Pages and Views on a SPA

An MPA has multiple-pages

- it's in the name

A SPA has a single-page

- still in the name

How do we present multiple "views" for the user?

Options for "Views"

- Conditional
    - Have state and decide what to show based on it
- A routing library
    - Does conditional work + deeplinking
    - react-router is most common choice for React

Conditional rendering

```jsx
function App() {
  const [page, setPage] = useState('Home');

  return (
    <div className="app">
      <Nav page={page} onClick={setPage}/>
      { page === 'Home' && <Home/> }
      { page === 'About' && <About/> }
      { page === 'Privacy' && <Privacy/> }
    </div>
  );
}
```

Changing Pages

Following a normal `<a>` tag link will have two effects:

- Loads a different url
- Is a page load

Changing URLs

Changing urls is a big deal!

Normally, this will be a different page entirely

- not the SPA

You can have issues here!

- create-react-app dev-server will still show the SPA
- but a static file server may not

Depends on the configuration of the webserver

Page Load

Even if the server returns the same SPA

- You've done a page load
- Like hitting "reload/refresh"
- All state is reset to initial state

A common source of "weird" bugs

- You don't want this behavior in a SPA

Links in a SPA

If you have a SPA and a `<a>` tag link

- `e.preventDefault()`
    - Don't navigate
- update state instead

You may recall a similar issue with forms in SPA

Always starting in same initial state

You may have seen this:

- you are on a web page
- you save the link
    - for later use
    - to share

But the link gets you the page in a different state

- without whatever you had selected shown

Page loaded in initial state

Same thing as "reload/refresh"

- State was lost
- Page loaded in initial state

This _is_ fixable

- but not trivial

Deeplinking

Deeplinking

- a url that will give SPA different state when loaded
- page has to check url on load to change state
- page has to change url when state changes
    - without triggering a page load
- usually only done for significant state changes
    - "significant" is subjective

Deeplinking via hash fragments

Remember hash fragments in urls?

- Default: link to element with id in page
- Does not trigger a page load

One simple deeplinking solution:

- Nav links update hash fragment (`<a href="#about">About</a>`)
- on page load, check hash fragment
  - then set initial state
  - `document.location.hash`

More with hash fragment

- Link-based gives only bit of state
- Only changes on navigation
- But we have other options!

Have a format for state you want to save

- Ex: `#cats/details/maru`
- when you want to save a current state
    - assign to `document.location.hash`
        - don't forget `#` at start
    - if link/form, remember to preventDefault

Can also deeplink via path

- Requires server still serve the same SPA
    - create-react-app dev-server does
    - most servers will require extra config
- Don't assign to `document.location.pathname`
    - it will navigate
- Instead we can add to the browser history stack
- on page load, read `document.location.pathname`
- Can ALSO react to back button on the same page
    - can save more info in the history `state` object

MDN Working with the History API

A few details of path-based deeplinking

- When state you want to save changes
    - call `history.pushState()`
    - Ex: `history.pushState({}, '', '/about')`
    - or `.replaceState()` to not spam history
        - Looking at you, slideshows
    - url bar will change
        - no navigation or page load!
- on page load read `document.location.pathname`
- "path" will be fake - it's whatever info you want
    - not a path to your HTML file
    - server needs to understand this.

Deeplinking Decisions

Path-based is often more desirable than hash-based

- but more involved to implement

What state is "significant"?

- What does the user expect?
- What change would confuse the user?

Can the user get "stuck" in a bad state?

Design vs Implementation

Having Deeplinking has an impact on UX

Implementing Deeplinking is outside the scope of this course

- even react-router might require server changes
- a lot of concepts to process
- using the fragment of the url avoids the need for server changes

No deeplinking required for the final

- Consider what to be linkable in your designs

Summary - Views

- SPA is all one page
- "Navigation" is a state change
    - not actual browser navigation
- preventDefault on links
    - update state instead

Summary - Deeplinking

- SPAs will use initial state when loaded
  - making it hard to share an app with a state
- Deeplinks used by SPA to load alternate state
- Requires Decisions
  - What state is significant?
  - How to represent it as a url?
- Requires SPA changes
  - when to update url with current state
  - reading url on load to set state
- Possibly server changes