===========================================================

\src\api\todoApi.js

===========================================================

import axios from "axios"

export const API\_SERVER\_HOST = 'http://localhost:8080'

const prefix = `${API\_SERVER\_HOST}/api/todo`

export const getOne = async (tno) => {

const res = await axios.get(`${prefix}/${tno}` )

return res.data

}

export const getList = async ( pageParam ) => {

const {page,size} = pageParam

const res = await axios.get(`${prefix}/list`, {params: {page:page,size:size }})

return res.data

}

export const postAdd = async (todoObj) => {

const res = await axios.post(`${prefix}/` , todoObj)

return res.data

}

export const deleteOne = async (tno) => {

const res = await axios.delete(`${prefix}/${tno}` )

return res.data

}

export const putOne = async (todo) => {

const res = await axios.put(`${prefix}/${todo.tno}`, todo)

return res.data

}

===========================================================

\src\App.css

===========================================================

.App {

text-align: center;

}

.App-logo {

height: 40vmin;

pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

.App-logo {

animation: App-logo-spin infinite 20s linear;

}

}

.App-header {

background-color: #282c34;

min-height: 100vh;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

font-size: calc(10px + 2vmin);

color: white;

}

.App-link {

color: #61dafb;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

===========================================================

\src\App.js

===========================================================

import {RouterProvider} from "react-router-dom";

import root from "./router/root";

function App() {

return (

<RouterProvider router={root}/>

);

}

export default App;

===========================================================

\src\App.test.js

===========================================================

import { render, screen } from '@testing-library/react';

import App from './App';

test('renders learn react link', () => {

render(<App />);

const linkElement = screen.getByText(/learn react/i);

expect(linkElement).toBeInTheDocument();

});

===========================================================

\src\components\common\PageComponent.js

===========================================================

const PageComponent = ({serverData, movePage}) => {

return (

<div className="m-6 flex justify-center">

{serverData.prev ?

<div

className="m-2 p-2 w-16 text-center font-bold text-blue-400 "

onClick={() => movePage({page:serverData.prevPage} )}>

Prev </div> : <></>}

{serverData.pageNumList.map(pageNum =>

<div

key={pageNum}

className={ `m-2 p-2 w-12 text-center rounded shadow-md text-white ${serverData.current === pageNum? 'bg-gray-500':'bg-blue-400'}`}

onClick={() => movePage( {page:pageNum})}>

{pageNum}

</div>

)}

{serverData.next ?

<div

className="m-2 p-2 w-16 text-center font-bold text-blue-400"

onClick={() => movePage( {page:serverData.nextPage})}>

Next

</div> : <></>}

</div>

);

}

export default PageComponent;

===========================================================

\src\components\common\ResultModal.js

===========================================================

const ResultModal = ( {title,content, callbackFn} ) => {

return (

<div

className={`fixed top-0 left-0 z-[1055] flex h-full w-full justify-center bg-black bg-opacity-20`}

onClick={() => {

if(callbackFn) {

callbackFn()

}

}}>

<div

className="absolute bg-white shadow dark:bg-gray-700 opacity-100 w-1/4 rounded mt-10 mb-10 px-6 min-w-[600px]">

<div className="justify-center bg-warning-400 mt-6 mb-6 text-2xl border-b-4 border-gray-500">

{title}

</div>

<div className="text-4xl border-orange-400 border-b-4 pt-4 pb-4">

{content}

</div>

<div className="justify-end flex ">

<button

className="rounded bg-blue-500 mt-4 mb-4 px-6 pt-4 pb-4 text-lg text-white"

onClick={() => {

if(callbackFn) {

callbackFn()

}

}}>Close Modal</button>

</div>

</div>

</div>

);

}

export default ResultModal;

