1. Ajax

서버에 새로고침 없이 요청을 할 수 있게 도와줌

2. 요청

get 요청 : 특정 페이지/ 자료 읽기

post 요청 : 서버로 중요 정보 전달

3. Ajax 설치

- jQuery 설치해서 $.ajax() 쓰든가

- axios 설치해서 axios.get() 쓰든가\*\*\*\*\*

- 쌩자스 fetch() 쓰든가

4. axios 설치

yarn add axios

import axios from 'axios';

이제 서버에 ajax 요청을 해서 상품 데이터 3개를 더 가져올거임

1. 버튼누르면 서버에 데이터 가져오기

axios.get(‘데이터 요청할 url’);

<button className='btn btn-primary' onClick={()=>{

        axios.get('https://codingapple1.github.io/shop/data2.json');

      }}>더보기</button>

2. 성공하면 then, 실패하면 catch

<button className='btn btn-primary' onClick={()=>{

        axios.get('https://codingapple1.github.io/shop/data2.json')

        .then((result)=>{

          // console.log(result);

          console.log(result.data);

        })

        .catch(()=>{});

      }}>더보기</button>

3. axios는 json을 object로 변환해줌

<json 과 object의 차이?

json

텍스트이(가) 표시된 사진

자동 생성된 설명

object

텍스트이(가) 표시된 사진

자동 생성된 설명

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';

import axios from 'axios';

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 as={Link} to ="/"> Home </Nav.Link>

      <Nav.Link as={Link} to ="/detail"> Detail </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>

<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={a} i={i} key={i}/>

        })

      }

      </div>

      <button className='btn btn-primary' onClick={()=>{

        axios.get('https://codingapple1.github.io/shop/data2.json')

        .then((result)=>{

          // console.log(result);

          console.log(result.data);

        })

        .catch(()=>{});

      }}>더보기</button>

    </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>

  )

}

export default App;