React Hook簡介

Yu Hsiang 2019/01/11

Hook

React官方原本是以class方式進行網站開發,而在近期逐漸轉向利用function進行開發,Hook讓使用者可以只利用官方提供 函式去控制整個網頁的資料呈現。

參考: https://reactjs.org/docs/hooks-intro.html

前置作業 - 安裝

1.建立React專案

npx create-react-app my-app

網址:

https://facebook.github.io/create-react-app/

前置作業 - 改為支援Hook版本

2.更改為Hook支援版本 my-app/package.json

```
    package.json 

    x

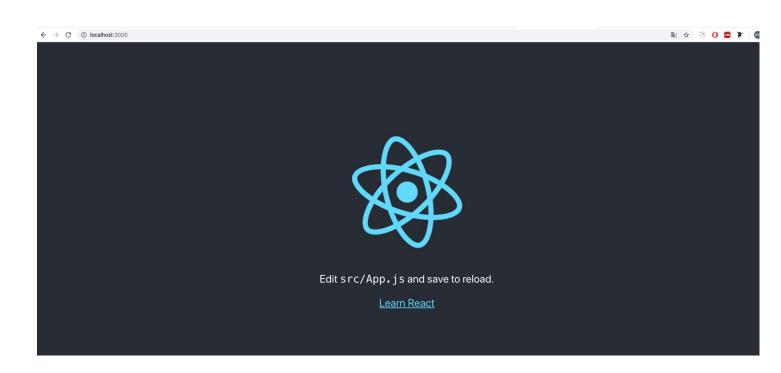
                                                           package.json ×
         "name": "my-app",
                                                                  "name": "my-app",
         "version": "0.1.0",
                                                                   "version": "0.1.0",
         "private": true.
                                                                  "private": true,
  5
         "dependencies": {
                                                                   "dependencies": {
  6
           "react": "^16.7.0",
  7
                                                                     "react": "16.7.0-alpha.2",
           "react-dom": "^16.7.0",
  8
           "react-scripts": "2.1.3"
                                                                     "react-dom": "16.7.0-alpha.2",
                                                                     "react-scripts": "2.1.2"
                                                           8
 10
         "scripts": {
 11
           "start": "react-scripts start",
                                                          10
                                                                  "scripts": {
 12
           "build": "react-scripts build",
                                                                     "start" "react_scripts start"
                                                          11
 13
           "test": "react-scripts test",
```

前置作業 - 安裝並運行

重新安裝:

npm install

npm start



利用Hook撰寫頁面

以下範例參考 https://reactjs.org/docs/hooks-state.html

撰寫第一個頁面

程式碼: http://codepad.org/4qh5qih1

編輯 my-app/src/App.js為新程式碼, 並且儲存

```
JS App.js
      import React, { Component } from 'react';
      import logo from './logo.svg';
      import './App.css';
      class App extends Component {
         render() {
           return (
             <div className="App">
  7
               <header className="App-header">
                 <imq src={logo} className="App-logo" alt="logo" />
  9
 10
                   Edit <code>src/App.js</code> and save to reload.
 11
 12
                 13
 14
                   className="App-link"
 15
                   href="https://reactjs.org"
 16
                   target=" blank"
 17
                   rel="noopener noreferrer"
 18
 19
                   Learn React
 20
                 </a>
 21
               </header>
 22
             </div>
 23
 24
 25
      export default App;
```

```
JS App.js
           ×
      import React, { useState } from 'react'
  3
      function App() {
        // Declare a new state variable, which we'll call "count"
  4
        const [count, setCount] = useState(0);
  6
  7
        return (
          <div>
  9
             You clicked {count} times
 10
             <button onClick={() => setCount(count + 1)}>
              Click me
 11
 12
            </button>
 13
          </div>
 14
 15
 16
 17
18
      export default App
 19
```

回到網站,網頁被更改了

http://localhost:3000/



```
程式碼檢視
```

```
JS App.js
          ×
                                                                         → 2.元件名稱 App
      import React,{ useState } from 'react'
      function App() {
                                                                          3.變數容器
          Declare a new state variable, which we'll call "count"
                                                                          變數 count
        const [count, setCount] = useState(0);
                                                                          以及 setter setCount
        return (
                                                                          初始值 ○
         <div>
           You clicked {count} times
 10
           <button onClick={() => setCount(count + 1)}>
                                                                           4.HTML主體
 11
             Click me
 12
           </button>
                                                                           按鈕的 onClick 上綁定了
 13
         </div>
                                                                           setter setCount
 14
                                                                           因此, setCount(count+1)
 15
                                                                           讓數值加 1
 16
 17
                                                     將元件允許被匯出
 18
      export default App
 19
```

