

Module 18 – 體會 SOP，實作 CORS

情節描述：

寫作 Web API 服務並公開到網際網路後，用戶端很可能不會與我們的伺服器位於同一網域。基於安全理由，瀏覽器會限制跨域存取，以致於許多用戶端無法使用我們提供的 Web API 服務。

預設目標：

- 完成 Web API 服務。
- 體驗果然無法跨域使用 Web API 服務。
- 實作 CORS 實現跨域呼叫 Web API。

估計時間： 20 分鐘

★★★ Lab01: 體會 SOP 同源政策的限制 ★★★

1. 啟動 VS Code 並開啟 c:\Lab\Lab_cors 資料夾。
2. 在 VS Code 開啟檔案 index.js。瀏覽一下這支程式：

```
// 以 Express 建立 Web 伺服器
var express = require("express");
var app = express();

// Web 伺服器的靜態檔案置於 public 資料夾
app.use( express.static( "public" ) );

// 一切就緒，開始接受用戶端連線
// app.listen(process.env.PORT);
app.listen(3000);
console.log("Web 伺服器就緒，開始接受用戶端連線.");
```

```
console.log("鍵盤「Ctrl + C」可結束伺服器程式.");  
  
// -----  
  
app.get("/hello/:text", function (request, response) {  
    response.send("Hello! " + request.params.text);  
});
```

3. 以系統管理員身份啟動「記事本」。

<提示>

組合「視窗鍵 + S」，搜尋 *notepad*，滑鼠右鍵點按「記事本」，選擇「以系統管理員身份執行」。

4. 「記事本」功能表檔案 | 開啟舊檔 | 檔案名稱輸入：

C:\Windows\System32\drivers\etc\hosts

加入下列兩行並且存檔：

127.0.0.1 www.beauty.com

127.0.0.1 www.monster.com

5. 在 VS Code 按下組合鍵「Ctrl + `」啟動「Terminal 終端機視窗」，輸入下列指令以啟動網站伺服器。

npm install express

npm install cors

node index.js

6. 打開瀏覽器，在網址列上分別輸入

`http://www.monster.com/hello/Jeremy`

`http://www.beauty.com/hello/Mary`

應該會看「Hello! Jeremy」、「Hello! Mary」字樣。

<提示>其實，這兩個網址都模擬成目前這部電腦。

7. 在 VS Code 開啟 index.html 網頁，瀏覽一下內容：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
</head>
<body>
  Your name:
  <input type="text" id="userName" />
  <input type="button" id="helloButton" value="Hello" />
  <div id="report"></div>
  <script>
    $("#helloButton").click(function () {
      let userName = $("#userName").val();
      let url = "http://www.beauty.com/hello/";
      // let url = "http://www.monster.com/hello/"
      $.get(url + userName, function (data) {
        $("#report").text(data);
      })
    })
  </script>
</body>
</html>
```

```
</script>  
</body>  
</html>
```

8. 瀏覽器的網址列輸入下列網址:

<http://www.beauty.com/index.html>

在文字輸入方塊輸入你的名字，然後點按「Hello」按鈕。

9. 在 VS Code 開啟 index.html 網頁，將 let url 那行，改成:

```
let url = "http://www.monster.com/hello/"
```

10. 瀏覽器的網址列輸入下列網址:

<http://www.beauty.com/index.html>

在文字輸入方塊輸入你的名字，然後點按「Hello」按鈕。

<提示>這回，程式久久沒有反應。

11. 請想一下，為何同一支程式，用 beauty.com 可以使用，改成 monster.com 卻不可以。

★★★ Lab02: 實作 CORS ★★★

1. VS Code 的終端機視窗，按下組合鍵「Ctrl + C」中斷我們的網站伺服器。

2. 編輯檔案 index.js，在 `var app = express();` 的下一行，加入下列程式：

```
// 允許跨域使用本服務  
var cors = require("cors");  
app.use(cors());
```

3. 在 VS Code 按下組合鍵「Ctrl + `」啟動「Terminal 終端機視窗」，輸入下列指令以啟動網站伺服器。

```
node index.js
```

4. 瀏覽器的網址列輸入下列網址：

```
http://www.beauty.com/index.html
```

在文字輸入方塊輸入你的名字，然後點按「Hello」按鈕。

<提示> 這回，雖然跨域了，但瀏覽器會接受伺服器傳回的資料。

5. 編輯檔案 index.js，針對「// 允許跨域使用本服務」，改成：

```
// 允許跨域使用本服務
var cors = require("cors");
// app.use(cors());
const corsOptions = {
  origin: [
    'http://www.beauty.com',
    'http://localhost',
  ],
  methods: 'GET,HEAD,PUT,PATCH,POST,DELETE,OPTIONS'
};
app.use(cors(corsOptions));
```

6. VS Code 的終端機視窗，按下組合鍵「Ctrl + C」中斷我們的網站伺服器。重新以「node index.js」重新啟動伺服器。

7. 在瀏覽器的確認網址瀏覽器仍可接受伺服器傳回的資料。

8. 記得用記事本，清除 C:\Windows\System32\drivers\etc\hosts 檔案的下列兩行：

```
127.0.0.1 www.beauty.com
127.0.0.1 www.monster.com
```