這個proj主要在於學習react的routing

<https://www.pexels.com/zh-tw/api/>

登入之後臉選頭像，點選影像和影片API

他會叫你寫一些相關資訊，唬爛一下，取得API key如下

每個人的都不一樣

563492ad6f91700001000001d7b5ed3eacc94d4fbeff83c83ddee5a5

Doc在這裡

<https://www.pexels.com/zh-tw/api/documentation/>

* **React: route setting using react-router-dom**

接著要透過react-router-dom進行

CMD > npm inatall react-router-dom >

Index.js

*import* React *from* "react";

*import* ReactDOM *from* "react-dom";

*import* App *from* "./App";

*import* { BrowserRouter } *from* "react-router-dom"; //從套件中import BrowserRouter

ReactDOM.render(

  <React.StrictMode>

    <BrowserRouter> <--BrowserRouter寫成一個Tag，並且將Component包起來

      <App />

    </BrowserRouter>

  </React.StrictMode>,

  document.querySelector("#root")

);

App.js

*import* React *from* "react";

*import* Nav *from* "./components/Nav";

*import* Footer *from* "./components/Footer";

*import* Homepage *from* "./pages/Homepage";

*import* { Routes , Route } *from* "react-router-dom"; //從套件中import Routes, Route

const App = () => {

*return* (

    <div *className*="App">

      <Nav />

        <Routes>

        <Route *element*={<Homepage />/>

          <Route *element*={<About />/>

        </Routes>

      <Footer />

    </div>

  );

};

*export* *default* App;

React的所有網頁都是藉由.JS檔生成，預設會去讀取src資料夾中index.js

* **最基礎的React運作原理:**

1. **寫個function去生成html tag(createElement())**
2. **透過ReactDOM.render()指定呈現的位置**

* 引入Route之後: react-router-dom

1. 在最上層(index.js) 做render()

(裡面的Component:<React.strictMode>內要再加一層 <BrowserRouter>)

2. 內層(App.js) 透過JSX寫上整體架構(Nav, Routes, Footer) export給index.js用↑

3. 最內層的各個js檔都是作為reusuable component 只寫簡單的function export給App.js用↑

Index.js

*import* React *from* "react";

*import* ReactDOM *from* "react-dom";

*import* App *from* "./App";

*import* { BrowserRouter } *from* "react-router-dom";

ReactDOM.render(

  <React.StrictMode>

    <BrowserRouter>

      <App />

    </BrowserRouter>

  </React.StrictMode>,

  document.querySelector("#root")

);

App.js

*import* React *from* "react";

*import* Nav *from* "./components/Nav"; //將各個reusuable component引入app.js內

*import* Footer *from* "./components/Footer"; //將各個reusuable component引入app.js內

*import* Homepage *from* "./pages/Homepage"; //將各個reusuable component引入app.js內

*import* About *from* "./pages/About"; //將各個reusuable component引入app.js內

*import* { Routes, Route } *from* "react-router-dom"; //用來在React環境下設定route

*import* "./styles/style.css"; //引入css

*//設定Route的方法: 在<Route>裡面設定property path以及exact*

*(舊版react-router-dom設定一定要有exact才會正常顯示)*

*//注意: 這邊的Route是新版本的寫法 <Routes><Route path=""element={<component/>}></Routes>*

*//path 裡面assign你要設定的route名稱 (首頁就"/"; About分頁"/About")*

//【記得】React要藉function生成HTML Element 但還是受限於return只能傳一個物件所以要用<div>包

const App = () => {

*return* (

    <div *className*="App">

      <Nav />

      <Routes>

        <Route *path*="/" *element*={<Homepage />} />

        <Route *path*="/About" *element*={<About />} />

      </Routes>

      <Footer />

    </div>

  );

};

*export* *default* App;

Nav.js

*import* React *from* "react";

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

const Nav = () => {

*return* (

*//原本的<a href=""></a>在router中需改寫為<Link to=""></Link>*

    <nav>

      <ul>

        <li>

          <Link *to*="/">Home</Link>

        </li>

        <li>

          <Link *to*="/about">About</Link>

        </li>

      </ul>

    </nav>

  );

};

*export* *default* Nav;

* Footer.js

*import* React *from* "react";

const Footer = () => {

*return* <div *className*="footer">Johnny Lin 2022©</div>;

};

*export* *default* Footer;

* About.js

*import* React *from* "react";

const About = () => {

*return* (

    <div>

      <h1>About</h1>

    </div>

  );

};

*export* *default* About;

* Search.js

*import* React *from* "react";

const Search = ({ searchButtonHandler, setInput }) => {

//繼承從Homepage.js props繼承來的searchButtonHandler, setInput

  const inputHandler = (e) => {

    setInput(e.target.value);

