Common Express Login Confusion

- What does a route (path) do?
- When do I redirect?

This changes in the future

These answers address **server-generated pages**

- Some nuance when it comes to services
  - Later in semester
  - Common concepts, learn now!

We will cover this here

- Then move on to SPA + services

Better than working

Working code may not be good code!

- Lots of things "work"
- We want the best options
    - to be understood
    - to handle change with minimal complexity

What is the path?

Remember that Web is Request/Response

- paths can be confusing

A login path

- the path with the form?
- the path that PROCESSES the form?

Answer: Neither

- the path is what you REQUEST

Path is what you request

User makes request

- May get what expected
- May get something else

Example: Request main page `/`

- May get data and a form to change it
- May get a login form

Redirects aren't flow control

Don't use redirects like calling a function

- (BAD) "redirect to /error"

Based on a good instinct! But

- Redirects involve overhead
- Error pages are weird "pages"
    - Reload?
    - Search for?

Instead of redirecting for flow control

Common functionality in functions on server

- Ex: return HTML from a function
    - From anywhere you need that response
    - Can pass params to function!
        - For message text, etc

Redirecting POST results

When do we redirect?

- Redirecting POST results often good
    - Keeps user from reloading a page
        - that would change app state
    - Reduces repeat logic
- Redirects are always GET

Ex:

- GET `/` shows login form without session
- POST `/login` redirects to `/` on success

Maintaining info

Another common case of redirecting:

- Sending to a complex login process
    - Maintaining desired path
- Not in assignment

When you have many pages that require login

- Redirect to/return a login form
- Pass requested URL in query param
- After successful login
    - User is redirected to original request path

Imagine how you would implement this

- GET /login displays a login form
    - Will POST to /login
- These exist: GET /users, GET /stuff, GET /more
    - They only display for logged in users
    - Other users are redirected
        - To `/login?url=THE_PATH_THEY_REQUESTED`
        - Example: `/login?url=/users`
- After a successful POST to /login
    - If url was supplied, redirect user to that path
    - If no url query param, redirect user to `/`

Section Summary - Server Generated HTML

- Server generated sites
  - Can mingle static+dynamic pages
- Ultimately generate HTML
  - Lots of tedium that libraries can make easier

Section Summary - Session Data

session

- Connects multiple requests from a user+browser
- Stored on server
- Usually based on session id cookie shared with browser

Section Summary - State

state

- Values that can change in app
    - static files don't depend on state
- session data is just one kind of state
- When server-generated HTML
    - Data on server