

## Lab 6

164

spring 2023

This lab is a follow-along. For demo, you should show the TA your login view.

### 1. Make sure Rails is installed

```
> gem install rails
```

### 2. Create a new Rails application

```
> rails new newapp
```

### 3. Run server

```
> rails server
```

if you see Rails version page, then Rails works fine

### 4. Manually create a "access" controller for login function

create three routes in config/routes.rb

```
get '/access/login'  
get '/access/authenticate'  
get '/access/logout'
```

create controller: app/controller/access\_controller.rb

```

class AccessController < ApplicationController
  def login
  end

  def authenticate
    #p [params[:userid], params[:password]]
    if params[:userid] == "yuan" && params[:password] == "123"
      render(:welcome)
    else
      flash[:notice] = "login failed"
      redirect_to '/login'
    end
  end

  def logout
    flash[:notice] = "logged out"
    redirect_to '/access/login'
  end
end

```

note: if login successful, render a welcome page  
 if login failed, set flash[:notice], it is a session variable that lives for two visits (after next visit, it will be gone)  
 then redirect to login

create necessary views:

first, create folder "access" in app/views folder

then create:

login.html.erb

```

<% if flash[:notice] %>
<h2> <%= flash[:notice] %></h2>
<% end %>
<h1>please login</h1>
<form action='/access/authenticate' method="getx">
  <input type="text" name="userid"><br>
  <input type="password" name="password"><br>
  <input type="submit" value="login">
</form>

```

this view, first display flash message if any

- create welcome page with login link on it

welcome.html.erb

```
<h1> welcome </h1>
<a href="/access/login">login</a><br>
<a href="/access/logout">logout</a>
```

modify layout file and add anything you like: (this is optional)

application.html.erb

```
<!DOCTYPE html>
<html>
  <head>
    <title>App1</title>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <%= csrf_meta_tags %>
    <%= csp_meta_tag %>

    <%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
    <%= javascript_importmap_tags %>
  </head>

  <body>
    <h1> this is my layout file</h1>
    <%= yield %>
  </body>
</html>
```

## 5. Use the rails generator to generate the above "access" controller

```
> rails generate controller NewAccess login authenticate logout
```

observe the generated code and naming convention, compare the difference with your manually created code.

## 6. add the missing code to the generated controller to make it function the same

## 7. check current route:

```
> rails routes
```

## 8. add random route to the application:

for root page, add route:

```
get '/', to: 'root#main'
```

and add missing controller

create a controller: root\_controller.rb

```
class RootController < ApplicationController
  def main
  end
end
```

create root folder under views directory

create main.html.erb file with any content.

In root\_controller.rb, redirect to other action of other controller:

```
class RootController < ApplicationController
  def main
    redirect_to controller: 'access', action: 'login'
  end
end
```

or just

redirect\_to '/access/login'

## 9. user helper method to replace HTML tags

for example: for <a> tag, use "link\_to" method:

in welcome.html.erb, try to use "link\_to" like this:

```
<h1> welcome </h1>
<a href="/access/login">login</a><br>
<%= link_to "login", access_login_path %><br>
<a href="/access/logout">logout</a>
```

observe how link\_to is used:

"login" is the clickable text

"access\_login\_path" is a method that represent the '/access/login'

## 10. use instance variable to pass data from controller to view

after successfully logged on, in welcome view, show the userid/password that the user entered.

access\_controller.rb

```
class AccessController < ApplicationController
  def login
  end

  def authenticate
    #p [params[:userid], params[:password]]
    @userid = params[:userid]
    @password = params[:password]
    if params[:userid] == "yuan" && params[:password] == "123"
      render(:welcome)
    else
      flash[:notice] = "login failed"
      redirect_to '/login'
    end
  end

  def logout
    flash[:notice] = "logged out"
    redirect_to '/access/login'
  end
end
```

in welcome.html.erb

```
<h1> welcome </h1>
userid: <%= @userid %><br>
password: <%= @password %><br>
<a href="/access/login">login</a><br>
<%= link_to "login", access_login_path %><br>

<a href="/access/logout">logout</a>
```

## 11. try "resource route"

add the following statement to routes.rb

```
resources :students
```

then check routes by running

> rails routes

observer what new routes you have

there will be 7 new routes added by this resources statement

but controllers#actions are missing, you can add controllers#actions if you want

**Finally, add this line to your routes.rb:**

root "access#login"

This will make your '/' path go to login

**To see your app working, start the app and go to localhost:3000/. Then log in with username 'yuan' and password '123' (That's your demo).**

**12. Rails document is at: <https://guides.rubyonrails.org/>**

**13. If have more time, try scaffold:**

**lets say we want to create a student resource:**

firstname	lastname	email

**run:**

**rails generate scaffold Student firstname:string lastname:string email:string**

**then observe routes.rb file  
and run**

**rails routes**

**to see new routes.**

**then from browser:**

**localhost:3000/students**

**and try functions generated**

**check new code generated by scaffolding in controller/views/model/migration**