===========================================================

\src\components\menus\BasicMenu.js

===========================================================

import { Link } from "react-router-dom";

const BasicMenu = () => {

return (

<nav id='navbar' className=" flex bg-blue-300">

<div className="w-4/5 bg-gray-500" >

<ul className="flex p-4 text-white font-bold">

<li className="pr-6 text-2xl">

<Link to={'/'}>Main</Link>

</li>

<li className="pr-6 text-2xl">

<Link to={'/about'}>About</Link>

</li>

<li className="pr-6 text-2xl">

<Link to={'/todo/'}>Todo</Link>

</li>

</ul>

</div>

<div className="w-1/5 flex justify-end bg-orange-300 p-4 font-medium">

<div className="text-white text-sm m-1 rounded" >

Login

</div>

</div>

</nav>

);

}

export default BasicMenu;

===========================================================

\src\components\todo\AddComponent.js

===========================================================

import { useState } from "react";

import { postAdd } from "../../api/todoApi";

import ResultModal from "../common/ResultModal";

import useCustomMove from "../../hooks/useCustomMove";

const initState = {

title:'',

writer: '',

dueDate: ''

}

const AddComponent = () => {

const [todo, setTodo] = useState({...initState})

const [result, setResult] = useState(null) //결과 상태

const {moveToList} = useCustomMove() //useCustomMove 활용

const handleChangeTodo = (e) => {

todo[e.target.name] = e.target.value

setTodo({...todo})

}

const handleClickAdd = () => {

//console.log(todo)

postAdd(todo)

.then(result => {

console.log(result)

setResult(result.TNO) //결과 데이터 변경

setTodo({...initState})

}).catch(e => {

console.error(e)

})

}

const closeModal = () => {

setResult(null)

moveToList() //moveToList( )호출

}

return (

<div className = "border-2 border-sky-200 mt-10 m-2 p-4">

{/\* 모달 처리 \*/}

{result ? <ResultModal title={'Add Result'} content={`New ${result} Added`} callbackFn={closeModal}/>: <></>}

<div className="flex justify-center">

<div className="relative mb-4 flex w-full flex-wrap items-stretch">

<div className="w-1/5 p-6 text-right font-bold">TITLE</div>

<input className="w-4/5 p-6 rounded-r border border-solid border-neutral-500 shadow-md"

name="title"

type={'text'}

value={todo.title}

onChange={handleChangeTodo}

>

</input>

</div>

</div>

<div className="flex justify-center">

<div className="relative mb-4 flex w-full flex-wrap items-stretch">

<div className="w-1/5 p-6 text-right font-bold">WRITER</div>

<input className="w-4/5 p-6 rounded-r border border-solid border-neutral-500 shadow-md"

name="writer"

type={'text'}

value={todo.writer}

onChange={handleChangeTodo}

>

</input>

</div>

</div>

<div className="flex justify-center">

<div className="relative mb-4 flex w-full flex-wrap items-stretch">

<div className="w-1/5 p-6 text-right font-bold">DUEDATE</div>

<input className="w-4/5 p-6 rounded-r border border-solid border-neutral-500 shadow-md"

name="dueDate"

type={'date'}

value={todo.dueDate}

onChange={handleChangeTodo}

>

</input>

</div>

</div>

<div className="flex justify-end">

<div className="relative mb-4 flex p-4 flex-wrap items-stretch">

<button type="button"

className="rounded p-4 w-36 bg-blue-500 text-xl text-white "

onClick={handleClickAdd}

>

ADD

</button>

</div>

</div>

</div>

);

}

export default AddComponent;

===========================================================

\src\components\todo\ListComponent.js

===========================================================

import { useEffect, useState } from "react";

import { getList } from "../../api/todoApi";

import useCustomMove from "../../hooks/useCustomMove";

import PageComponent from "../common/PageComponent";

const initState = {

dtoList:[],

pageNumList:[],

pageRequestDTO: null,

prev: false,

next: false,

totoalCount: 0,

prevPage: 0,

nextPage: 0,

totalPage: 0,

current: 0

}

const ListComponent = () => {

const {page, size, refresh, moveToList, moveToRead} = useCustomMove()//refresh

//serverData는 나중에 사용

const [serverData, setServerData] = useState(initState)

useEffect(() => {

getList({page,size}).then(data => {

console.log(data)

setServerData(data)

})

