1.detail.js import할때, 데이터바인딩까지 같이 해보자 (props랑 배열)

(App.js 파일)

import 많은 곳;

function App(){

return (

<div>

<Navbar></Navbar> (상단메뉴 레이아웃)

<Route path="/">

<Jumbotron></Jumbotron> (Jumbotron 대문 레이아웃)

<>상품3개 카드 레이아웃</>

</Route>

<Route path="/detail">

<Detail shoes={shoes}/>

</Route>

</div>

)

}

▲ 1. 여기서 <Detail shoes={shoes}/> 이렇게 원하는 state를 골라 전송하고

(Detail.js 파일)

import React from 'react';

function Detail(props){

return (

<div className="container">

<div className="row">

<div className="col-md-6">

<img src="https://codingapple1.github.io/shop/shoes1.jpg" width="100%" />

</div>

<div className="col-md-6 mt-4">

<h4 className="pt-5">{props.shoes[0].title}</h4>

<p>{props.shoes[0].content}</p>

<p>{props.shoes[0].price}원</p>

<button className="btn btn-danger">주문하기</button>

</div>

</div>

</div>

)

};

export default Detail

▲ 2. Detail 컴포넌트 안에선 props 등록하고 자유롭게 보낸 변수나 state를 쓰시면 됩니다.

2. 상세페이지 3개 만들기

 URL 주소 :

**/detail/0으로 접속하면 0번째 상품의 상세페이지**

**/detail/1으로 접속하면 1번째 상품의 상세페이지**

**/detail/2으로 접속하면 2번째 상품의 상세페이지**

(App.js 파일)

function App(){

return (

<div>

<나머지HTML/>

<Route path="/detail/0">

<Detail shoes={shoes}/>

</Route>

<Route path="/detail/1">

<Detail shoes={shoes}/>

</Route>

<Route path="/detail/2">

<Detail shoes={shoes}/>

</Route>

</div>

)

}

▲ 이런 식으로 3개의 페이지를 만들었습니다. 완성!

근데 URL 만드실 땐 반복문은 안쓰고 보통 URL 파라미터 문법을 이용해 축약을 시켜줍니다.

(App.js 파일)

function App(){

return (

<div>

<나머지HTML/>

<Route path="/detail/:id">

<Detail shoes={shoes}/>

</Route>

</div>

)

}

▲ 저번시간에 잠깐 했던 : 이거 콜론기호를 쓰시면 되는데, 이게 뭔뜻이냐면

:id 자리에 아무 문자나 입력하면 <Detail> 컴포넌트를 보여주세요~ 입니다.

-->이렇게 하면 문제점 : 상품명은 똑같이 뜨겠지 주소(링크)만 다르게 표현한거니까

그럼 Detail.js는 최종적으로 이렇게 되겠군요

(Detail.js 파일)

import React from 'react';

import { useHistory, useParams } from 'react-router-dom';

function Detail(props){

let { id } = useParams();

return (

<div className="container">

<div className="row">

<div className="col-md-6">

<img src="https://codingapple1.github.io/shop/shoes1.jpg" width="100%" />

</div>

<div className="col-md-6 mt-4">

<h4 className="pt-5">{props.shoes[id].title}</h4>

<p>{props.shoes[id].content}</p>

<p>{props.shoes[id].price}원</p>

<button className="btn btn-danger">주문하기</button>

</div>

</div>

</div>

)

};

export default Detail

그래서 이것이 리액트에서 상세페이지 만드는 패턴이라고 보면 되겠습니다.

-->이렇게 하면 문제점 : 정렬기능 넣어버리면 0번째 1번째들이 바뀜

그럼 Detail.js는 최종적으로 이렇게 되겠군요

(Detail.js 파일)

import React from 'react';

import { useHistory, useParams } from 'react-router-dom';

function Detail(props){

let { id } = useParams();

let 찾은상품 = props.shoes.find(function(상품){

return 상품.id == id

});

return (

<div className="container">

<div className="row">

<div className="col-md-6">

<img src="https://codingapple1.github.io/shop/shoes1.jpg" width="100%" />

</div>

<div className="col-md-6 mt-4">

<h4 className="pt-5">{찾은상품.title}</h4>

<p>{찾은상품.content}</p>

<p>{찾은상품.price}원</p>

<button className="btn btn-danger">주문하기</button>

</div>

</div>

</div>

)

};

export default Detail

find() 라는 ES6 신문법이 있습니다. Array 안에서 원하는 자료를 찾고싶을 때 사용합니다.

