Pages and Views on a SPA

- An MPA/website has multiple-pages
- A SPA has a single-page

How does a SPA look like multiple "pages" to user?

How do we handle?

- User "reloads" in browser?
- User shares/bookmarks "page"?
- User uses Back/Forward?

Default: Browser reloads/leave SPA

- See default/initial state/content

Options for "Views"

A **Routing library**

- Provides **deep links**
    - Diff URLs for diff SPA content
    - App updates state to reflect URL
- `react-router`; `@tanstack/router`
- Let's understand the concepts without a library
    - Use libraries once not magical

"Pages" via Conditional Rendering

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

  return (
    <>
      { page === '/' && <Home setPage={setPage}/> }
      { page === '/about' && <About setPage={setPage}/> }
      { page === '/privacy' && <Privacy setPage={setPage}/> }
    </>
  );
}
```

- Components would be passed and call `setPage()`

Pages vs "Pages"

Load Page:

- Browser makes page request to webserver
- New JS state for rendered page is created

Service Calls:

- Browser does NOT leave actual page
- JS state updated: existing HTML updated
- May LOOK like a different "page"

Link, Forms, Reload, Back/Forward:

- Default: **actually Load Page**

Dealing with the Confusion: Deep linking

Deep linking is having a url

- Reflects the content shown to user
- Shows content matching url (re)loaded
- Works with Back/Forward

Addresses this problem SPAs created

- Not normally needed without SPA

The Browser History API

- Allows us to change the "history stack" in browser
- Also changes current page URL in browser
    - Without triggering a reload/navigation!
- Back/Forward over added entries
    - Allows Back/Forward
    - Will not trigger a reload/navigation!
    - We will have to detect and change content

window.history.pushState() to add to history stack

```
// window.history.pushState(state, '', url);
window.history.pushState(null, '', '/cats');
```

- `state` param is not our state
    - Whatever content we want
    - Only works with back/forward
    - We want url to be enough for content
    - We send `null` and ignore
- Second param historical mistake
    - `''` is safe (per MDN)
- `url` is absolute or relative path

Modifying our App

- On changing "page" with link/form
    - `preventDefault` of link
    - Push deep link URL onto history stack
    - Update state to render new "page"
- On page load of deep link URL
    - Load SPA HTML anyway
    - Update state to render "page" matching URL
- On Back/Forward
    - Update state to render "page" matching URL

Changing the URL using History API

```
function Nav({ setPage }) {
  function changePage(e) {
    e.preventDefault(); // stop actual browser navigation
    // change url to match path from link
    window.history.pushState(null, '', e.target.pathname);
    // Update our state to show different content
    setPage(e.target.pathname);
  }

  return (
    <nav className="nav">
      <a href="/" onClick={ changePage } >Home</a>
      <a href="/about" onClick={ changePage } >About</a>
    </nav>
  );
}
```

Behavior after setting History on page change

- 😻 App can change "page"
- 😻 URL update on "page" change
- 😿 Content doesn't match URL on load/reload

Back/Forward over pushed history entries

- 😻 Does NOT leave the app unexpectedly
- 😿 Content doesn't match new URL

Set state on load based on URL

App.jsx

```
function App() {
  const path = document.location.pathname;
  const [ page, setPage ] = useState(path);

  return (
    <>
      <Nav setPage={setPage}/>
      { page === '/' && <Home setPage={setPage}/> }
      { page === '/about' && <About setPage={setPage}/> }
      { page === '/privacy' && <Privacy setPage={setPage}/> }
    </>
  );
}
```

After changes on page load

- 😻 App can change "page"
- 😻 URL update on "page" change
- 😻 Content matches URL on load/reload

Back/Forward over pushed history entries

- 😻 Does NOT leave the app unexpectedly
- 😿 Content doesn't match new URL

`popstate` event when Back/Forward

Back/Forward over pushed history entries

- Does NOT yet change state
- Will fire a `popstate` event
    - on `window`
    - `window` is not controlled by React

We need to add an eventListener to window

- Outside of React
- When? On Page Load
    - `useEffect()`

Adding popstate listener

```
function App() {
  const path = document.location.pathname;
  const [page, setPage] = useState(path);

  useEffect( () => {
    window.addEventListener('popstate', () => {
      setPage(document.location.pathname); // on back/forward
    });
  }, []);

  return (
    <>
      <Nav setPage={setPage}/>
      { page === '/' && <Home setPage={setPage}/> }
      { page === '/about' && <About setPage={setPage}/> }
      { page === '/privacy' && <Privacy setPage={setPage}/> }
    </>
  );
}
```

Listener is added twice

- Double the effect
- Doesn't BREAK anything
    - But good practice to notice and fix
    - With **cleanup function**
- Removing event listeners is a bit weird
    - `.removeEventListener()`
    - With *same* function
        - Not different function doing same thing
    - Make named handler callback

Cleanup popstate event listener

```
useEffect( () => {

  function onPageChange() {
    setPage(document.location.pathname);
  }

  // Back/Forward
  window.addEventListener('popstate', onPageChange);

  return () => { // useEffect cleanup
    window.removeEventListener('popstate', onPageChange);
  }
}, []);
```

After adding popstate listener

- 😻 App can change views ("pages")
- 😻 URL DOES change on view change
- 😻 JS State matches URL on load/reload

Back/Forward over pushed history entries

- 😻 Does NOT leave the app unexpectedly
- 😻 Does NOT cause a page load
- 😻 DOES change our page state

How do these URLs load instead of `404 Not Found`?

Deep Linking in production

- We have all the parts working, right?
    - All happy cats!

EXCEPT...

- When we do `npm run build`
- And run with only the express server
    - It only PARTIALLY works
- Navigation, back/forward, all work
    - But reloading a url like `/about` FAILS

Server SPA Configuration

Vite Dev Server returns SPA HTML/CSS/JS

- For most requests
- Including for `/about`
    - Loads /index.html
    - Loads SPA JS
    - Sets page state

Express server looks in `./dist` for file named `"about"`

- Finds none
- Returns 404

We need a special configuration

This wasn't a problem for actual static files

- Files with matching filenames existed

Wasn't a problem for dynamic server-generated HTML

- Server routes matching filenames/paths existed

We need a special configuration to return `index.html`

Config for an express server

- Other servers use their language/conventions
- Should be LAST entry before `app.listen()`
    - So other options get chance to match request

```
app.get('*', (req, res) => { // Default to sending index.html
    res.sendFile(path.join(__dirname, "dist", "index.html"));
});
```

- Any GET request (`*`) will send dist/index.html
- Last choice, so more specific matches happen instead

Summary - Views

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

Summary - Deep Linking

- SPAs will use initial state when loaded
  - Making it hard to share an app with a state
- **Deep link URLs** used to show matching content
  - Rarely all state, just content "page"
- Requires SPA changes
  - Usually done with a **routing library**
  - On load: set initial state
  - On mode change: push onto browser history
  - On Back/Forward: set app state to match URL
- Requires **server config for paths in url**