

# In this lecture, we will discuss...

- ✧ Sessions controller and view
- ✧ Locking down our app

# views/sessions/new.html.erb

## FOLDERS

- ▼ i\_reviewed
  - ▼ app
    - ▶ assets
    - ▶ controllers
    - ▶ helpers
    - ▶ mailers
    - ▶ models
    - ▼ views
      - ▶ books
      - ▶ layouts
      - ▶ notes
      - ▼ sessions

new.html.erb

```
new.html.erb
1 |<h1>Login</h1>
2
3 |<%= form_for(:reviewer, url: sessions_path) do |f| %>
4
5 |   <div class="field"><%= f.label :name %> <br/> <%= f.text_field :name %></div>
6
7 |   <p/>
8
9 |   <div class="field"><%= f.label :password %> <br/> <%= f.password_field :password %></div>
10
11 |   <div class="actions"><%= f.submit "Login" %></div>
12 |<%= end %>
```



# Login Page

The screenshot displays a web browser at the address `localhost:3000/login`. The page features a login form with the following elements:

- Name:** A text input field containing the value "Jim".
- Password:** A password input field with masked characters ".....".
- Login:** A button to submit the form.

The browser's developer tools are open, showing the HTML structure. The `div.field` element is selected, which contains the form fields. The HTML code is as follows:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <h1>Login</h1>
    <form method="post" accept-charset="UTF-8" action="/sessions">
      <input type="hidden" value="✓" name="utf8">
      <input type="hidden" value="Nn7/7JIjyXpnEWnE56qGct3r4Kbm001eKvoQRqmCeA1c+FLC7yLXTPClh5hXaLot+4ERUioRo/QGX300Y17G1Q==" name="authenticity_token">
      <div class="field">
        <label for="reviewer_name">Name</label>
        <br>
        <input id="reviewer_name" type="text" name="reviewer[name]">
      </div>
      <p></p>
      <div class="field">
        <label for="reviewer_password">Password</label>
        <br>
        <input id="reviewer_password" type="password" name="reviewer[password]">
      </div>
      <div class="actions">
        <input type="submit" value="Login" name="commit">
      </div>
    </form>
  </body>
</html>
```

# Sessions Controller

## FOLDERS




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

sessions\_controller.rb \*

```
1 class SessionsController < ApplicationController
2   def new
3     # Login Page - new.html.erb
4   end
5
6   def create
7     reviewer = Reviewer.find_by(name: params[:reviewer][:name])
8     password = params[:reviewer][:password]
9
10    if reviewer && reviewer.authenticate(password)
11      session[:reviewer_id] = reviewer.id
12      redirect_to root_path, notice: "Logged in successfully"
13    else
14      redirect_to login_path, alert: "Invalid username/password combination"
15    end
16  end
17
18  def destroy
19    reset_session # wipe out session and everything in it
20    redirect_to login_path, notice: "You have been logged out"
21  end
22 end
```



# Logged In

   localhost:3000

Logged in successfully

## Listing Books

Name	Author	
Eloquent Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Beginning Ruby	Peter Cooper	<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>
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](#)

# Cookie in the Browser

Cookies

Search: i\_re

The following cookies match your search:

Site	Cookie Name
localhost	_i_reviewed_session

Name: \_i\_reviewed\_session

Content: dnNTTINmdU9BOXITaWhqTkNkck1IdERQQkpMbFZJbm5ScFlxRklzMzZ1MzZLZEJZM0hjZE05VjdUd0pla0RncIEvc3hqVXBESi8v

Host: localhost

Path: /

Send For: Any type of connection

Expires: At end of session

Remove Selected

Remove All

# Locking Down The App

- ✧ We can have a `before_action` in the `ApplicationController` (from which all the other controllers **inherit**) that will **make you login** if you are not yet logged in
- ✧ But if everything is **blocked off** – how will we get to the login page? Hmm...
- ✧ Controllers can **override** `before_action` with `skip_before_action`



# 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

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
8   protected
9     def ensure_login
10      # Always go to login page unless session contains
11      # reviewer_id
12      redirect_to login_path unless session[:reviewer_id]
13    end
14 end
```



