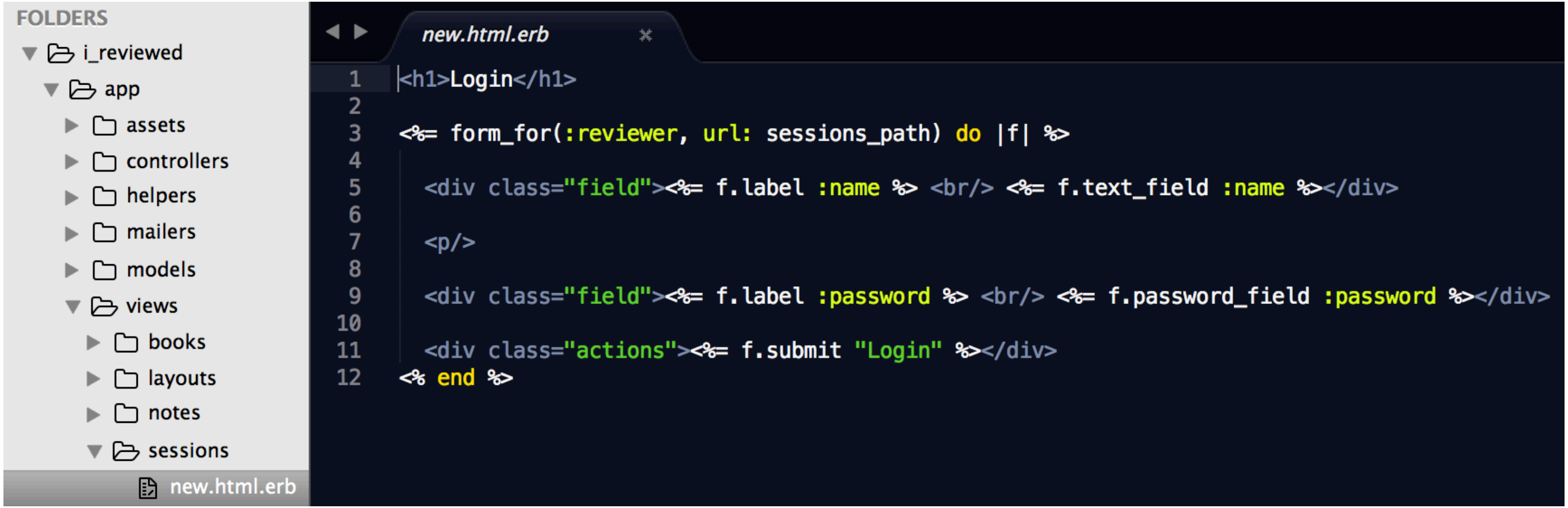
In this lecture, we will discuss…

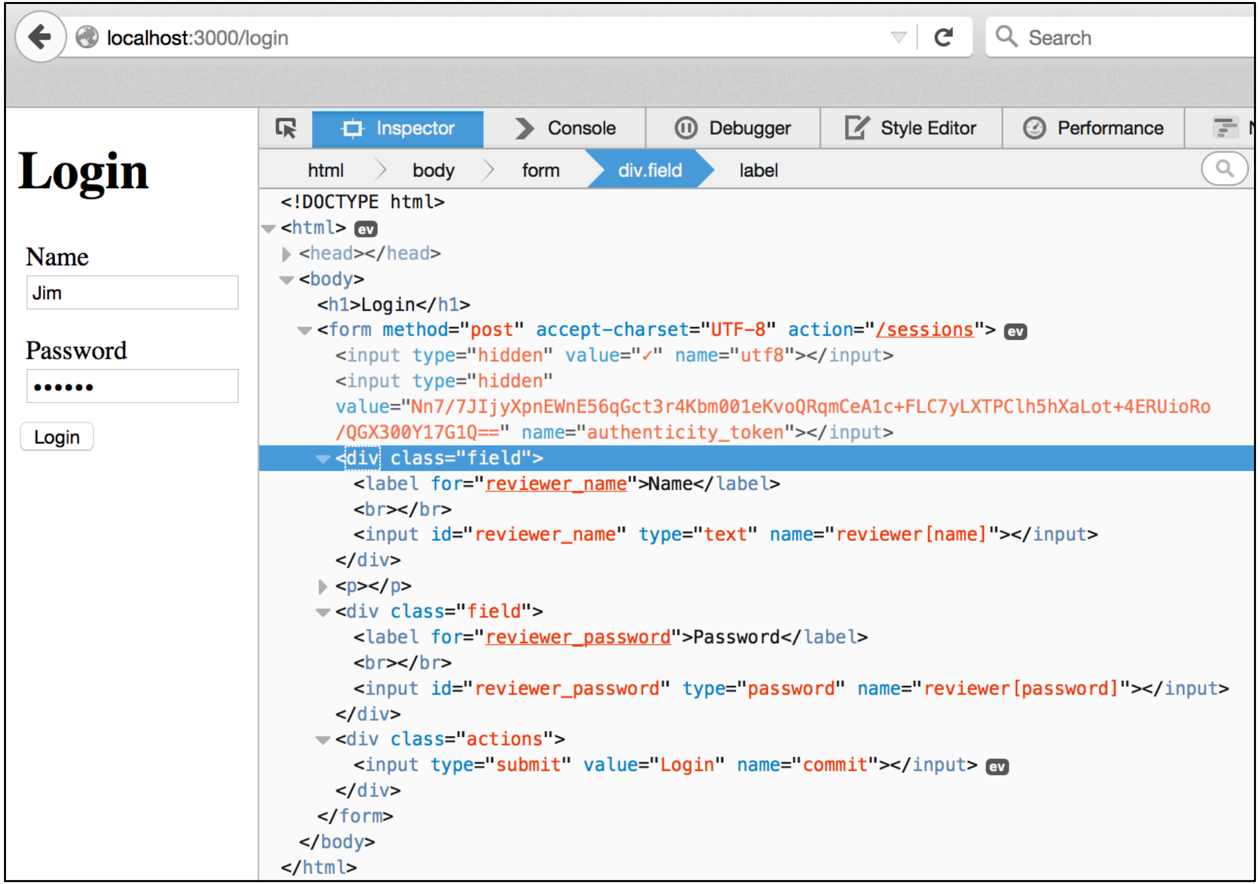
- Sessions controller and view
- Locking down our app

views/sessions/new.html.erb



```erb
<h1>Login</h1>

<%= form_for(:reviewer, url: sessions_path) do |f| %>

  <div class="field"><%= f.label :name %> <br/> <%= f.text_field :name %></div>

  <p/>

  <div class="field"><%= f.label :password %> <br/> <%= f.password_field :password %></div>

  <div class="actions"><%= f.submit "Login" %></div>
<% end %>
```

