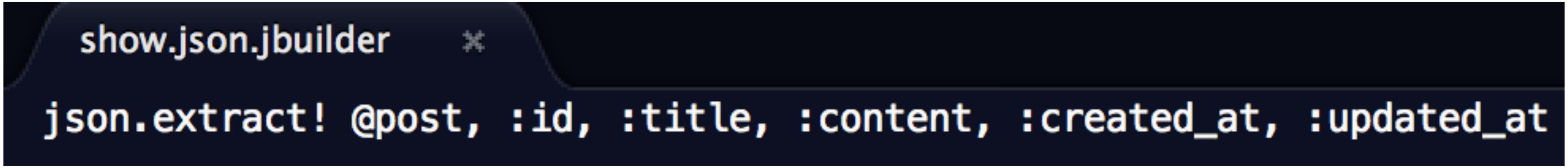
localhost:3000/posts/1

Title: Welcome

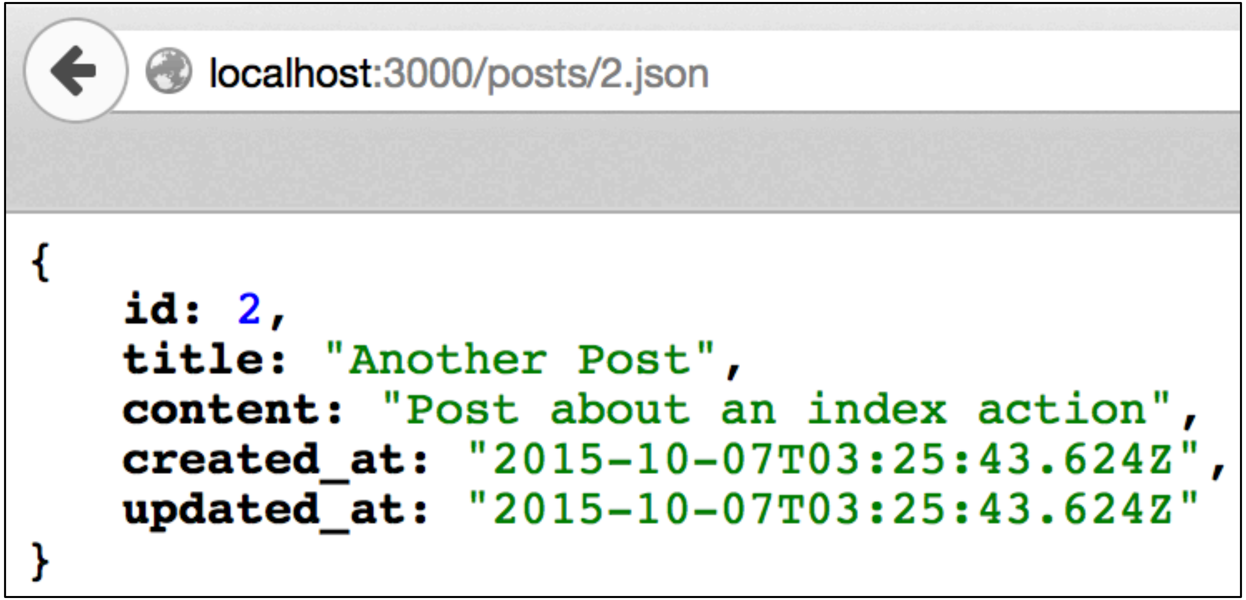
Content: Happy Action Packing!

Edit | Back

show.json.jbuilder



```ruby
json.extract! @post, :id, :title, :content, :created_at, :updated_at
```



```
{
    id: 2,
    title: "Another Post",
    content: "Post about an index action",
    created_at: "2015-10-07T03:25:43.624Z",
    updated_at: "2015-10-07T03:25:43.624Z"
}
```