}, [page,size, refresh])

return (

<div className="border-2 border-blue-100 mt-10 mr-2 ml-2">

<div className="flex flex-wrap mx-auto justify-center p-6">

{serverData.dtoList.map(todo =>

<div

key= {todo.tno}

className="w-full min-w-[400px] p-2 m-2 rounded shadow-md"

onClick={() => moveToRead(todo.tno)} //이벤트 처리 추가

>

<div className="flex ">

<div className="font-extrabold text-2xl p-2 w-1/12">

{todo.tno}

</div>

<div className="text-1xl m-1 p-2 w-8/12 font-extrabold">

{todo.title}

</div>

<div className="text-1xl m-1 p-2 w-2/10 font-medium">

{todo.dueDate}

</div>

</div>

</div>

)}

</div>

<PageComponent serverData={serverData} movePage={moveToList}></PageComponent>

</div>

);

}

export default ListComponent;

===========================================================

\src\components\todo\ModifyComponent.js

===========================================================

import { useCallback, useEffect, useState } from "react";

import { deleteOne, getOne, putOne } from "../../api/todoApi";

import ResultModal from "../common/ResultModal";

import useCustomMove from "../../hooks/useCustomMove";

const initState = {

tno:0,

title:'',

writer: '',

dueDate: '',

complete: false

}

const ModifyComponent = ({tno, moveList, moveRead}) => {

const [todo, setTodo] = useState({...initState})

//모달 창을 위한 상태

const [result, setResult] = useState(null)

//이동을 위한 기능들

const {moveToList, moveToRead} = useCustomMove()

const handleClickModify = () => { //버튼 클릭시

//console.log(todo)

putOne(todo).then(data => {

console.log("modify result: " + data)

setResult('Modified')

})

}

const handleClickDelete = () => { //버튼 클릭시

deleteOne(tno).then( data => {

console.log("delete result: " + data)

setResult('Deleted')

})

}

//모달 창이 close될때

const closeModal = () => {

if(result ==='Deleted') {

moveToList()

}else {

moveToRead(tno)

}

}

useEffect(() => {

getOne(tno).then(data => setTodo(data))

},[tno])

const handleChangeTodo = (e) => {

todo[e.target.name] = e.target.value

setTodo({...todo})

}

const handleChangeTodoComplete = (e) => {

const value = e.target.value

todo.complete = (value === 'Y')

setTodo({...todo})

}

return (

<div className = "border-2 border-sky-200 mt-10 m-2 p-4">

{result ? <ResultModal title={'처리결과'} content={result} callbackFn={closeModal}></ResultModal> :<></>}

<div className="flex justify-center mt-10">

<div className="relative mb-4 flex w-full flex-wrap items-stretch">

<div className="w-1/5 p-6 text-right font-bold">TNO</div>

<div className="w-4/5 p-6 rounded-r border border-solid shadow-md bg-gray-100">

{todo.tno}

</div>

</div>

</div>

<div className="flex justify-center">

<div className="relative mb-4 flex w-full flex-wrap items-stretch">

<div className="w-1/5 p-6 text-right font-bold">WRITER</div>

<div className="w-4/5 p-6 rounded-r border border-solid shadow-md bg-gray-100">

{todo.writer}

</div>

</div>

</div>

<div className="flex justify-center">

<div className="relative mb-4 flex w-full flex-wrap items-stretch">

<div className="w-1/5 p-6 text-right font-bold">TITLE</div>

<input className="w-4/5 p-6 rounded-r border border-solid border-neutral-300 shadow-md"

name="title"

type={'text'}

value={todo.title}

onChange={handleChangeTodo}

>

</input>

</div>

</div>

<div className="flex justify-center">

<div className="relative mb-4 flex w-full flex-wrap items-stretch">

<div className="w-1/5 p-6 text-right font-bold">DUEDATE</div>

<input className="w-4/5 p-6 rounded-r border border-solid border-neutral-300 shadow-md"

name="dueDate"

type={'date'}

value={todo.dueDate}

onChange={handleChangeTodo}

>

</input>

</div>

</div>

<div className="flex justify-center">

<div className="relative mb-4 flex w-full flex-wrap items-stretch">

<div className="w-1/5 p-6 text-right font-bold">COMPLETE</div>

<select

name="status"