Login Page



Login

Name

Jim

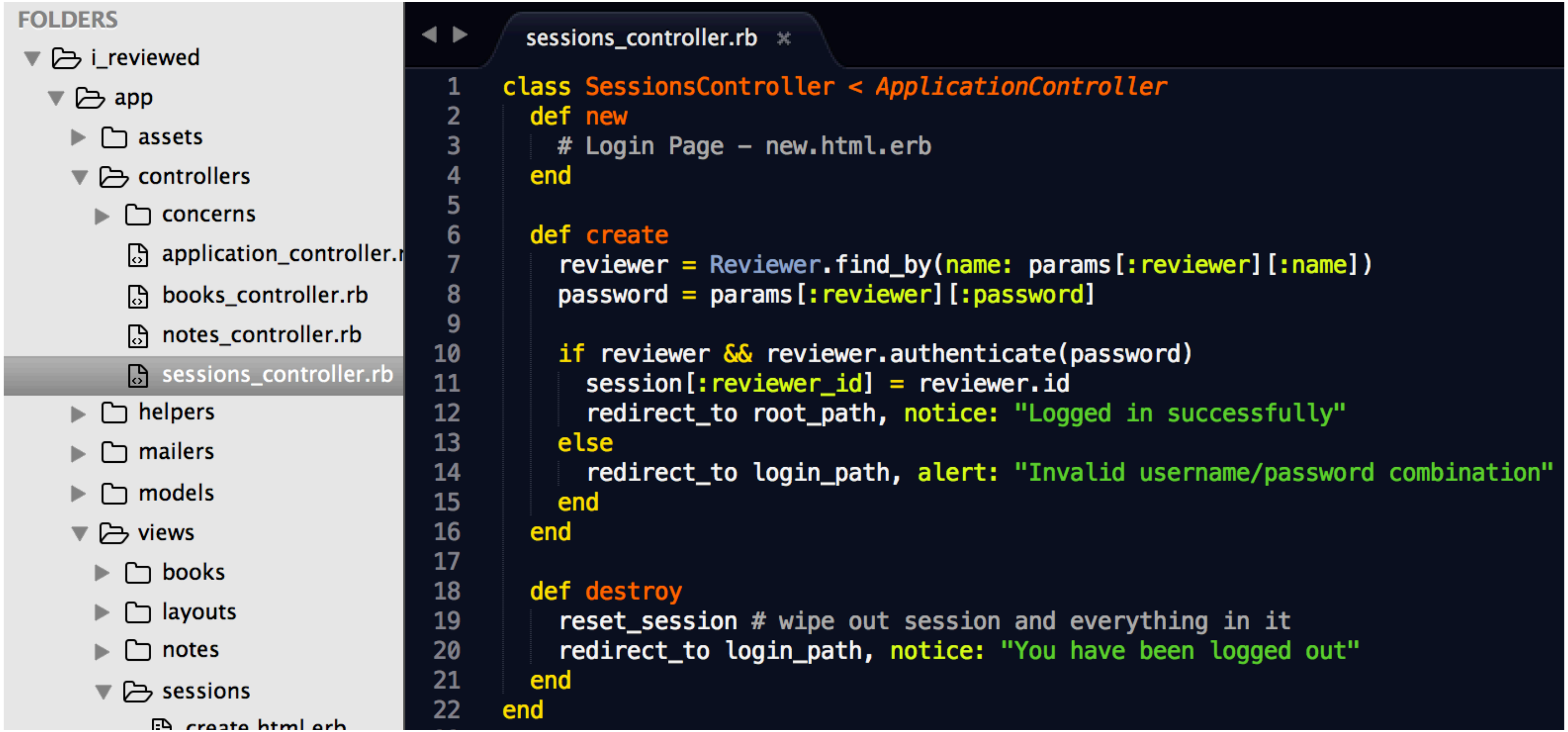
Password

••••••

Login

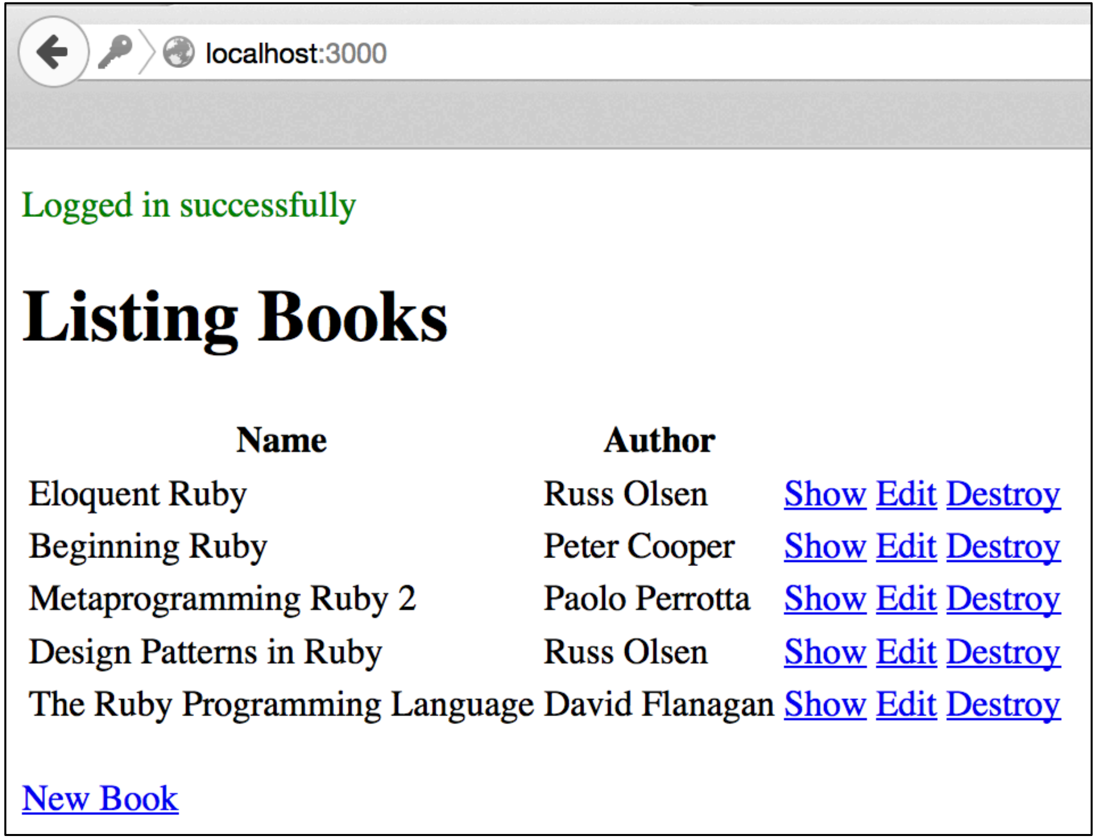
```
<!DOCTYPE html>
<html> ev
  <head></head>
  <body>
    <h1>Login</h1>
    <form method="post" accept-charset="UTF-8" action="/sessions"> ev
      <input type="hidden" value="✓" name="utf8"></input>
      <input type="hidden"
      value="Nn7/7JIjyXpnEWnE56qGct3r4Kbm001eKvoQRqmCeA1c+FLC7yLXTPClh5hXaLot+4ERUioRo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

Sessions Controller



