

Midterm Preparation 期中測驗準備

這份文件是為了給自己做複習，順便分享給你做參考。有問題可以一起討論！

本期目標參考老師於Iclass上的敘述所使用，有可能與實際Code不符。還請諒解哈！

Github Repo : https://github.com/marx-w/1121-WP1_midterm.git

Midterm Preparation 期中測驗準備	1
Requirement	1
後端設計	2
Postgres App使用	2
Express JS的路由	4
前端設計	7
Route的使用	7
API fetch跟State的用法	8

Requirement

為了協助同學們整合W1~W7上課所學，以便準備W10考試，老師準備在 Week 8 給同學們一次期中模擬考，預定

18:10 下載題目，老師會提供相關上課的 code

18:20 講解題目後斷網

可以問助教問題，但同學間不可以交談。

同學們必須到學校機房考試，繳交每張圖片都必須要有機房左側icon截圖，如 W7上課老師所說。

考試重點放在

PostgreSQL 後端：

- 要熟悉pgAdmin 使用
- 要能建立資料庫
- 要能透過 SQL 建立資料表
- 要能透過 SQL 建立相關資料
- 要能透過 api 呼叫取得 json資料

React 前端：

- 要能將 html theme，透過路由呈現靜態頁面
- 要能將 api 呼叫取得的json資料轉成json陣列
- 要能透過路由，自 json 陣列取得資料來呈現
- 要能透過路由，直接透過 api呼叫，從node server 端取得json資料並呈現在前端

