

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

```
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

Implementing Deeplinking

- On your own
- Using a library
 - Example: react-router

Deeplinking Concepts

As with much, libraries are often the right path

- But this course teaches concepts

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
- `Profile`
- 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 ``
- To update without navigation
 - Update `document.location.hash`

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`

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