className="border-solid border-2 rounded m-1 p-2"

onChange={handleChangeTodoComplete}

value = {todo.complete? 'Y':'N'} >

<option value='Y'>Completed</option>

<option value='N'>Not Yet</option>

</select>

</div>

</div>

<div className="flex justify-end p-4">

<button type="button"

className="inline-block rounded p-4 m-2 text-xl w-32 text-white bg-red-500"

onClick={handleClickDelete}

>

Delete

</button>

<button type="button"

className="rounded p-4 m-2 text-xl w-32 text-white bg-blue-500"

onClick={handleClickModify}

>

Modify

</button>

</div>

</div>

);

}

export default ModifyComponent;

===========================================================

\src\components\todo\ReadComponent.js

===========================================================

import { useEffect, useState } from "react"

import {getOne} from "../../api/todoApi"

import useCustomMove from "../../hooks/useCustomMove"

const initState = {

tno:0,

title:'',

writer: '',

dueDate: null,

complete: false

}

const ReadComponent = ({tno}) => {

const [todo, setTodo] = useState(initState) //아직 todo는 사용하지 않음

const {moveToList, moveToModify} = useCustomMove()

useEffect(() => {

getOne(tno).then(data => {

console.log(data)

setTodo(data)

})

}, [tno])

return (

<div className = "border-2 border-sky-200 mt-10 m-2 p-4 ">

{makeDiv('Tno', todo.tno)}

{makeDiv('Writer', todo.writer)}

{makeDiv('Title', todo.title)}

{makeDiv('Due Date', todo.dueDate)}

{makeDiv('Complete', todo.complete ? 'Completed' : 'Not Yet')}

{/\* buttons.........start \*/}

<div className="flex justify-end p-4">

<button type="button"

className="rounded p-4 m-2 text-xl w-32 text-white bg-blue-500"

onClick={() => moveToList()}

>

List

</button>

<button type="button"

className="rounded p-4 m-2 text-xl w-32 text-white bg-red-500"

onClick={() => moveToModify(tno)}

>

Modify

</button>

</div>

</div>

)

}

const makeDiv = (title,value) =>

<div className="flex justify-center">

<div className="relative mb-4 flex w-full flex-wrap items-stretch">

<div className="w-1/5 p-6 text-right font-bold">{title}</div>

<div className="w-4/5 p-6 rounded-r border border-solid shadow-md">

{value}

</div>

</div>

</div>

export default ReadComponent

===========================================================

\src\hooks\useCustomMove.js

===========================================================

import { useState } from "react"

import { createSearchParams, useNavigate, useSearchParams } from "react-router-dom"

const getNum = (param, defaultValue) => {

if(!param){

return defaultValue

}

return parseInt(param)

}

const useCustomMove = () => {

const navigate = useNavigate()

const [refresh, setRefresh] = useState(false)

const [queryParams] = useSearchParams()

const page = getNum(queryParams.get('page'), 1)

const size = getNum(queryParams.get('size'),10)

const queryDefault = createSearchParams({page, size}).toString() //새로 추가

const moveToList = (pageParam) => {

let queryStr = ""

if(pageParam){

const pageNum = getNum(pageParam.page, 1)

const sizeNum = getNum(pageParam.size, 10)

queryStr = createSearchParams({page:pageNum, size: sizeNum}).toString()

}else {

queryStr = queryDefault

}

navigate({

pathname: `../list`,

search:queryStr

})

setRefresh(!refresh) //추가

}

const moveToModify = (num) => {

console.log(queryDefault)

navigate({

pathname: `../modify/${num}`,

search: queryDefault //수정시에 기존의 쿼리 스트링 유지를 위해

})

}

const moveToRead =(num) => {

console.log(queryDefault)

navigate({

pathname: `../read/${num}`,

search: queryDefault

})

}

return {moveToList, moveToModify, moveToRead, page, size, refresh} //refresh 추가

}

export default useCustomMove

===========================================================

\src\index.css

===========================================================

@tailwind base;