思路

先創立資料庫，設置好後使用ExpressJS與資料庫進行對接。

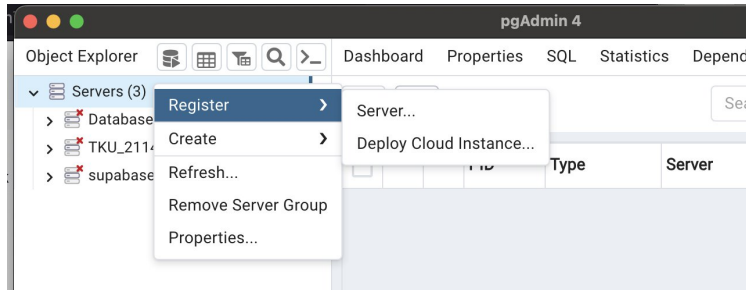
完成測試後，架設React前端畫面。

想法有了，咱們開始吧！

後端設計

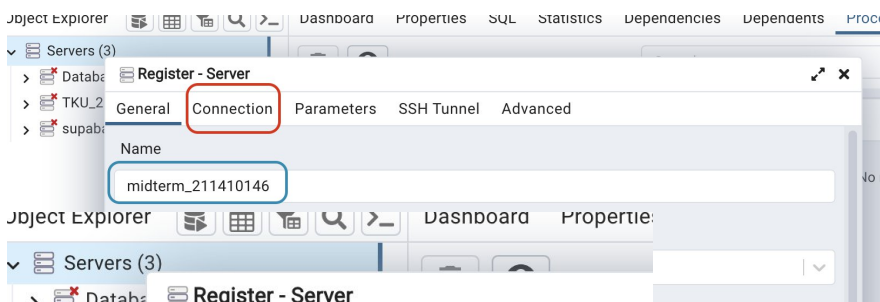
Postgres App使用

請注意，我的版本是使用Mac作為開發環境，可能創建過程會略為不同。但理論上應該相同。如果有問題直接跟我說，我再補一版Windows。



這是PG Admin，也是一個GUI介面操作Postgres的資料庫。我們就從這裡開始吧！

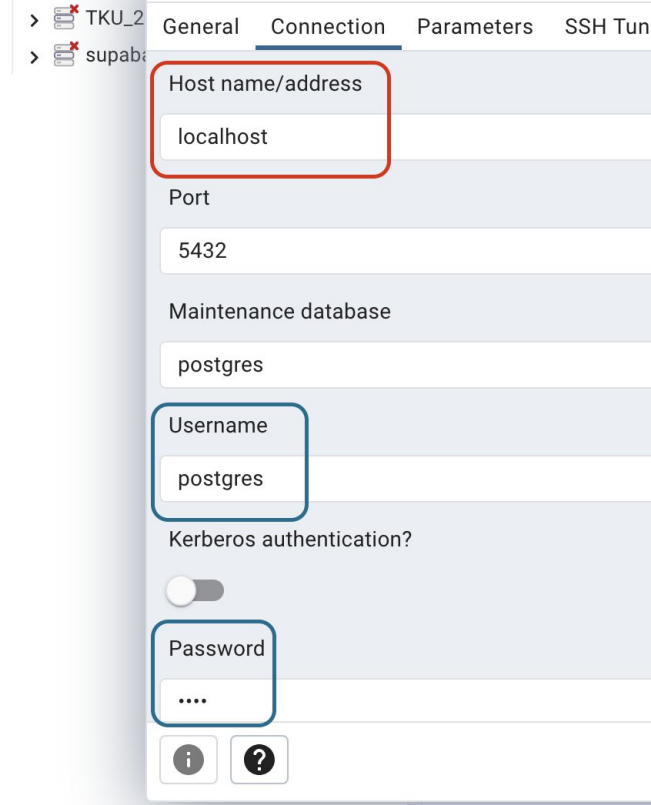
首先對著Servers按右鍵，進入Register，進入Server。



進來後先取名，通常老師會使用學號作為區隔，請注意老師要求。

然後點進去Connection。

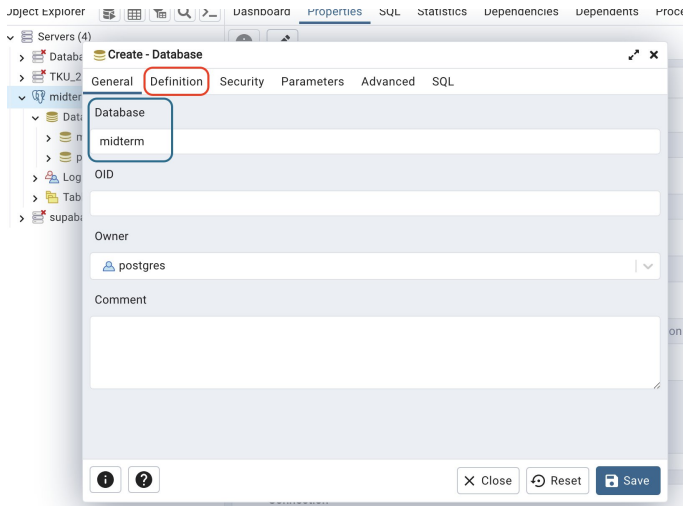
點進來會看見一份琳琅滿目，主要需要關注的在這框起來的就好。



Host Name那輸入localhost或127.0.0.1。無線網路老師告訴我們那是代表本機的意思，意味著我們要在這台電腦上設置Postgres Database。

Username的部分請記好，Express的時候有大用；Password我忘了他會不會預設，如果沒有那就好，如果有請記得密碼。我目前是隨便設0000。

緊接著就來到創建Database跟Table了。



右鍵點你剛才創立好的Server，然後選擇Create，選擇Database。名字設好理論上就ok；如果遇到一個Local Provider的問題，點選Definition，選擇Local Provider後換一個即可。

快速進入Table的部分。

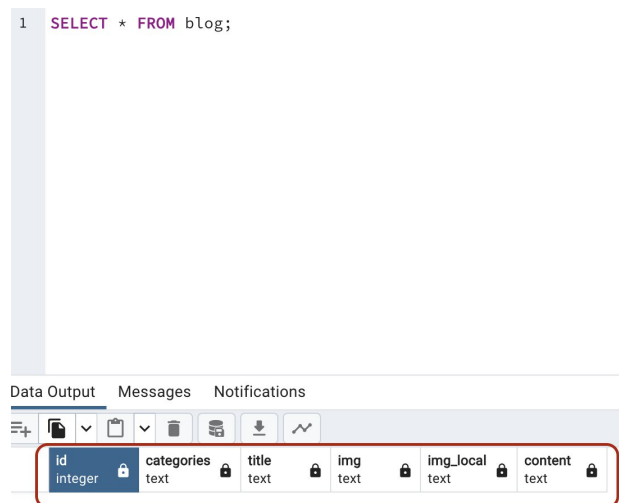
在創立的Database那右鍵，選取Query tool。

由於我預先使用CoPilot產生了好幾筆測資，格式也仿造老師的格式。

```
```SQL
CREATE TABLE blog (
 id int,
 categories text,
 title text,
 img text,
 img_local text,
 content text
);
```

