

# 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

# Two options for deeplinking

- Hash-based urls
  - URLs change hash fragment after filename
- Path-based urls
  - URLs change the path/filename

Either way, need to consider

- Page Load
- Navigation via link/form
- Changing URL without link/form
- Back/Forward

# Hash-based Navigation

- Links/form submits can use hash relative paths
  - Browser will not navigate
  - No need to `preventDefault()`
- Still need to consider
  - Page load
  - Changing URL without link/submit
  - Back/Forward

# Hash-based Page Load

- Can set initial state using `window.location.hash`

```
function App() {  
  // A very simple example  
  // where we use a directory-like structure in hash fragment  
  // Like: #/ or #/about or #/privacy  
  const initialPage = window.location.hash || '#/';  
  const [page, setPage] = useState(initialPage);  
}
```

# Changing Hash in URL without link/submit

- App might change "view" for reasons other than user clicking link/submitting form
  - Want to change URL to match new state
  - So user can save/share/reload that url
- Can simply assign to `window.location.hash` when needed

```
window.location.hash = '#/about';
```

- Will not cause page load



# Managing back/forward with hash-based URL

- Link/submit/`window.location.hash` updates URL
  - We only set state on initial page load
  - We need to detect when hash changes
- There is a `hashchange` event!
  - But it only happens to `window`
  - Not on an element React controls
  - Can't use `onXXX` to listen for event
- We can use a normal event listener
  - Add it in a `useEffect` in App.jsx
  - Make sure to clean up!
  - Event will update state based on new hash

# Listening to hashchange event

```
function App() {  
  useEffect( () => {  
    const onHashChange = () => {  
      // Very simple example  
      setPage(window.location.hash);  
    };  
    window.addEventListener( 'hashchange', onHashChange);  
    return () => { // Cleanup  
      window.removeEventListener('hashchange', onHashChange);  
    };  
  }, []); // empty dependency array, run only once  
}
```

# Hash-based url routing

Remember: Normally want libraries to do this

- I'm making sure you understand what they do
  - And could write it if you needed
  - Deeplinking not required for final
  - But if you do, must do WITHOUT library

Hash-based routing

- Easier for developer
  - No server-side config
- Not good for SEO
- Not good for logging/analysis

# Path-based URL Routing

Same needs

- Set state from url on initial Page Load
- Manage links/form submits
- Changing URL without link/submit to show state
- Manage state change on Back/Forward

# Path-based Server configuration

Key issue:

- Server considers different paths = different files
- Loading a different path = 404 :(

Server config can change this

- Dev server already configured like this!
- But your server.js is not!

Server-side (express example):

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

# Path-based Routing Initial Page Load

- Can set initial state using `window.location.pathname`

```
function App() {  
  // A very simple example  
  // Where our view in state is a path (no filename)  
  // Like: / or /about or /privacy  
  const initialPage = window.location.pathname || '/';  
  const [page, setPage] = useState(initialPage);  
}
```

# Path-based Routing Navigation

- We must `preventDefault()` on links/submits
- But then URL doesn't change
  - Must push new entry in the browser "history"
  - Uses browser `history` API

# Browser History API

- `window.history.pushState()`
  - 3 arguments (weird)
  - First is "state" (but not our state)
    - We will url instead, so just pass `null`
  - Second is always
    - Historical "mistake"
  - Third is URL
    - Absolute or relative
    - Can be hash!

```
window.history.pushState(null, '', '/about');
```



## **Path-based Routing URL changes**

- Changing URL path for link/submit or because app wants to is all the same
  - Push new url into browser history
- Changing URL doesn't change state though
  - You need to change state in same code that changes URL

# Path-based Routing Changes

- Just like `hashchange`
  - a `popstate` event happens
- Details are different
  - `hashchange` happens when you change url
  - `popstate` happens when you Back/Forward
    - Including w/urls with hashes
    - Both a `hashchange` and `popstate` then
- Not a drop-in replacement for each other
  - Should only use one of them
- Listen just like w/ `hashchange`

# Listening to popstate

```
function App() {  
  useEffect( () => {  
    const onPopState = () => {  
      // Very simple example  
      setPage(window.location.pathname);  
    };  
  
    window.addEventListener('popstate', onPopState);  
  
    return () => { // Cleanup  
      window.removeEventListener('popstate', onPopState);  
    };  
  }, []); // empty dependency array, run only once  
}
```

Can be used for Hash-based or Path-based

# Path-based url routing

Remember: Normally want libraries to do this

- MUST HAVE SERVER CONFIGURATION!
  - Otherwise Back/Forward will work
    - But Reload will 404
  - Libraries don't configure server!
- Better for SEO
- Better for logging/analysis

## 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 path changes