//這段的意思是user <input>輸入的值只要變化，輸入的值就同步改變input這個state(Homepage.js)

//

  };

*return* (

    <div *className*="search">

      <input *onChange*={inputHandler} *type*="text" />

      <button *onClick*={searchButtonHandler}>Search</button>

//子元素繼承父元素的方法，所以這邊才可以使用searchButtonHandler去bind click event

    </div>

  );

};

*export* *default* Search;

* Picture.js

*import* React *from* "react";

const Picture = ({ data }) => { //這邊的data就是從homepage繼承來的API回傳資料

*return* ( //把API 的response.json()後就轉為object，用dot notation可以使用property

    <div *className*="picture">

      <p>{data.photographer}</p> //使用時記得還是要{}包覆

      <div *className*="imageContainer">

        <img *src*={data.src.large} *alt*="" />

      </div>

      <p>

        Download Image:{" "}

        <a *target*="\_blank" *href*={data.src.large}>

          Click Here

        </a>

      </p>

    </div>

  );

};

*export* *default* Picture;

* Homepage.js

*import* { React, useState, useEffect } *from* "react";

*import* Search *from* "../components/Search";

*import* Picture *from* "../components/Picture";

const Homepage = () => {

  const [input, setInput] = useState("");//這邊的state (input)會跟著search.js連動更新

  let [data, setData] = useState(null);

  const authkey = "563492ad6f91700001000001d7b5ed3eacc94d4fbeff83c83ddee5a5";

  const initialURL = "https://api.pexels.com/v1/curated?page=1&per\_page=20";

  const searchURL = `https://api.pexels.com/v1/search?query=${input}&page=1&per\_page=20`;

//上面的state (input)又被用來做為API endpoint使用(這個API用來搜尋圖片)

  let [currentSearch, setcurrentSearch] = useState("");

*//fetch data from pexels API*

  let searchButtonHandler = async (url) => { //這邊的searchButtonHandler在search.js實作

    setPage(2);

fetch()這個method一般就帶入API endpoint，其實也可以另外加入第二parameter

調整headers物件(ex:設定送過去的method(GET,POST,PUT/PATCH,DELETE), 設定authKEY【API doc沒特別寫的話都是GET request, auth key寫在options裡】)

    const dataFetch = *await* fetch(url, {

      method: "GET", //透過API告訴他我的request是怎麼樣的，這邊是請求資料的GET

      headers: { //包含在GET request一同送過去的相關資訊會放進headers

        Accept: "application/json", //強烈建議API你回傳的資料最好給我是JSON格式

(JSON 的官方定義 MIME 為 application/json)

        Authorization: authkey, //告訴API我的AUTH KEY是啥

      },

    });

    let parsedData = *await* dataFetch.json();

裡可以直接使用fetch API中內建的.json() method將json data

轉換成javascript object我們就能使用他

    setData(parsedData.photos);

*//把回傳的parsedData打開來看會發現我們要的照片都在photo這個property。*

  };

*// fetch data when the page loads up*

  useEffect(() => {

*//注意:這邊的不用放return*

    searchButtonHandler(initialURL);

  }, []); *//第二個代表你要讓哪個state改變時被觸發，如果都沒放就是任何改變都觸發*

  useEffect(() => {

*if* (currentSearch === "") {

      searchButtonHandler(initialURL);

    } *else* {

      searchButtonHandler(searchURL);

    }

  }, [currentSearch]);

  let [page, setPage] = useState(1);

  let loadMorePicture = async () => {

    let newURL;

*if* (input === "") {

      newURL = `https://api.pexels.com/v1/curated?page=${page}&per\_page=20`;

    } *else* {

      newURL = `https://api.pexels.com/v1/search?query=${input}&page=${page}&per\_page=20`;

    }

    setPage(page + 1);

    const dataFetch = *await* fetch(newURL, {

      method: "GET",

      headers: {

        Accept: "application/json",

        Authorization: authkey,

      },

    });

    let parsedData = *await* dataFetch.json();

    setData(data.concat(parsedData.photos));

  };

*return* (

    <div>

      <Search

*searchButtonHandler*={() => {

*//js閉包*

          setcurrentSearch(input);

*// searchButtonHandler(searchURL);*

        }}

*setInput*={setInput}

      />

      <div *className*="pictures">

        {data &&

          data.map((d) => {

*return* <Picture *data*={d} />;

          })}

      </div>

      <div *className*="morePicture">

        <button *onClick*={loadMorePicture}>Load more</button>

      </div>

    </div>

  );

};

*export* *default* Homepage;