```ruby
class SessionsController < ApplicationController
  def new
    # Login Page - new.html.erb
  end

  def create
    reviewer = Reviewer.find_by(name: params[:reviewer][:name])
    password = params[:reviewer][:password]

    if reviewer && reviewer.authenticate(password)
      session[:reviewer_id] = reviewer.id
      redirect_to root_path, notice: "Logged in successfully"
    else
      redirect_to login_path, alert: "Invalid username/password combination"
    end
  end

  def destroy
    reset_session # wipe out session and everything in it
    redirect_to login_path, notice: "You have been logged out"
  end
end
```

Logged In



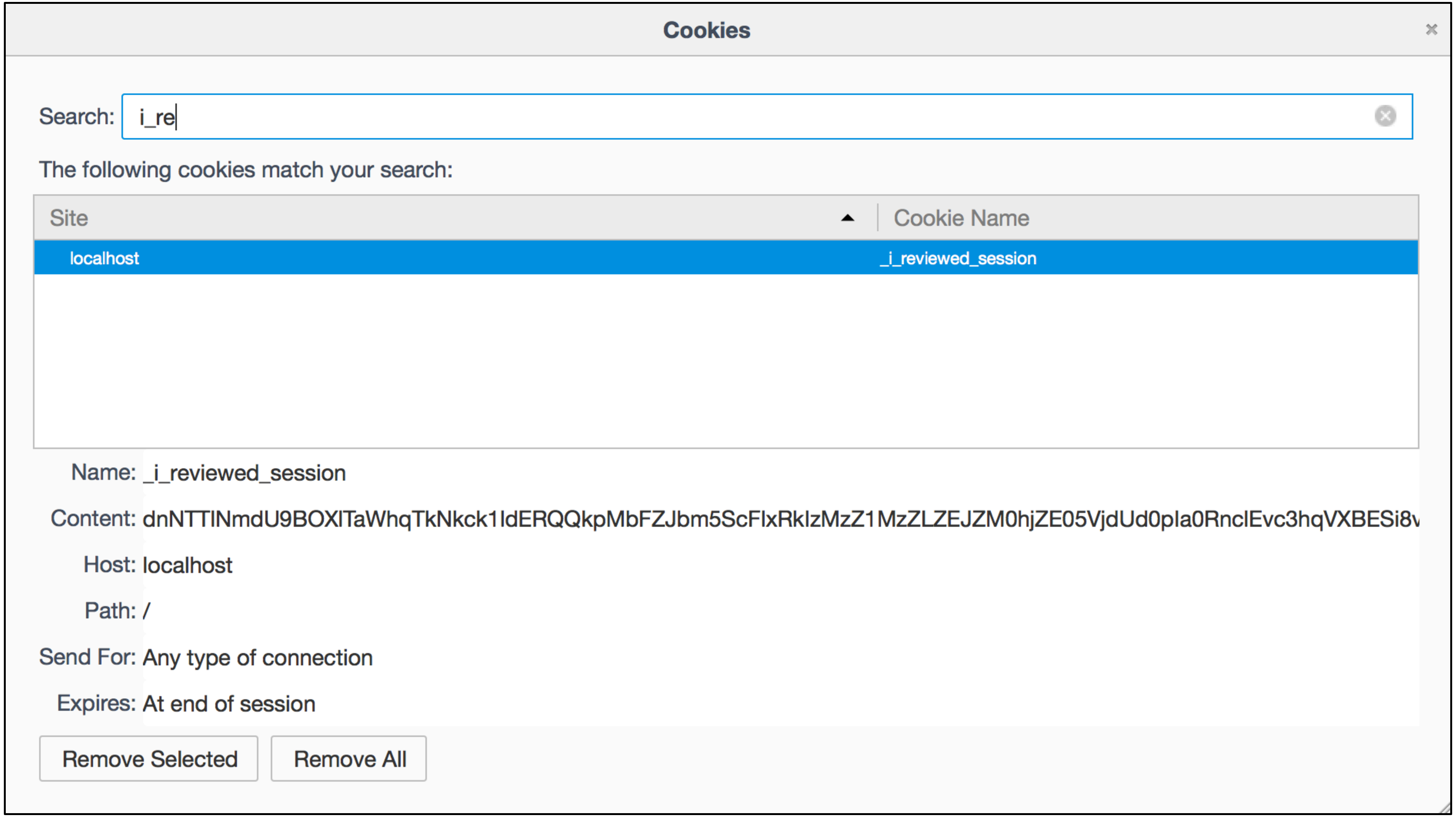
Logged in successfully

Listing Books

Name	Author	
Eloquent Ruby	Russ Olsen	Show Edit Destroy
Beginning Ruby	Peter Cooper	Show Edit Destroy
Metaprogramming Ruby 2	Paolo Perrotta	Show Edit Destroy