1.用import匯入React套件

類似C#的using

讓程式更簡潔些 ES6 arrow function

參考

: https://developer.mozilla.org/zh-TW/docs/ Web/JavaScript/Reference/Functions/Arro w_functions

```
function App (){
}
```

可改為匿名函式的寫法

於是變成

```
JS App.js
          ×
                                                    import React, { useState } from 'react'
      import React, { useState } from 'react'
                                                    const App = () => {
  3
      function App() {
                                                          Declare a new state variable, which we'll call "count"
        // Declare a new state variable, which we
        const [count, setCount] = useState(0);
                                                      const [count, setCount] = useState(0);
        return (
                                                       return (
  8
         <div>
                                                         <div>
           You clicked {count} times
 10
           <button onClick={() => setCount(count
                                                           You clicked {count} times
 11
             Click me
                                                           <button onClick={() => setCount(count + 1)}>
                                              10
 12
           </button>
                                              11
                                                             Click me
 13
         </div>
                                              12
                                                           </button>
 14
                                                         </div>
 15
                                              13
 16
                                              14
 17
                                              15
 18
      export default App
                                              16
 19
                                              17
                                              18
                                                    export default App
                                              19
```

什麼是Hook?

React 提供在函式內管理UI上資料的方法,而基本上只會常用到紅字的方法

基本:

useState - 管理UI上的資料 useEffect - 用於載入頁面時與API拿資料 useContext

進階:

useReducer useCallback useMemo useRef useImperativeMethods useLayoutEffect

23

24

25 26

27

</div>

export default App

```
const [list, setList] = useState([])
抓取API資料並顯示
                                                        建立一個list變數與setter setList
JS App.is
                                                        並初始化為 [] 空陣列
     import React, { useState, useEffect } from 'react'
     const App = () => {
       // Declare a new state variable, which we'll call "count
  3
       const [count, setCount] = useState(0)
  5
       const [list, setList] = useState([])
       useEffect(() => { // 網頁第一次載入時呼叫API
  7
         //抓取資料
                                                            useEffect(() => { // 網頁第一次載入時呼叫API
         fetch('https://facebook.github.io/react-native/movies.json')
                                                              //抓取資料
           .then(response => response.json())
                                                              fetch('https://facebook.github.io/react-native/movies.json')
 10
           .then(responseJson => {
                                                                .then(response => response.json())
 11
            setList(responseJson.movies)
 12
                                                                .then(responseJson => {
 13
                                                                 setList(responseJson.movies)
 14
       return (
 15
         <div>

 [])

 16
          You clicked {count} times
                                                             載入網頁時跟API拿資料後利用setter setList設定
 17
          <button onClick={() => setCount(count + 1)}>
 18
            Click me
 19
           </button>
 20
            {list.map((it, index) => 第 {index + 1}項 {it.title}
 21
 22
```

{list.map((it, index) => 第 {index + 1}項 {it.title}

<11>

印出資料

回到網站, 呈現出 API資料

程式碼: http://codepad.org/gOE2whSD



You clicked 0 times

Click me

- 第 1項 Star Wars
- 第 2項 Back to the Future
- 第3項 The Matrix
- 第 4項 Inception
- 第 5項 Interstellar

概念上就是把資料整理成顯示的模式,

然後在視圖上印出, 不直接控制DOM

說到視圖, 可以將 邏輯 與 視圖 拆分

邏輯 與 視圖 拆分

```
JS App.js
      import React, { useState, useEffect } from 'react'
      // 視圖
      const AppView = ({ count, list, setCount }) =>
        <div>
          You clicked {count} times
          <button onClick={() => setCount(count + 1)}>
            Click me
          </button>
          10
            {list.map((it, index) => 第 {index + 1}項 {it.title}
 11
          12
        </div>
 13
       // 邏輯
      const App = () => {
 14
 15
        const [count, setCount] = useState(0)
 16
        const [list, setList] = useState([])
 17
        useEffect(() => { // 網頁第一次載入時呼叫API
          //抓取資料
 18
          fetch('https://facebook.github.io/react-native/movies.json')
 19
            .then(response => response.json())
 20
            .then(responseJson => {
 21
 22
              setList(responseJson.movies)
            })
 23
 24
        }, [])
 25
        return <AppView count={count} list={list} setCount={setCount} />
 26
      export default App
```

回到網站, 畫面還是一樣, 但程式碼抽離了

程式碼: http://codepad.org/ND5sHxCM



You clicked 0 times

Click me

- 第 1項 Star Wars
- 第 2項 Back to the Future
- 第 3項 The Matrix
- 第 4項 Inception
- 第 5項 Interstellar

就這簡單的操作, 便可以控制整個頁面上

的邏輯了

