

Guess a number between 1 to 100

Practice Homework #5

Backend 練習 — Number Guessing Game

Due : 9pm, Monday, 2021-05-10

WEB PROGRAMMING 109-2

作業規定與建議步驟：

1. 這個作業是要大家使用 React/Axios/Express 框架，寫出一個簡單的全端應用 “Number Guessing Game”，也就是猜數字遊戲。
2. 這個作業已經將大部分的 **code** 寫好，大家只要完成 **code** 裡面標明 **TODO** 的部分就可以完成 **baseline** 的要求。

3. 請先到 wp1092 目錄下，創建一個 hw5 的子目錄：

```
cd wp1092
mkdir hw5
```

4. 到 hw5 目錄底下，用 “create-react-app” 建立 “own” 目錄：

```
cd hw5
create-react-app own
```

5. 到 “own” 目錄底下，安裝一些 server 需要用到的套件：

```
cd own
yarn add axios cors express
```

6. 同樣在 “own” 目錄底下，安裝一些開發時需要用到的套件：

```
cd own
yarn add -D @babel/cli @babel/core @babel/node @babel/preset-env nodemon
```

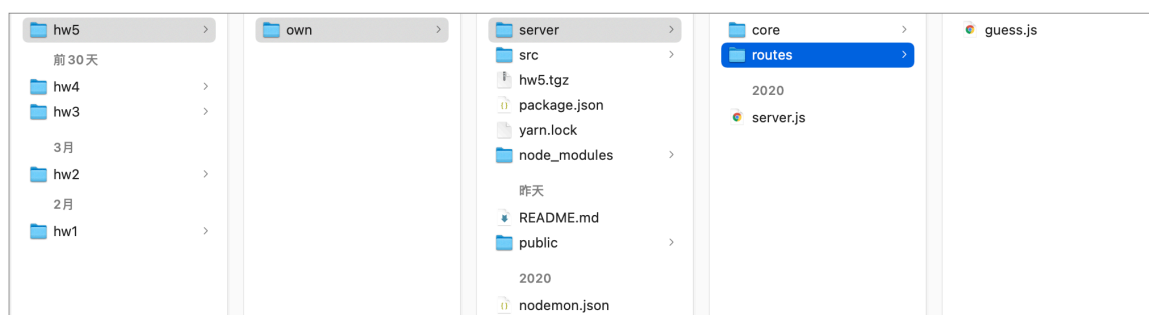
7. 移除 “create-react-app” 幫你建立的 “src” 子目錄，然後從 Ceiba 下載 Guessing Game 的 reference code “hw5.tgz” 到 “own”。解壓縮後，你應該會看到 reference code 的 “src” 以及 “server” 目錄，以及 “.babelrc” 和 “nodemon.json” 這兩個檔案：

```
cd own
rm -rf src
(從 Ceiba 下載 hw5.tgz)
tar zxvf hw5.tgz
rm -f hw5.tgz
```

解壓縮完請刪掉 hw5.tgz 這個檔案。

你如果解壓縮時多產生了一個 “xxxx_hw5” 的目錄，請將該目錄底下的 “src” 以及 “server” 目錄，以及 “.babelrc” 和 “nodemon.json” 這兩個檔案，搬到 “own” 底下來，然後刪掉那個空的目錄。如果因故目錄或是檔案的名稱變了，請把他們改回來。