@tailwind components;

@tailwind utilities;

===========================================================

\src\index.js

===========================================================

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<App />

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

===========================================================

\src\layouts\BasicLayout.js

===========================================================

import BasicMenu from "../components/menus/BasicMenu";

const BasicLayout = ({children}) => {

return (

<>

{/\* 기존 헤더 대신 BasicMenu\*/ }

<BasicMenu/>

{/\* 상단 여백 my-5 제거 \*/}

<div

className="bg-white my-5 w-full flex flex-col space-y-1 md:flex-row md:space-x-1 md:space-y-0">

<main

className="bg-sky-300 md:w-4/5 lg:w-3/4 px-5 py-5"> {/\* 상단 여백 py-40 변경 flex 제거 \*/}

{children}

</main>

<aside

className="bg-green-300 md:w-1/5 lg:w-1/4 px-5 flex py-5"> {/\* 상단 여백 py-40 제거 flex 제거 \*/}

<h1 className="text-2xl md:text-4xl">Sidebar</h1>

</aside>

</div>

</>

);

}

export default BasicLayout;

===========================================================

\src\logo.svg

===========================================================

<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 841.9 595.3"><g fill="#61DAFB"><path d="M666.3 296.5c0-32.5-40.7-63.3-103.1-82.4 14.4-63.6 8-114.2-20.2-130.4-6.5-3.8-14.1-5.6-22.4-5.6v22.3c4.6 0 8.3.9 11.4 2.6 13.6 7.8 19.5 37.5 14.9 75.7-1.1 9.4-2.9 19.3-5.1 29.4-19.6-4.8-41-8.5-63.5-10.9-13.5-18.5-27.5-35.3-41.6-50 32.6-30.3 63.2-46.9 84-46.9V78c-27.5 0-63.5 19.6-99.9 53.6-36.4-33.8-72.4-53.2-99.9-53.2v22.3c20.7 0 51.4 16.5 84 46.6-14 14.7-28 31.4-41.3 49.9-22.6 2.4-44 6.1-63.6 11-2.3-10-4-19.7-5.2-29-4.7-38.2 1.1-67.9 14.6-75.8 3-1.8 6.9-2.6 11.5-2.6V78.5c-8.4 0-16 1.8-22.6 5.6-28.1 16.2-34.4 66.7-19.9 130.1-62.2 19.2-102.7 49.9-102.7 82.3 0 32.5 40.7 63.3 103.1 82.4-14.4 63.6-8 114.2 20.2 130.4 6.5 3.8 14.1 5.6 22.5 5.6 27.5 0 63.5-19.6 99.9-53.6 36.4 33.8 72.4 53.2 99.9 53.2 8.4 0 16-1.8 22.6-5.6 28.1-16.2 34.4-66.7 19.9-130.1 62-19.1 102.5-49.9 102.5-82.3zm-130.2-66.7c-3.7 12.9-8.3 26.2-13.5 39.5-4.1-8-8.4-16-13.1-24-4.6-8-9.5-15.8-14.4-23.4 14.2 2.1 27.9 4.7 41 7.9zm-45.8 106.5c-7.8 13.5-15.8 26.3-24.1 38.2-14.9 1.3-30 2-45.2 2-15.1 0-30.2-.7-45-1.9-8.3-11.9-16.4-24.6-24.2-38-7.6-13.1-14.5-26.4-20.8-39.8 6.2-13.4 13.2-26.8 20.7-39.9 7.8-13.5 15.8-26.3 24.1-38.2 14.9-1.3 30-2 45.2-2 15.1 0 30.2.7 45 1.9 8.3 11.9 16.4 24.6 24.2 38 7.6 13.1 14.5 26.4 20.8 39.8-6.3 13.4-13.2 26.8-20.7 39.9zm32.3-13c5.4 13.4 10 26.8 13.8 39.8-13.1 3.2-26.9 5.9-41.2 8 4.9-7.7 9.8-15.6 14.4-23.7 4.6-8 8.9-16.1 13-24.1zM421.2 430c-9.3-9.6-18.6-20.3-27.8-32 9 .4 18.2.7 27.5.7 9.4 0 18.7-.2 27.8-.7-9 11.7-18.3 22.4-27.5 32zm-74.4-58.9c-14.2-2.1-27.9-4.7-41-7.9 3.7-12.9 8.3-26.2 13.5-39.5 4.1 8 8.4 16 13.1 24 4.7 8 9.5 15.8 14.4 23.4zM420.7 163c9.3 9.6 18.6 20.3 27.8 32-9-.4-18.2-.7-27.5-.7-9.4 0-18.7.2-27.8.7 9-11.7 18.3-22.4 27.5-32zm-74 58.9c-4.9 7.7-9.8 15.6-14.4 23.7-4.6 8-8.9 16-13 24-5.4-13.4-10-26.8-13.8-39.8 13.1-3.1 26.9-5.8 41.2-7.9zm-90.5 125.2c-35.4-15.1-58.3-34.9-58.3-50.6 0-15.7 22.9-35.6 58.3-50.6 8.6-3.7 18-7 27.7-10.1 5.7 19.6 13.2 40 22.5 60.9-9.2 20.8-16.6 41.1-22.2 60.6-9.9-3.1-19.3-6.5-28-10.2zM310 490c-13.6-7.8-19.5-37.5-14.9-75.7 1.1-9.4 2.9-19.3 5.1-29.4 19.6 4.8 41 8.5 63.5 10.9 13.5 18.5 27.5 35.3 41.6 50-32.6 30.3-63.2 46.9-84 46.9-4.5-.1-8.3-1-11.3-2.7zm237.2-76.2c4.7 38.2-1.1 67.9-14.6 75.8-3 1.8-6.9 2.6-11.5 2.6-20.7 0-51.4-16.5-84-46.6 14-14.7 28-31.4 41.3-49.9 22.6-2.4 44-6.1 63.6-11 2.3 10.1 4.1 19.8 5.2 29.1zm38.5-66.7c-8.6 3.7-18 7-27.7 10.1-5.7-19.6-13.2-40-22.5-60.9 9.2-20.8 16.6-41.1 22.2-60.6 9.9 3.1 19.3 6.5 28.1 10.2 35.4 15.1 58.3 34.9 58.3 50.6-.1 15.7-23 35.6-58.4 50.6zM320.8 78.4z"/><circle cx="420.9" cy="296.5" r="45.7"/><path d="M520.5 78.1z"/></g></svg>