Design Patterns in Ruby	Russ Olsen	Show Edit Destroy
The Ruby Programming Language	David Flanagan	Show Edit Destroy

New Book

Cookie in the Browser



Search: i_re

The following cookies match your search:

Site	Cookie Name
localhost	_i_reviewed_session

Name: _i_reviewed_session

Content: dnNTTINmdU9BOXITaWhqTkNkck1IdERQQkpMbFZJbm5ScFlxRkIzMzZ1MzZLZEJZM0hjZE05VjdUd0pIa0RncIEvc3hqVXBESi8v

Host: localhost

Path: /

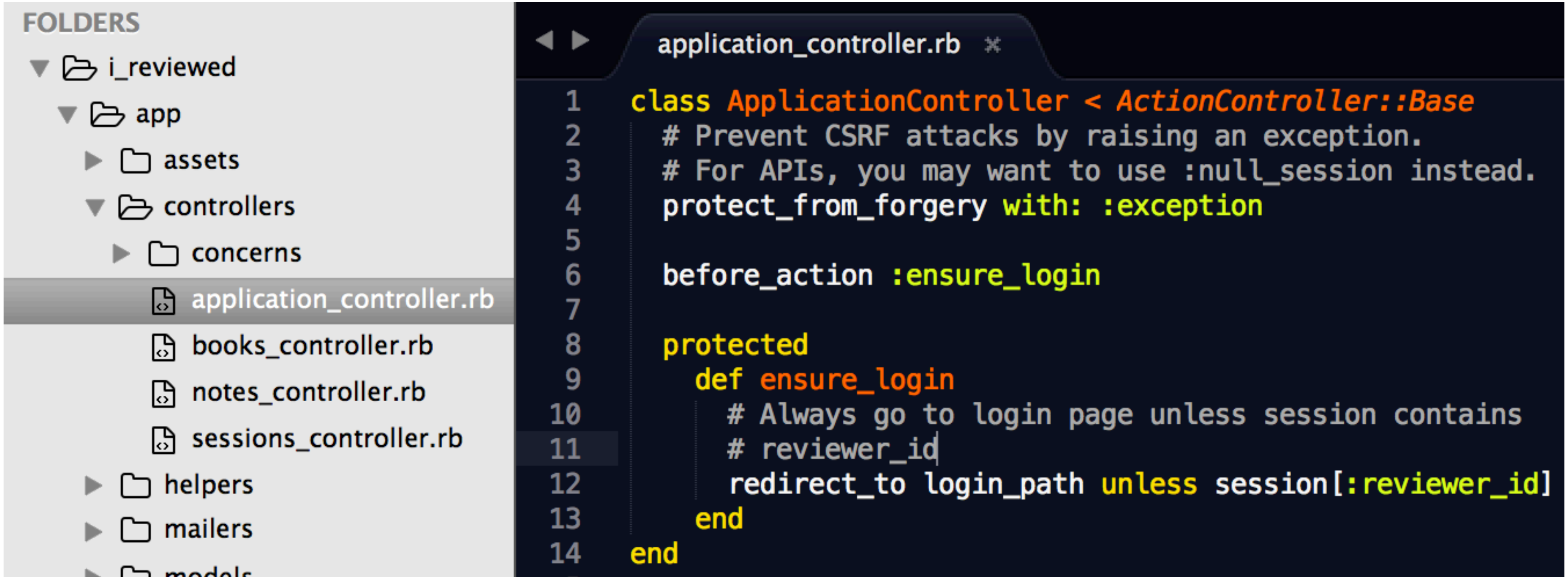
Send For: Any type of connection

Expires: At end of session

Locking Down The App

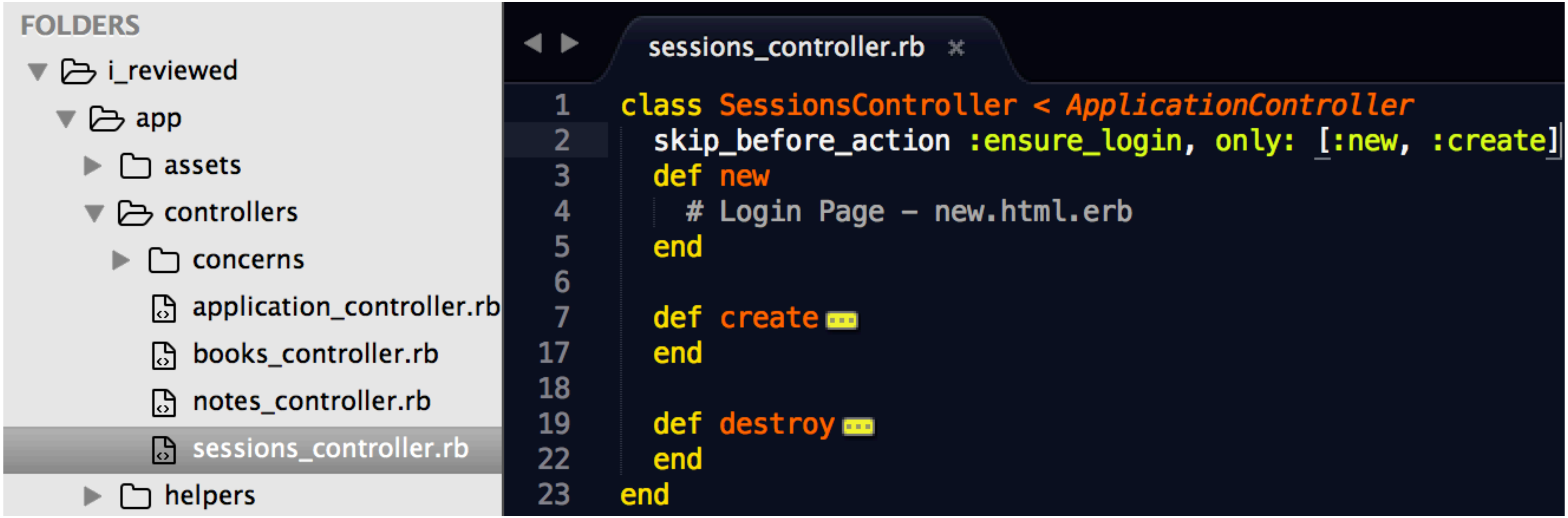
- We can have a `before_action` in the `ApplicationController` (from which all the other controllers inherit) that will make you login if you are not yet logged in
- But if everything is blocked off – how will we get to the login page? Hmm…
- Controllers can override `before_action` with `skip_before_action`