你應該會在 own 底下看到如下之檔案與目錄：



WEB PROGRAMMING 109-2

8. 請在 “package.json” 檔案裡頭的 “scripts” 中 “test”: “react-scripts test” 後面加入一行：
`"server": "nodemon",` // 請注意，最後要加 ‘,’
 9. 執行本 app, 請在 “own” 目錄底下，先用 “yarn server” 開啟 server (at Port 4000), 然後在另外一個 terminal, 也是在 “own” 目錄底下，用 “yarn start” 開啟猜數字的網頁 (at Port 3000)。
 10. “Number Guessing Game” 的基本要求功能如下 (都 specify 在 code 裡面的 TODOs)：
 - (src/App.js) 處理當 “guess!” 按鈕按下去之後，把輸入的數字透過 Axios 傳給 server 端去做判斷，並且接收 server 傳回來的猜測狀態，設定對應的 states 值。
 - (server/core/getNumber.js) 當 “forceStart” 為 true 或 “number” 是 undefined 時，重新產生一個介於 1 ~ 100 的隨機亂數。否則，就直接回傳 “number”。
 - (server/routes/guess.js) 檢查猜的數字是否正確，並且回傳建議 (status) - Bigger (猜大一點), Smaller (猜小一點), Equal (答對了)
 - (server/routes/guess.js) 實作 “router.post('/restart,...)”, 以利在遊戲結束後可以重新開始
 - (src/axios.js) 修改 “guess” 去處理 server 回傳的 HTTP error (400), 並印出像 ‘Error: “xx” is not a valid number (1 - 100)’ 這樣的 error message.
 11. “Number Guessing Game” 的進階要求功能如下：
 - 在 “src/axios.js” 的各個 functions 處理 server not responding or not connected 的情況 (例如：server 被關掉)，並且在畫面 (src/App.js) 給予適當的訊息。(Note: 當 server 回來以後應該要可以繼續遊戲)
 - 在 “server” 底下開個 “log” 資料夾，把每次 server ON 以後遊戲的紀錄存在一個檔案裡，檔案名稱為 “yyyy-mm-dd-hh-mm.log” (例如：“2021-04-30-18-32.log”)，即以 server ON 的時間為檔名，記錄每個遊戲的動作，每個動作一行，格式建議如下 (記錄到 “秒”)：
 - ✓ start number=57 2021-04-30-18-33-27
 - ✓ guess 50 2021-04-30-18-33-50
 - ✓ guess 75 2021-04-30-18-34-08
 - ✓ guess 57 2021-04-30-18-34-22
 - ✓ end-game
 - ✓ restart number=38 2021-05-02-10-03-17

Note: 寫 log 檔如果發生錯誤，應屬於 server side 的問題，於 server side 印出適當的 error/alert 即可，不用回傳錯誤訊息給 client, 也不應該中斷遊戲。

Note: 每次 server 重開應該都要重新開啟一個 log 檔，直到 server 關掉為止

Note: 請將 “/log” 目錄加到 “.gitignore”, 不要把 log 也 check in 到 GitHub 上面去
 - 任何 UI/UX 的改進 (請在 README.md) 中說明
 12. 如果你 hw5 資料夾中有些檔案是沒有需要 push 到 GitHub 上去的，請記得把他們的檔名加到 .gitignore 去。
 13. 以上說明，如有不夠詳盡之處，請自行判斷與定義，不用問我們「這樣可不可以」、「這個要不要做」之類的問題，再次聲明 practice homework 不會有詳細的規格，「定義規格」也是 practice 的一部分，但為了讓 reviewer 能夠正確地使用你的 app, 請儘可能地在 README.md 中提供必要的說明。
-

繳交

請在 deadline (9pm, Monday, 2021-05-10) 之前把你的作業 (i.e. hw5 整個目錄) push 到你 GitHub 上 wp1092 這個 repo 上面 (Note: 我們不接受遲交哦!)。你應該會用到的指令有 `git add hw5` (i.e. 把整個 hw5 一次加進去), `git commit -m "你想留的 comment"`, 以及 `git push`. 你也可以用 `git status` 來檢查一下檔案是否都已經成功 push 到 GitHub 上面, 或者是登入 GitHub 去檢查。請注意, 你不用建立 "review" 這個資料夾, 助教會在指派 review 作業時再幫你加進去你的 repo。

如有任何的問題, 請到 FB 社團的 "HW#5 Q&As" 貼文底下發問, 或者是寫信到老師/助教信箱: eewebprogramming@googlegroups.com。