코드조각

App.js

/\* eslint-disable \*/

import logo from './logo.svg';

import React, {useState} from 'react';

import './App.css';

import { Navbar,Button,Nav,NavDropdown,Form,FormControl,Jumbotron } from 'react-bootstrap';

import Data from './data.js';

import Detail from './detail.js'

import {Link, Route, Switch} from 'react-router-dom';

function App() {

  let [shoes, shoes변경] = useState(Data);

  return (

    <div className="App">

<Navbar bg="light" expand="lg">

  <Navbar.Brand href="#home">ShoeShop</Navbar.Brand>

  <Navbar.Toggle aria-controls="basic-navbar-nav" />

  <Navbar.Collapse id="basic-navbar-nav">

    <Nav className="mr-auto">

      <Nav.Link > <Link to ="/"> Home </Link></Nav.Link>

      <Nav.Link > <Link to ="/detail">Detail</Link></Nav.Link>

      <NavDropdown title="Dropdown" id="basic-nav-dropdown">

        <NavDropdown.Item href="#action/3.1">Action</NavDropdown.Item>

        <NavDropdown.Item href="#action/3.2">Another action</NavDropdown.Item>

        <NavDropdown.Item href="#action/3.3">Something</NavDropdown.Item>

        <NavDropdown.Divider />

        <NavDropdown.Item href="#action/3.4">Separated link</NavDropdown.Item>

      </NavDropdown>

    </Nav>

    <Form inline>

      <FormControl type="text" placeholder="Search" className="mr-sm-2" />

      <Button variant="outline-success">Search</Button>

    </Form>

  </Navbar.Collapse>

</Navbar>

<Switch>

        {/\* Switch : 여러개가 맞아도 하나만 보여주셈 (택1해주셈) \*/}

<Route exact path="/">

    <Jumbotron className="background">

      <h1>20% Season Off</h1>

      <p>

        This is a simple hero unit, a simple jumbotron-style component for calling

        extra attention to featured content or information.

      </p>

      <p>

        <Button variant="primary">Learn more</Button>

      </p>

    </Jumbotron>

    <div className="container">

      <div className="row">

      {

        shoes.map((a, i)=>{

          // return <Card shoes={shoes[i]}/>

          return <Card shoes={a} i={i} key={i}/>

        })

      }

      </div>

    </div>

</Route>

<Route path="/detail/:id">

  <Detail shoes={shoes}/>

</Route>

<Route path="/:id">

  <div>아무거나 적었을 때 이거 보애주셈</div>

</Route>

</Switch>

    </div>

  );

}

function Card(props){

  return(

        <div className='col-md-4'>

          <img src={'https://codingapple1.github.io/shop/shoes'+ (props.i + 1) +'.jpg' } width="100%"/>

          <h4>{props.shoes.title}</h4>

          <p> {props.shoes.content} & {props.shoes.price}</p>

        </div>

  )

}

// src에 넣은 파일은 파일명이 변경되고 압축됨

// 근데 public 에 넣으면 보존됨 근데 절대경로로 사용해야됨

export default App;

detail.js

import React, {useState} from 'react';

import { useHistory, useParams } from 'react-router-dom';

function Detail(props){

  let { id } = useParams(); // {}가 출력됨

  // let { id,id1, id2 } = useParams();

  let 찾은상품 = props.shoes.find(function(상품){

    return 상품.id == id

  });

  let history = useHistory();

    return(

        <div className="container">

              <div className="row">

                <div className="col-md-6">

                  <img src="https://codingapple1.github.io/shop/shoes1.jpg" width="100%" />

                </div>

                <div className="col-md-6 mt-4">

                  {/\* <h4 className="pt-5">{props.shoes[id].title}</h4>

                  <p>{props.shoes[id].content}</p>

                  <p>{props.shoes[id].price}원</p> \*/}

                  <h4 className="pt-5">{찾은상품.title}</h4>

                  <p>{찾은상품.content}</p>

                  <p>{찾은상품.price}원</p>

                  <button className="btn btn-danger">주문하기</button>

                  <button className="btn btn-danger" onClick={ ()=> {

                    // history.goBack();

                    history.push('/');

                   }}>뒤로가기</button>

                </div>

              </div>

        </div>

    )

  }

export default Detail;

import React, {useState} from 'react';

import { useHistory, useParams } from 'react-router-dom';

function Detail(props){

  // useParams() : 라우터 라이브러리를 사용할 수 있는 훅

  let { id } = useParams(); // {}가 출력됨

          // let { id,id1, id2 } = useParams(); 이렇게 , 로 여러 변수 쓰기 ㄱㄴ

  // 아무튼 이 id라는 변수에다가 데이터바인딩 값을 넣어서 저기 밑의 [id]로 보내는거다

  let 찾은상품 = props.shoes.find(function(상품){

    return 상품.id == id

  });

  let history = useHistory();

    return(

        <div className="container">

              <div className="row">

                <div className="col-md-6">

                  <img src="https://codingapple1.github.io/shop/shoes1.jpg" width="100%" />

                </div>

                <div className="col-md-6 mt-4">

                  {/\* <h4 className="pt-5">{props.shoes[id].title}</h4>

                  <p>{props.shoes[id].content}</p>

                  <p>{props.shoes[id].price}원</p> \*/}

                  <h4 className="pt-5">{찾은상품.title}</h4>

                  <p>{찾은상품.content}</p>

                  <p>{찾은상품.price}원</p>

                  <button className="btn btn-danger">주문하기</button>

                  <button className="btn btn-danger" onClick={ ()=> {

                    // history.goBack();

                    history.push('/');

                   }}>뒤로가기</button>

                </div>

              </div>

        </div>

    )

  }

export default Detail;