

# In this lecture, we will discuss...

- ✧ Authorization



# Security Helpers

- ✧ Let's add `logged_in?` and `current_user` helpers to `ApplicationController` and make them available as helper methods to all controllers and **views** via `helper_method`
- ✧ Then, we can add **logic** to `application.html.erb` for logging out and information about the user who is logged in



# application\_controller.rb

FOLDERS

- ▼ i\_reviewed
  - ▼ app
    - ▶ assets
    - ▼ controllers
      - ▶ concerns
      - application\_controller.rb
      - books\_controller.rb
      - notes\_controller.rb
      - sessions\_controller.rb
    - ▶ helpers
    - ▶ mailers
    - ▶ models
    - ▼ views
      - ▶ books
      - ▼ layouts
        - application.html.erb
      - ▶ notes
      - ▶ sessions

application\_controller.rb \*

```
1 class ApplicationController < ActionController::Base
2   # Prevent CSRF attacks by raising an exception.
3   # For APIs, you may want to use :null_session instead.
4   protect_from_forgery with: :exception
5
6   before_action :ensure_login
7   helper_method :logged_in?, :current_user
8
9   protected
10    def ensure_login
11      # Always go to login page unless session contains
12      # reviewer_id
13      redirect_to login_path unless session[:reviewer_id]
14    end
15
16    def logged_in?
17      session[:reviewer_id] # nil is false
18    end
19
20    def current_user
21      @current_user ||= Reviewer.find(session[:reviewer_id])
22    end
23  end
```



# views/layouts/application.html.erb

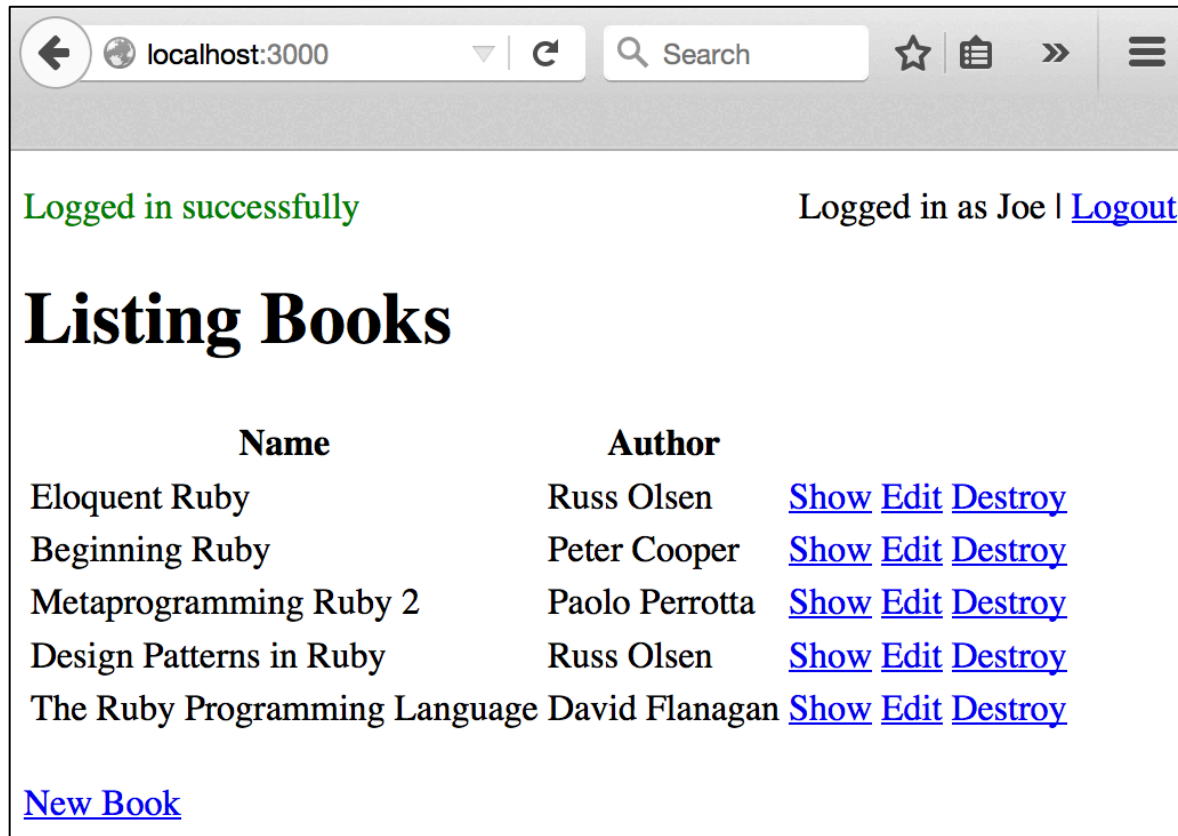
FOLDERS

- ▼ i\_reviewed
  - ▼ app
    - ▶ assets
    - ▶ controllers
    - ▶ helpers
    - ▶ mailers
    - ▶ models
  - ▼ views
    - ▶ books
    - ▼ layouts
      - application.html.erb
- ▶ notes
- ▶ sessions
- ▶ bin
- ▶ config
- ▼ db
  - ▶ migrate

application.html.erb \*

```
1 <!DOCTYPE html>
2 <html>
3 <head> ...
8 </head>
9 <body>
10
11 <%= if logged_in? %>
12   <div style='float: right;'>
13     Logged in as <%= current_user.name %> |
14     <%= link_to "Logout", logout_path, method: :delete %>
15   </div>
16 <%= end %>
17
18 <%= flash.each do |key, value| %>
19   <p id='<%= key %>'><%= value %></p>
20 <%= end %>
21
22 <%= yield %>
23
24 </body>
25 </html>
```

# views/layouts/application.html.erb



# Authorization

- ✧ We have implemented basic *Authentication*, but this still does nothing for our *Authorization*
- ✧ Anybody who logs into the system can edit anyone else's books and notes?!
- ✧ **SOLUTION:** We can go back to the `BooksController` and *scope things down* based on the `current_user` (instance of `Reviewer`)



# Authorization – index, new, create

```
books_controller.rb *  
1  class BooksController < ApplicationController  
2    before_action :set_book, only: [:show, :edit, :update, :destroy]  
3  
4    def index  
5      @books = current_user.books.all  
6    end  
7  
8    def new  
9      @book = current_user.books.new  
10   end  
11  
12  
13   def create  
14     @book = current_user.books.new(book_params)  
15   end  
16 end
```







# Authorization – the Other Actions




```
books_controller.rb
1 class BooksController < ApplicationController
2   before_action :set_book, only: [:show, :edit, :update, :destroy]
3
4   def show
5   end
6
7   def edit
8   end
9
10  def update ...
11  end
12
13  def destroy ...
14  end
15
16  private
17  # Use callbacks to share common setup or constraints between actions.
18  def set_book
19    @book = current_user.books.find(params[:id])
20  end
21
```





# Authorization

 localhost:3000  

 Search  









## Listing Books

Logged in as Joe | [Logout](#)

Name	Author	
Eloquent Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Metaprogramming Ruby 2	Paolo Perrotta	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[New Book](#)

# Authorization

  localhost:3000       

## Listing Books

Logged in as Jim | [Logout](#)

Name	Author	
Beginning Ruby	Peter Cooper	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Design Patterns in Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
The Ruby Programming Language	David Flanagan	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[New Book](#)

# Summary

- ✧ Can use the info stored in the `session` to store current user id
- ✧ Then, can look up resources associated with the current user

## What's Next?

- ✧ Pagination