===========================================================

\src\pages\AboutPage.js

===========================================================

import BasicLayout from "../layouts/BasicLayout";

const AboutPage = () => {

return (

<BasicLayout>

<div className=" text-3xl">About Page</div>

</BasicLayout>

);

}

export default AboutPage;

===========================================================

\src\pages\MainPage.js

===========================================================

import BasicLayout from "../layouts/BasicLayout";

const MainPage = () => {

return (

<BasicLayout>

<div className=" text-3xl">Main Page</div>

</BasicLayout>

);

}

export default MainPage;

===========================================================

\src\pages\todo\AddPage.js

===========================================================

import AddComponent from "../../components/todo/AddComponent";

const AddPage = () => {

return (

<div className="p-4 w-full bg-white">

<div className="text-3xl font-extrabold">

Todo Add Page

</div>

<AddComponent/>

</div>

);

}

export default AddPage;

===========================================================

\src\pages\todo\IndexPage.js

===========================================================

import { Outlet, useNavigate } from "react-router-dom";

import BasicLayout from "../../layouts/BasicLayout";

import { useCallback } from "react";

const IndexPage = () => {

const navigate = useNavigate()

const handleClickList = useCallback(() => {

navigate({ pathname:'list' })

})

const handleClickAdd = useCallback(() => {

navigate({ pathname:'add' })

})

return (

<BasicLayout>

<div className="w-full flex m-2 p-2 ">

<div

className="text-xl m-1 p-2 w-20 font-extrabold text-center underline"

onClick={handleClickList}>

LIST

</div>

<div

className="text-xl m-1 p-2 w-20 font-extrabold text-center underline"

onClick={handleClickAdd}>

ADD

</div>

</div>

<div className="flex flex-wrap w-full">

<Outlet/>

</div>

</BasicLayout>

);

