147. Nested routing in <Shoppage />

a. in <Shoppage />, import { Route } from ‘react-router-dom’

b. Put the <CollectionOverviews /> inside the Route. Note that <Shoppage /> now have an access to match, history

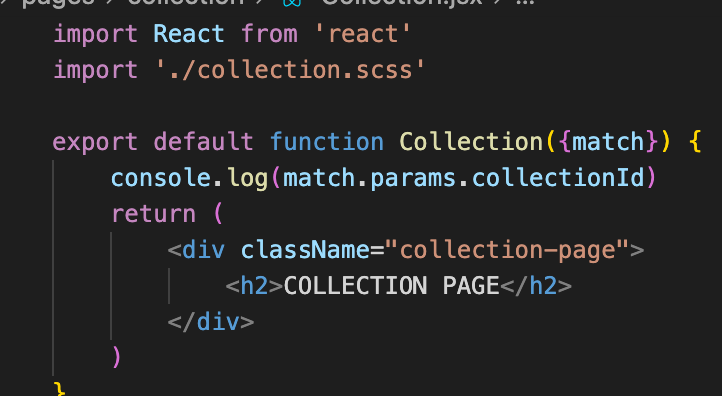
and location as the <Shoppage/> is wrapped in the <Route /> inside the <App />

c. Build the collection page

1. create the category folder and its jsx and scss files

2. Import this to <Shoppage /> afterwards





148. Correction of the Naming Convention – correction of category to collection ie; correctionId to categoryId

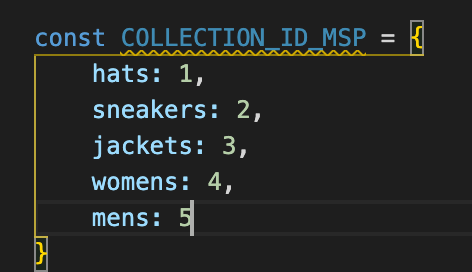
149. Collection Routing and Selector – create a Collection Selector to reach to the redux state of the collection items and get the right collection item base on match.params.collectionId. However need to convert first the id (as it is in number) to its collection name like, hats, jackets etc

a. Use the collectionId map method.

1. create the COLLECTION\_ID\_MAP in the sho-selector.js

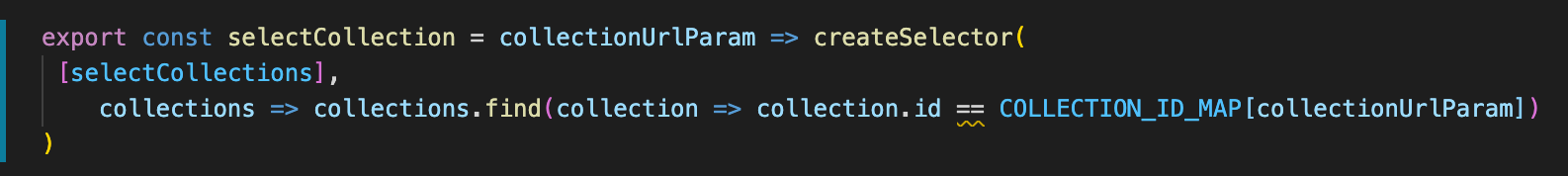
2. 

3. from above, let’s say hats = 1 etc



4. pass the string as the dynamic value of the property to get the correct id, then match it in the selector

5. create the selector



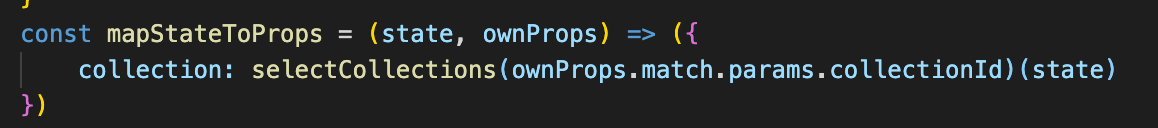
6. bring it to Collection.jsx

a. import [ connect ] from ‘react-redux’

b. import [ selectCollections ] from ‘../../redux……shop.selectors’

c. use the 2nd parameter of mapStateToProps (1st is the state), ownProps which is the props of

component that is wrapping connect



d. use the collection as props for Collection