

# Pages and Views on a SPA

An MPA has multiple-pages

- In the name

A SPA has a single-page

- Still in the name

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

How do we handle "views" when

- User "reloads" in browser?
- User shares/bookmarks "page"?

# Options for "Views"

- Conditional
  - Decide what to show based on state
- A Routing library
  - Does conditional work + deeplinking
  - react-router is most common choice for React

# Conditional rendering

```
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:

- Loads a different url
- Is a page load
  - Loses JS-based state!

# Changing URLs

Changing URLs is a big deal!

- **Usually a different page (not the SPA)**

You can have issues here!

- `create-react-app` dev server still shows SPA
  - Only used in development
- But a static file server will not
  - Unless configured to

# Page Load Means new JS State

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 `<form>` submit

# **Always starting in same initial state**

You may have seen this:

- You are on a web page
- You get the URL of page
  - For later use
  - To share

But the link loads 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

- 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

# Implementing Deeplinking

- On your own
- Using a library
  - Example: `react-router`, `@tanstack/router`

# Deeplinking Concepts

Libraries are often the right path

- But this course teaches concepts
- I banned react-router in final b/c this

We will not go deep (ha!)

Key ideas:

- Change url with some state changes
  - But don't reload page!
- On load, match state to url
  - Must load same SPA code for different urls

# Changing the url without page load

Two options:

- Use a hash reference (change the hash fragment)
- Change the url path

# Changing a hash fragment

## Benefits:

- Normal links do the job
- `<a href="#profile">Profile</a>`
- Automatically no page load!
- No server changes needed (always the same file)

## Issues:

- Less good for SEO
- Cannot handle server-side (for SSR)
  - Can impact future migration

# Hash Implementation

When navigating

- Use `<a href="#sometarget">`
- To update without navigation
  - Update `document.location.hash`
  - Or use `pushState` (coming up)

On page load

- Set initial state based on `document.location.hash`

# Changing the Url Path

## Benefits:

- "Cleaner" URLs
- Better SEO
- Potential for server-side help
- Option to redirect at server
  - "future-proof"

## Issues:

- Requires server configuration
  - to serve same HTML page
- Can't use normal links in-app



# Path Change Implementation - Server

Server-side (express example):

```
// Last route so api routes are checked first
app.get('*', (req,res) =>{
  res.sendFile('./build/index.html');
});
```

Already exists in create-react-app dev-server

- But not YOUR server.js (yet)

# Path Change Implementation - Client

When "navigating"

- Use `window.history.pushState({}, '', url);`
  - Initial object is a "state" (different state)
    - We won't use
  - Empty string, because they messed up
  - `url` can be absolute or relative path

On page load

- Set initial state based on `document.location.pathname`

# Using pushState impacts Back button!

Back button annoying to SPA

- Think any slideshow where you "Back" 30+ times
- `window.history.pushState()` manages it
- Also `window.history.replaceState()` also does

Navigation in page fires `popstate` event

- "Navigation" includes "Back"
- You have to add listener to `window`, not via React
  - `useEffect`
  - Don't forget to clean up

## 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
  - Rarely all state, just select "mode"
- Requires SPA changes
  - on load: set initial state
  - on mode change: update url
- Possibly server changes
  - If using url changes