```JSON
{
 "id": 0,
 "categories": "",
 "title": "",
 "img": "",
 "img_local": "",
 "content": ""
}
```
```

我把上面的SQL Command輸入進去Query tool內，建立了一個Schema。我懶得算字數，所以不選擇varchar或char，你可以有其他選擇。



接下來用SQL指令輸入剛才產生的測資。

```
```SQL
INSERT INTO blog (id, categories, title, img, img_local, content) VALUES
(1, 'Life Style', '5 Tips for a Healthier Lifestyle', 'https://unsplash.com/photos/woman-doing-yoga-meditation-on-brown-parquet-flooring-NTyBbu66_SI', './DataImages/img-01.jpg', 'Drink more water, eat more fruits and vegetables, exercise regularly, get enough sleep, and reduce stress. '),
(2, 'Tech', 'The Future of Artificial Intelligence', 'https://unsplash.com/photos/vintage-gray-game-console-and-joystick-p0j-mE6mGo4', './DataImages/img-02.jpg', 'Artificial intelligence will continue to advance and become more integrated into our daily lives, from self-driving cars to personalized healthcare. ');
```
```

這裡只將幾筆資料塞進去，剩下的我也開了一個Markdown，給你參考一下！

1

SELECT * FROM blog;

Data Output

Messages

Notifications

| | id
integer | categories
text | title
text | img
text |
|----|---------------|--------------------|--|-------------------------|
| 1 | 1 | Life Style | 5 Tips for a Healthier Lifestyle | https://unsplash.com/ph |
| 2 | 2 | Tech | The Future of Artificial Intelligence | https://unsplash.com/ph |
| 3 | 3 | Self Improvement | 5 Habits of Highly Successful People | https://unsplash.com/ph |
| 4 | 4 | News | COVID-19 Vaccines: What You Need to ... | https://unsplash.com/ph |
| 5 | 5 | Life Style | How to Stay Motivated to Exercise | https://unsplash.com/ph |
| 6 | 6 | Tech | The Benefits of Cloud Computing | https://unsplash.com/ph |
| 7 | 7 | Self Improvement | The Power of Positive Thinking | https://unsplash.com/ph |
| 8 | 8 | News | Climate Change: What You Need to Know | https://unsplash.com/ph |
| 9 | 9 | Life Style | Healthy Eating on a Budget | https://unsplash.com/ph |
| 10 | 10 | Tech | The Rise of 5G Technology | https://unsplash.com/ph |
| 11 | 11 | Self Improvement | The Importance of Time Management | https://unsplash.com/ph |
| 12 | 12 | News | The Future of Work: Remote vs. In-Person | https://unsplash.com/ph |

內容塞進去了、資料庫健全了。接下來該著手進行Express的部分了。

Express JS的路由

在Express JS，我們要建立起路由，讓我們能有一個安全的方式對資料庫進行操作。

首先進到終端機輸入

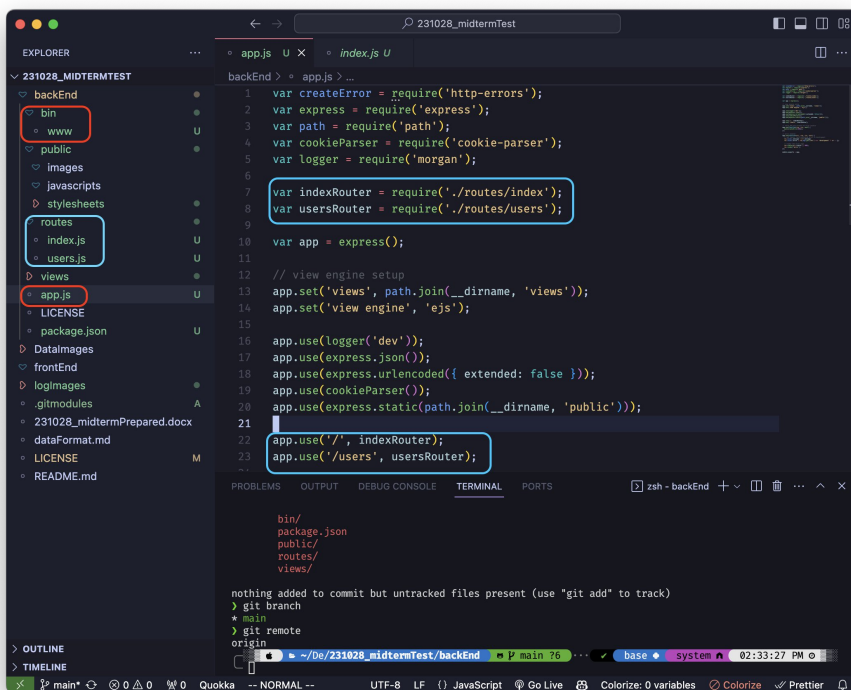
```
npm install -g express-generator
```

