

In this lecture, we will discuss...

- ✧ edit action
- ✧ update action

Examining Seven Actions – edit

1. Retrieve a post object based on the `id` provided (as part of the URI)
2. (Implicit) Look for `edit.html.erb`

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

  # GET /posts/1/edit
  def edit
  end

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



edit.html.erb

Partial – explained later

```
edit.html.erb *  
<h1>Editing Post</h1>  
<%= render 'form' %>  
  
<%= link_to 'Show', @post %> |  
<%= link_to 'Back', posts_path %>
```

edit.html.erb



A screenshot of a web browser window. The address bar shows a back button, a globe icon, and the URL 'localhost:3000/posts/3/edit'. The page title is 'Editing Post'. Below the title, there is a 'Title' label followed by a text input field containing 'No Strong Parameters'. Below that is a 'Content' label followed by a text area containing 'Not going to work'. At the bottom left, there is an 'Update Post' button. At the bottom, there are two links: 'Show' and 'Back', separated by a vertical bar.

localhost:3000/posts/3/edit

Editing Post

Title

Content

Update Post

[Show](#) | [Back](#)

This looks remarkably similar to
new...

Examining Seven Actions – update

1. **Retrieve** an existing post using **id** parameter
2. **Update** post object with (**strong**) parameters that were passed from the **edit** form
3. Try to **(re)save** the object to the **database**
4. If successful, **redirect** to **show** template
5. If unsuccessful, **render edit** action (template) again



update action

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

  # PATCH/PUT /posts/1
  # PATCH/PUT /posts/1.json
  def update
    respond_to do |format|
      if @post.update(post_params)
        format.html { redirect_to @post, notice: 'Post was successfully updated.' }
        format.json { render :show, status: :ok, location: @post }
      else
        format.html { render :edit }
        format.json { render json: @post.errors, status: :unprocessable_entity }
      end
    end
  end

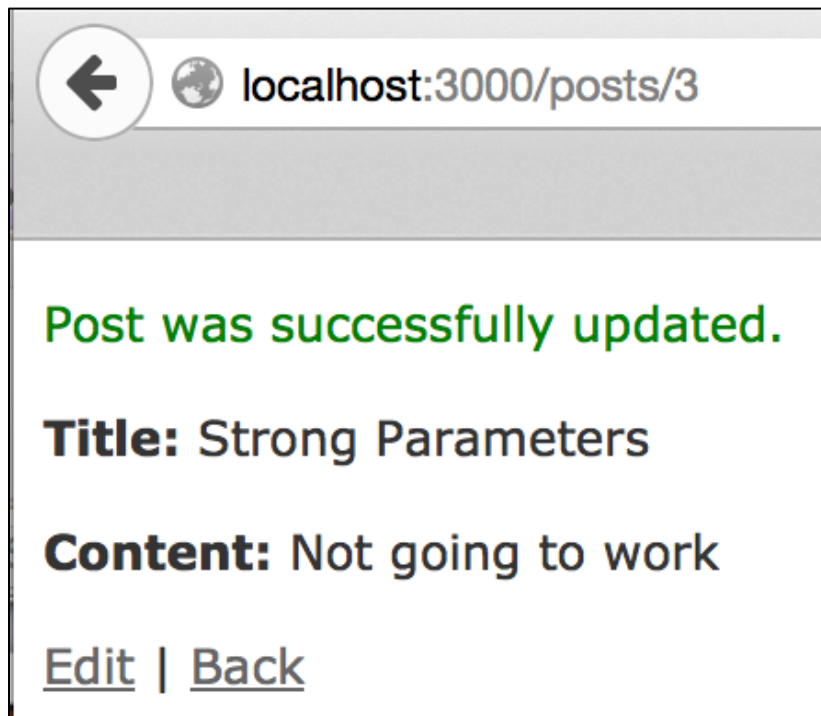
  private

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

  # Never trust parameters from the scary internet, only allow the white list through.
  def post_params
    params.require(:post).permit(:title, :content)
  end
end
```



update action



`show.html.erb` (with a notice)

Summary

- ✧ edit/update is very similar to new/create except there is an id of an existing resource that is being kept track of
- ✧ Strong parameters apply to updating a resource as well as creating one

What's Next?

- ✧ **Partials**