application_controller.rb



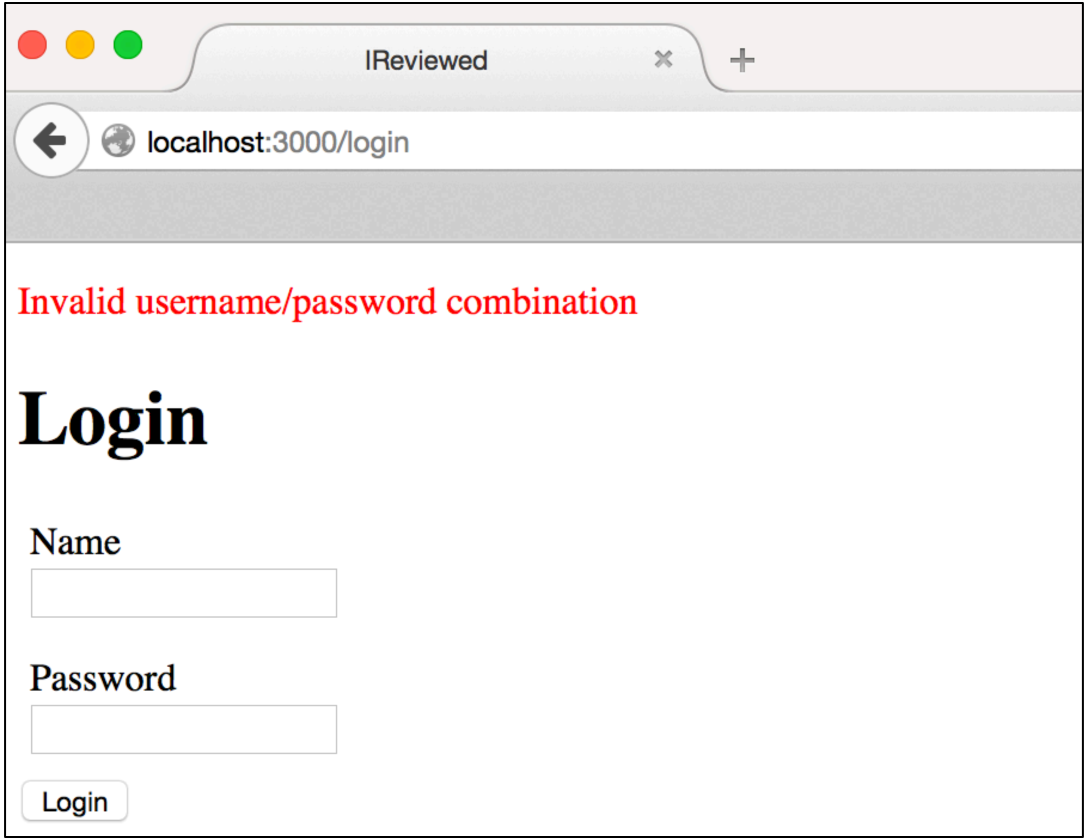
```ruby
class ApplicationController < ActionController::Base
  # Prevent CSRF attacks by raising an exception.
  # For APIs, you may want to use :null_session instead.
  protect_from_forgery with: :exception

  before_action :ensure_login

  protected
    def ensure_login
      # Always go to login page unless session contains
      # reviewer_id
      redirect_to login_path unless session[:reviewer_id]
    end
end
```

sessions_controller.rb



```ruby
class SessionsController < ApplicationController
  skip_before_action :ensure_login, only: [:new, :create]
  def new
    # Login Page - new.html.erb
  end

  def create …
  end

  def destroy …
  end
end
```

Unsuccessful Login



Invalid username/password combination

Login

Name

Password

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