}

export default IndexPage;

===========================================================

\src\pages\todo\ListPage.js

===========================================================

import ListComponent from "../../components/todo/ListComponent";

const ListPage = () => {

return (

<div className="p-4 w-full bg-white">

<div className="text-3xl font-extrabold">

Todo List Page Component

</div>

<ListComponent/>

</div>

);

}

export default ListPage;

===========================================================

\src\pages\todo\ModifyPage.js

===========================================================

import { useParams } from "react-router-dom";

import ModifyComponent from "../../components/todo/ModifyComponent";

const ModifyPage = () => {

const {tno} = useParams()

return (

<div className="p-4 w-full bg-white">

<div className="text-3xl font-extrabold">

Todo Modify Page

</div>

<ModifyComponent tno={tno}/>

</div>

);

}

export default ModifyPage;

===========================================================

\src\pages\todo\ReadPage.js

===========================================================

import { useParams } from "react-router-dom";

import ReadComponent from "../../components/todo/ReadComponent";

const ReadPage = () => {

const {tno} = useParams()

return (

<div className="font-extrabold w-full bg-white mt-6">

<div className="text-2xl ">

Todo Read Page Component {tno}

</div>

<ReadComponent tno={tno}></ReadComponent>

</div>

);

}

export default ReadPage;

===========================================================

\src\reportWebVitals.js

===========================================================

const reportWebVitals = onPerfEntry => {

if (onPerfEntry && onPerfEntry instanceof Function) {

import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {

getCLS(onPerfEntry);

getFID(onPerfEntry);

getFCP(onPerfEntry);

getLCP(onPerfEntry);

getTTFB(onPerfEntry);

});

}

};

export default reportWebVitals;

===========================================================

\src\router\root.js

===========================================================

import { Suspense, lazy } from "react";

import todoRouter from "./todoRouter";

const { createBrowserRouter } = require("react-router-dom");

const Loading = <div>Loading....</div>

const Main = lazy(() => import("../pages/MainPage"))

const About = lazy(() => import("../pages/AboutPage"))

const TodoIndex = lazy(() => import("../pages/todo/IndexPage"))

const TodoList = lazy(() => import("../pages/todo/ListPage"))

const root = createBrowserRouter([

{

path: "",

element: <Suspense fallback={Loading}><Main/></Suspense>

},

{

path: "about",

element: <Suspense fallback={Loading}><About/></Suspense>

},

{

path: "todo",

element: <Suspense fallback={Loading}><TodoIndex/></Suspense>,

children: todoRouter()

}

])

export default root;

===========================================================

\src\router\todoRouter.js

===========================================================

import { Suspense, lazy } from "react";

import { Navigate } from "react-router-dom";

const Loading = <div>Loading....</div>

const TodoList = lazy(() => import("../pages/todo/ListPage"))

const TodoRead = lazy(() => import("../pages/todo/ReadPage"))

const TodoAdd = lazy(() => import("../pages/todo/AddPage"))

const TodoModify = lazy(() => import("../pages/todo/ModifyPage"))

const todoRouter = () => {

return [

{

path: "list",

element: <Suspense fallback={Loading}><TodoList/></Suspense>

},

{

path: "",

element: <Navigate replace to="list"/>

},

{

path: "read/:tno",

element: <Suspense fallback={Loading}><TodoRead/></Suspense>

},

{

path: "add",

element: <Suspense fallback={Loading}><TodoAdd/></Suspense>

},

{

path: "modify/:tno",

element: <Suspense fallback={Loading}><TodoModify/></Suspense>

}

]

}

export default todoRouter;

===========================================================

\src\setupTests.js

===========================================================

// jest-dom adds custom jest matchers for asserting on DOM nodes.

// allows you to do things like:

// expect(element).toHaveTextContent(/react/i)

// learn more: https://github.com/testing-library/jest-dom

import '@testing-library/jest-dom';