# sessions\_controller.rb

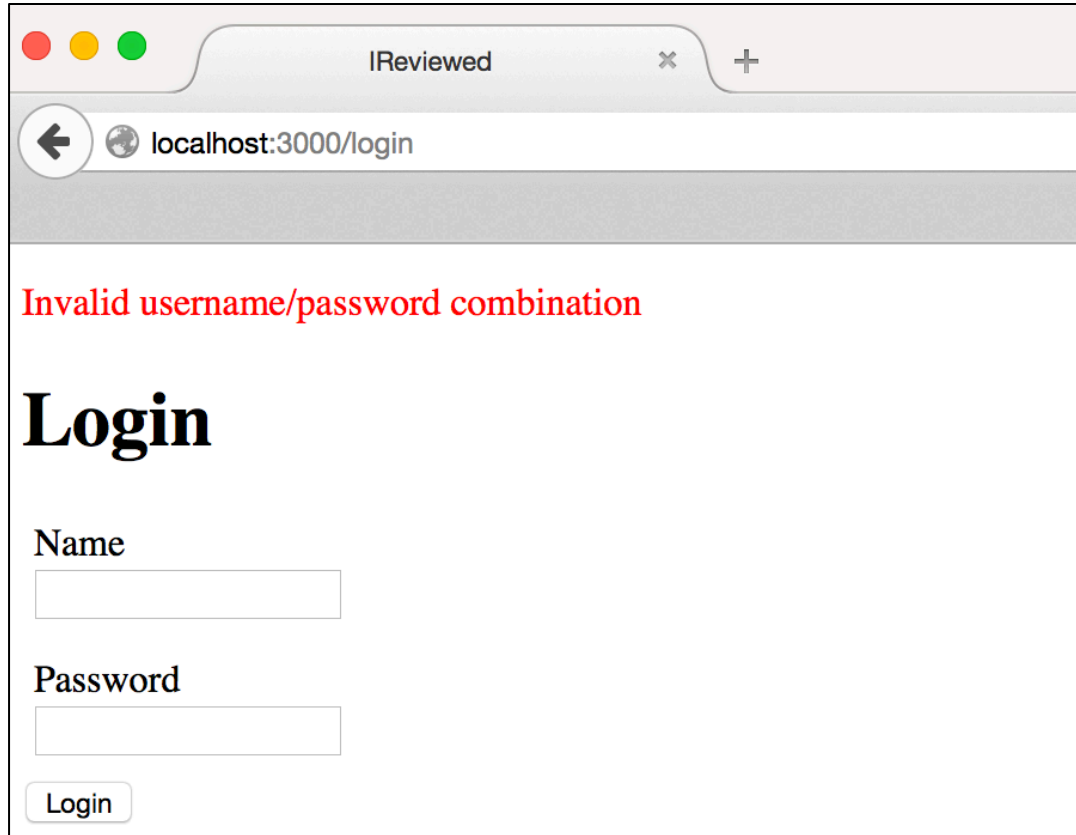
## FOLDERS

- ▼ i\_reviewed
  - ▼ app
    - ▶ assets
    - ▼ controllers
      - ▶ concerns
      - application\_controller.rb
      - books\_controller.rb
      - notes\_controller.rb
      - sessions\_controller.rb
    - ▶ helpers

```
sessions_controller.rb *  
1  class SessionsController < ApplicationController  
2    skip_before_action :ensure_login, only: [:new, :create]  
3    def new  
4      # Login Page - new.html.erb  
5    end  
6  
7    def create ...  
17  end  
18  
19  def destroy ...  
22  end  
23  end
```



# Unsuccessful Login



A screenshot of a web browser window. The title bar shows a single tab labeled "IReviewed". The address bar displays "localhost:3000/login". The main content area features a red error message, a large "Login" heading, and two input fields for "Name" and "Password". A "Login" button is positioned at the bottom of the form.

Invalid username/password combination

## Login

Name

Password

Login

# Summary

- ✧ Login page corresponds to `new` action `SessionsController`, but uses attributes from `Reviewer`
- ✧ Lock down the app by specifying a `before_action` in `ApplicationController`

## What's Next?

- ✧ Authorization