這是老師課上使用的方式，用來幫你設訂預設的程式框架出來。我們接下來輸入

```
express --view=ejs backEnd
```

backEnd 是我要將後端程式放置的位置，可以替換成其他路徑（如當前資料夾．）。

然後我們就得到以下內容。



紅色框是掌管程式運行的地方，我們不太需要動。

藍色框則是要關注的地方。Routes資料夾內的檔案是express公司為了方便你管理而放置的地方，但他沒有特別功能。App.js內的藍色框則是讓我們能在輸入網址時進行溝通的橋樑。如我們輸入localhost/他會回傳indexRouter裡面的內容。

我們接下來要做三個部分：創建新檔案、模組導入app.js、介接資料庫。

我在route資料夾下創立了一個card.js檔，並且在根目錄（最外層）的app.js中加了這段。

```
app.use('/cards', cardRouter);
```

我要求程式當伺服器接收到關於localhost/cards請求時，會執行cardRouter中（包含底下分支）的所有動作。所以如果輸入localhost/cards/title/MarxIsHandsome，那一樣是歸類在cardRouter的業務範疇內。

```
/* GET all cards. */  
  
router.get('/', function(req, res, next) {  
  res.json({"card": "This is a card."});  
});
```

來到card.js中，我先寫了一段Code來測試程式運行正常。router擔任起這個頁面底下分類、回傳的動作。也是回傳的內容。（有回傳才會動作！）

作！）

router.get（紅色箭頭）意思是獲取這個位置的內容（詳情請見[這裡](#)），請求對象是“/”這個地方。我們目標是在這個位置中將所有的卡片以JSON格式給展示出來，因此我們在Response的地方寫了以JSON方式回傳（藍色箭頭）。

既然我們已經將測資準備好，現在就來整進JavaScript吧！

```
const { Pool } = require('pg');  
  
let pool;  
  
pool = new Pool({  
  host: 'localhost',  
  port: 5432,  
  user: 'postgres',  
  password: '0000',  
  database: 'midterm',  
});  
  
module.exports = pool;
```

我在根目錄中創建一個Utils的目錄，並且開了一個新檔案將左邊的Code丟進去。下面的module.exports將Pool導出去，若是任何檔案需要操作這個資料庫，就直接使用這個變數就可以了。

藍色箭頭的部分需要注意一下，這是javascript的舊有語法。由於Javascript的語法整天推陳出新，因此框架使用的語法需要注意！（Debug半小時得人類）


```
var express = require('express');
var router = express.Router();
var database = require('../utils/connection');

router.get('/', async function(req, res, next) {
  const result = await database.query('SELECT * FROM blog');
  console.log(result.rows);
  res.json(result.rows);
});

router.get('/:id', async function(req, res){
  const reqId = parseInt(req.params.id);
  const result = await database.query('SELECT * FROM blog WHERE id = $1',
    [reqId]);
  res.json(result.rows);
});

module.exports = router;
```