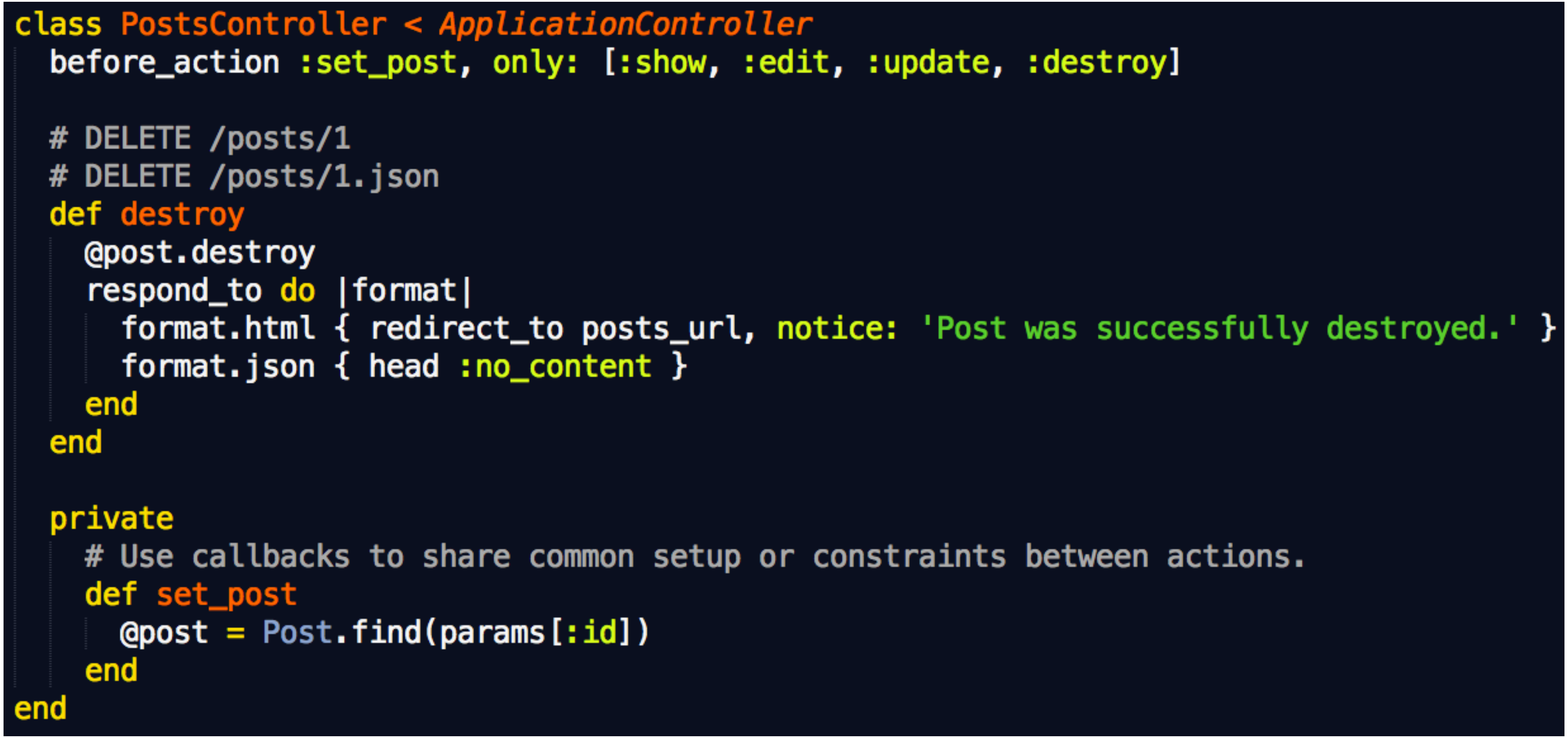
respond_to

- ✧ Rails helper that specifies how to respond to a request based on a request format
- ✧ Takes an optional block where the argument is the format (e.g. html, json, xml etc.)
- ✧ Block specifies how to handle each format:
  - `format.format_name` – matching template
  - `format.format_name { do_something_other_than_just_displaying_the_matching_template }`

redirect_to

- Instead of rendering a template – send a response to the browser: “Go here!”
- Usually takes a (full) URL as a parameter
- Could either be a regular URL (like http://google.com) or a named route
- If the parameter is an object – Rails will attempt to generate a URL for that object

Examining Seven Actions - destroy



```ruby
class PostsController < ApplicationController
  before_action :set_post, only: [:show, :edit, :update, :destroy]

  # DELETE /posts/1
  # DELETE /posts/1.json
  def destroy
    @post.destroy
    respond_to do |format|
      format.html { redirect_to posts_url, notice: 'Post was successfully destroyed.' }
      format.json { head :no_content }
    end
  end

  private
    # Use callbacks to share common setup or constraints between actions.
    def set_post
      @post = Post.find(params[:id])
    end
end
```

Why redirect?

- Even though redirect involves an extra step (roundtrip to the browser) – sometimes it just makes sense
- Obvious examples:
  - When you want the client to be able to bookmark a certain page or you don’t have a specific template to show (destroy action) and instead want the client to go to a generic page (index)

Summary

- ✧ show action involves retrieving a resource and showing it inside an HTML or JSON template
- ✧ destroy action destroys a resource and then redirects the browser to another page

What's Next?

- ✧ new and create actions