

CHAPTER 8

We are going to implement a basic, but functional log-in system.

Web applications requiring user login must use a *session* → semi-permanent connection between two computers (ex. client computer running a web browser & server running Rails).

We'll use the Rails method called **session** to make temporary sessions that expire automatically on browser close (most common way is using cookies, which we'll do in Ch. 9)

We generate a sessions controller with the following line of code:

- **rails generate controller Sessions new**

HTTP request	URL	Named route	Action	Purpose
GET	/login	login_path	new	page for a new session (login)
POST	/login	login_path	create	create a new session (login)
DELETE	/logout	logout_path	destroy	delete a session (log out)

Table 8.1: Routes provided by the sessions rules in [Listing 8.2](#).

Having defined the relevant controller and route, we then filled in the view for new sessions, i.e., the login form...

We then define a minimalist create action for the Sessions controller, along with empty new and destroy actions (Listing 8.6).

Authenticate Users: So, `params[:session][:email]` is the submitted email address and `params[:session][:password]` is the submitted password. That means inside create action the params hash has all info needed to authenticate users by email and password! Inside create action:

- **`user = User.find_by(email: params[:session][:email].downcase)`
`if user && user.authenticate(params[:session][:password])`**

We created an error flash for an invalid login attempt, so we generate a test...

- **rails generate integration_test users_login**

To run only test file using rails test, we can use the following

- **rails test test/integration/users_login_test.rb**

Logging in: we'll log the user in with a temporary session cookie that expires automatically upon browser close. Logging a user in is simple with the help of the session (hash like) method defined by Rails.

Finding current user in the session:

- `@current_user ||= User.find_by(id: session[:user_id])`

We then changed layout links which depend on whether user is logged in or not.

Log in user upon signup

- In `def create` add the following line `log_in @user`

Logging out: Because the “Log out” link has already been defined (Listing 8.19), all we need is to write a valid controller action to destroy user sessions.

- In the `log_out` method in the Sessions helper module, add the lines:
 - `session.delete(:user_id)`
 - `@current_user = nil`
- Put the `log_out` method to use in the Sessions controller’s `destroy` action
 - `log_out`
 - `redirect_to root_url`

After that, we tested the user logout and were in the green.

That’s everything...Don’t forget to merge, push, and deploy!

<https://taylorchapter3.herokuapp.com/>