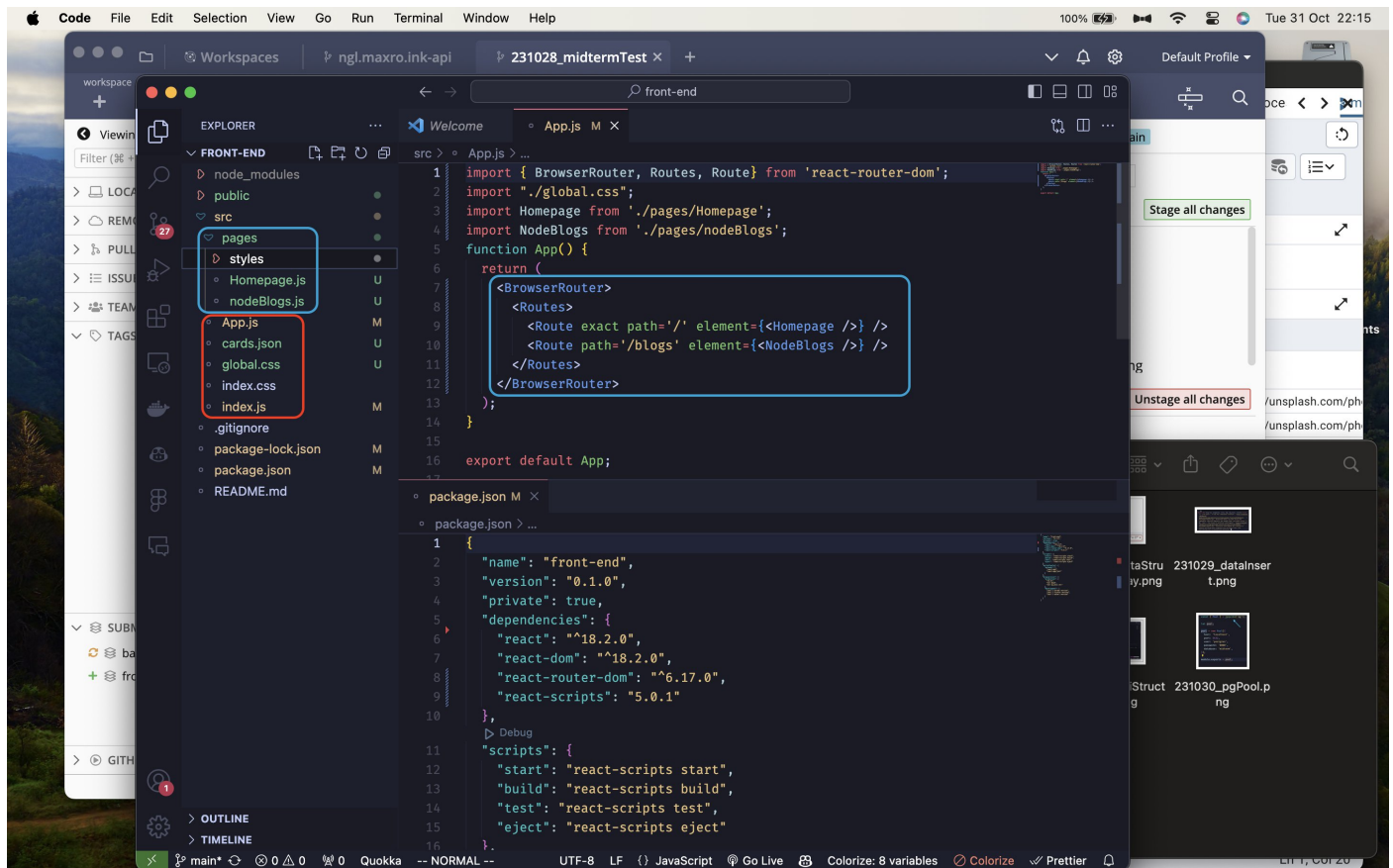
還記得前言我們寫了一個card.js嗎？我利用老師上課的方式對database送了query指令。並且比較暴力的直接用json回傳。至此，前端結束。

