

Module 18 –連接資料庫

情節描述：

本練習假設 Node.js 環境已經安裝在 Windows 電腦內。接下來，練習安裝 MySQL、安裝 mysql 模組，練習如何從 Node.js 連結 MySQL，並且使用相關的 API 顯示資料庫內的資料。

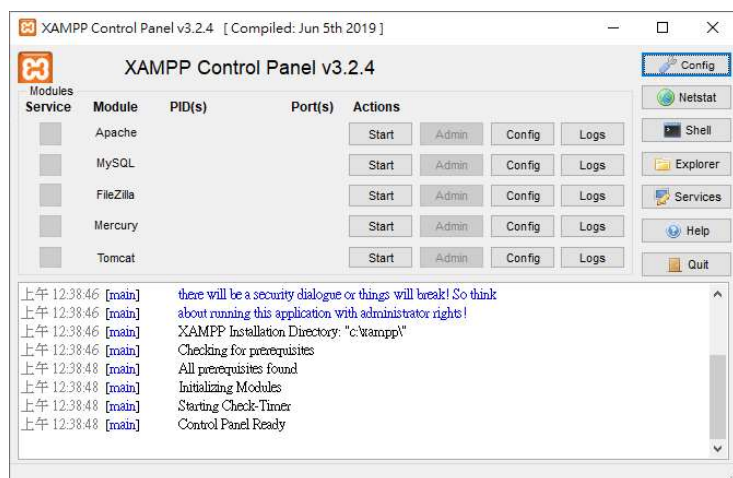
預設目標：

- 成功安裝 MySQL
- 從 Node.js 連接 MySQL
- 使用相關的 API 顯示資料庫內的資料

估計時間：30 分鐘

★★★★ Lab01: 安裝 mysql 伺服器 ★★★★★

1. 檢查 C:\xampp 底下是否有目錄，若已經有了則代表已經安裝過 MySQL
2. 從 XAMPP 官方網站(https://www.apachefriends.org/zh_tw/download.html)下載 XAMPP 安裝程式，若無特殊需求，執行安裝程式時，一路按 next 下一步到底。
3. 檢查 C:\xampp\xampp-control.exe 是否存在，該程式是 XAMPP 的主控程式，執行之的畫面如下：



4. 按下 MySQL 旁邊的「Start」按鈕，可啟動 MySQL 資料庫伺服器

★★★ Lab02: 安裝 mysql 模組套件 ★★★

1. 在 C 槽底下新建一個資料夾，命名為「Lab」，開啟「命令提示字元」，輸入「cd C:\Lab」，切換到工作目錄。
2. Lab 工作目錄主要是用來裝之後要建立的專案目錄，因為之後實際開發可能會有許多專案目錄。
3. 在「命令提示字元」輸入：mkdir myapp，建立名為 myapp 的資料夾作為專案名稱。
4. 輸入：cd myapp 切換到該目錄
5. 輸入「code .」，啟動 VS Code 編輯 myapp 資料夾。
6. 點選「新增檔案」按鈕，輸入 index.js 為檔名
7. 按下「Ctrl + 撇號」啟動終端機視窗，在終端機視窗輸入「npm init -f」將該資料夾轉成一個 node 專案。
8. 輸入：「npm install mysql --save」安裝 mysql 模組套件。
9. 輸入：「npm install express --save」安裝 express 模組套件。

★★★ Lab03: 連線 MySQL 資料庫伺服器 ★★★

1. 點按「新增檔案」按鈕，檔名：setupdb.js，編輯內容如下：

```
var mysql = require("mysql");
```

2. 繼續編寫資料庫連線程式：

```
var conn = mysql.createConnection({  
  host: '127.0.0.1',  
  user: 'root',  
  password: ''  
});  
  
conn.connect(function (err) {  
  if (err) {  
    console.log(JSON.stringify(err));  
  }  
});
```

```
        return;  
    }  
});
```

2. 繼續加入下列程式。送出 Create Database 指令給資料庫伺服器：

```
const databaseName = 'labdb';  
conn.query('CREATE DATABASE IF NOT EXISTS ${databaseName};', function (err) {  
    if (err) {  
        console.log(JSON.stringify(err));  
        return;  
    }  
});
```

3. 繼續加入下列程式。切換到 labdb 資料庫之後，建立 news 資料表：

```
conn.query('USE ${databaseName}', '', function (err) {  
    if (err) {  
        console.log(JSON.stringify(err));  
        return;  
    }  
})  
  
const tableName = 'news';  
const addTableSQL = 'CREATE TABLE IF NOT EXISTS ${tableName} (\n  
    newsId INT NOT NULL AUTO_INCREMENT,\  
    title VARCHAR(100) NOT NULL,\  
    ymd Date NOT NULL,\  
    PRIMARY KEY ( newsId )\  
    )ENGINE=InnoDB DEFAULT CHARSET=utf8';
```

```
conn.query(addTableSQL, '', function (err) {  
  if (err) {  
    console.log(JSON.stringify(err));  
    return;  
  }  
});
```

4. 繼續加入下列程式。新增一筆資料到 news 資料表：

```
const insertNewsSQL = `INSERT INTO ${tableName} (title, ymd) \\  
  VALUES ("play with nodejs", '2020-02-29')`;   
conn.query(insertNewsSQL, function (err) {  
  if (err) {  
    console.log(JSON.stringify(err));  
    return;  
  }  
  process.exit(0);  
});
```

5. 在終端機視窗輸入「node setupdb.js」

★★★ Lab04: 查詢資料並傳出 JSON 字串給用戶端 ★★★

1. 點按「新增檔案」按鈕，檔名：index.js，編輯內容如下：

```
var express = require("express");
var mysql = require('mysql');
var app = express();
app.listen(80);
```

2. 繼續編寫資料庫連線程式：

```
var conn = mysql.createConnection({
  host: '127.0.0.1',
  user: 'root',
  password: '',
  database: "labdb"
});

conn.connect(function (err) {
  if (err) {
    console.log(JSON.stringify(err));
    return;
  }
});
```

3. 繼續加入下列程式。查詢資料並且傳回 JSON 字串給用戶端：

```
app.get("/home/news", function (request, response) {
  conn.query('select * from news',
    '',
    function (err, rows) {
      if (err) {
        console.log(JSON.stringify(err));
        return;
      }
    }
  );
});
```

```
        }  
        response.send(JSON.stringify(rows));  
    }  
);  
})
```

4. 在終端機視窗輸入「node index.js」
5. 在瀏覽器，網址列上輸入「http://localhost/home/news」。
執行結果：
[{"newsId":1,"title":"play with nodejs","ymd":"2020-02-28T16:00:00.000Z"}]