順帶一提，下面/:id的部分則是所謂動態路徑，當你輸入localhost/cards/1，他會只給你id為1的資料。但這不是老師上課的內容，目前不需要注意。

那麼，是時候走到前端去啦！

前端設計

Route的使用

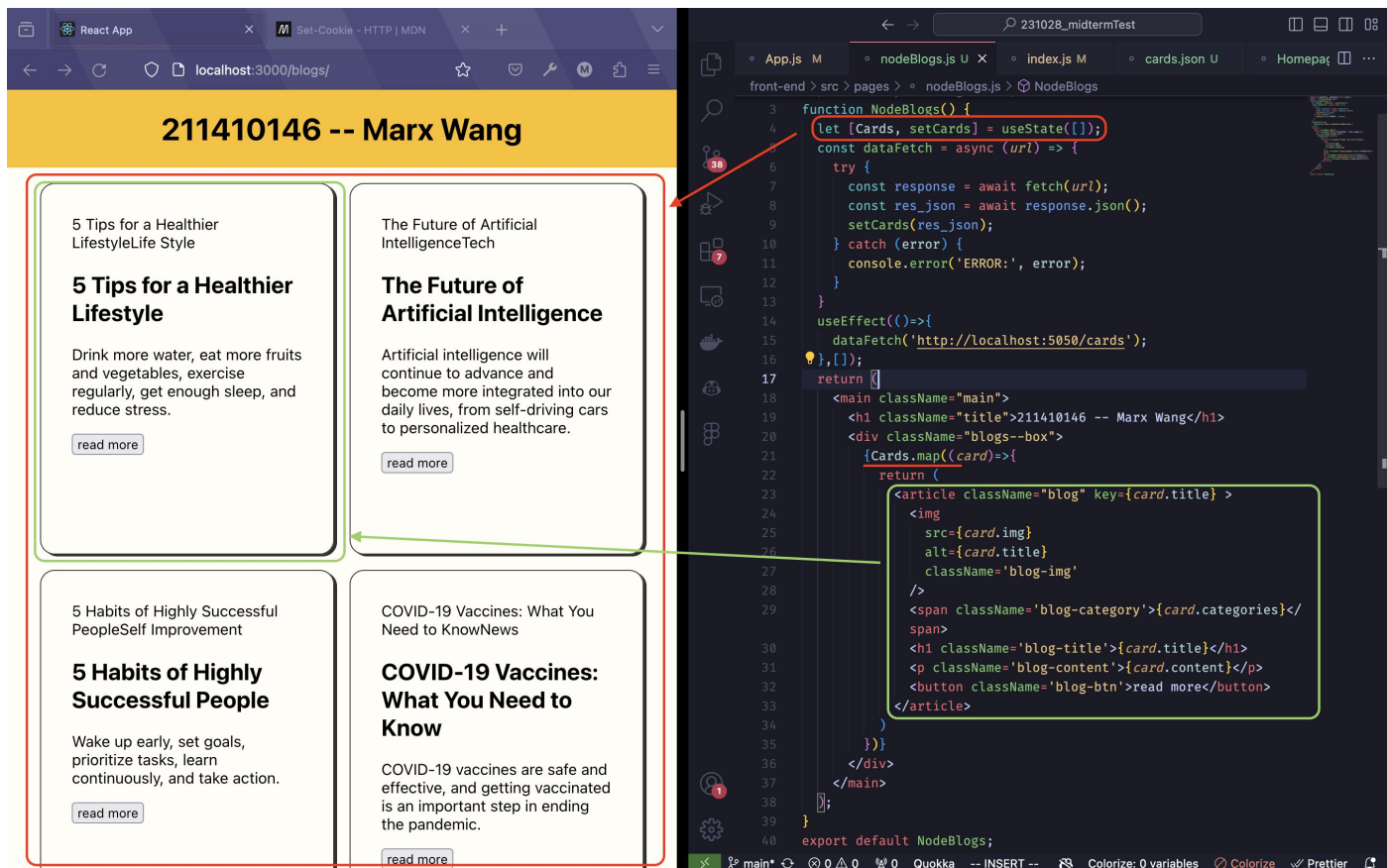


關注紅色區塊放兩個主要檔案，App.js跟index.js。index.js檔案負責render我們制定的Component，而Render的對象是App.js。老師的習慣是將App當成一個入口，而入口去決定要塞的Component。

接下來要講一嘴Router。上面藍色框框的Components全部來自react-router-dom。當我們打一個網址如marxw.me/cards，在靜態網頁時代回傳的是在這個Repository下cards這個資料夾的index.html。這種方式在許多框架都用（Next.js, Nuxt.js, Solid js之流）；React由於他產生頁面的方式是創立一個模擬的分支（Virtual DOM）並在分支下去管控，因此他在Routing的選擇用react-router-dom的模組進行管控，並選擇在「一個檔案內管理」的方式。

那我們選擇老師教的方式，在App.js中放入Browser Router，而且在Routes中放分支。請注意，放入Element中的東西是**Component**，不是Function，請記得加</>。

API fetch跟State的用法



上圖，我選擇用React的 `use state`來儲存API得來的資料。這裡要注意的是`useState`需要初始值，我在圖中寫的是array，因為我導入的是array形式。

至於理論上會顯示圖片，而實際只顯示字的問題，則是Firefox在CORS的控管+他把請求給拒絕了，所以請額外注意一下。有的時候錯的不是你，而是瀏覽器。