

## Module 17 – 使用 Bluebird

情節描述:

本 Lab 假設環境已安裝 MySQL (或 MariaDB) 資料庫，並已經有 Node.js/Express 套件模組，可透過 `mysql` 套件連線資料庫並讀取資料。接下來，學員要接著安裝 Bluebird 套件，並達成分頁資料 API 的建立。

預設目標:

利用 Bluebird 完成產品資料分頁 API。路由使用 `"/products/:page?"`，取得的 JSON 格式如下：

```
{
  "page": "目前頁數",
  "perPage": "每頁幾筆，數值可固定 6",
  "totalRows": "總筆數",
  "totalPages": "總頁數",
  "rows": "產品分頁的資料，陣列"
}
```

估計時間: 20 分鐘

### ★★★ Lab01: 安裝 Bluebird 模組 ★★★

1. 在命令提示字元視窗，工作目錄切換到專案目錄，並以下列安裝 Bluebird。

```
npm i bluebird
```

2. 在連線的模組中(例如檔名為 `_connect_db.js`)，引入 bluebird 並以 `promisifyAll()` 包裝 `mysql` 連線物件，範例碼如下：

```
const mysql = require('mysql');
const bluebird = require('bluebird'); // 引入 bluebird
```



```
const db = mysql.createConnection({
  host: 'localhost',
  user: 'root', // 連線 DB 的帳號
  password: '', // 密碼
  database: 'test', // 資料庫名稱
});
db.connect();
bluebird.promisifyAll(db); // 使用 promisifyAll 包裝
module.exports = db;
```

### 3. 產品資料表範例 SQL 如下。

```
CREATE TABLE IF NOT EXISTS `products` (
  `sid` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `price` int(11) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;

INSERT INTO `products` (`sid`, `name`, `price`) VALUES
(1, '口罩', 22),
(2, '衛生紙', 20),
(3, '帽子', 300),
(4, '雨傘', 200),
(5, '鉛筆', 10),
(6, '雨衣', 350),
(7, '鞋子', 1000),
(8, '手帕', 120);

ALTER TABLE `products`
  ADD PRIMARY KEY (`sid`);

ALTER TABLE `products`
  MODIFY `sid` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=9;
```

## ★★★ Lab02: 在 index.js 內建立有功能的路由 ★★★

1. 建立 “/products/:page?” 的路由內容如下：

```
app.get('/products/:page?', (req, res)=>{
  const perPage = 6; // 每頁幾筆
  let page = parseInt(req.params.page) || 1; // 預設為第一頁
  const output = { perPage, page };

  const sql1 = "SELECT COUNT(1) num FROM products"; // 取得總筆數的 SQL
  const sql2 = `SELECT * FROM products LIMIT ${ (page-1)*perPage }, ${perPage}`;
  db.queryAsync(sql1)
    .then(results=>{
      output.total = results[0].num; // 取得總筆數
      output.totalPages = Math.ceil(output.total/perPage); // 計算總頁數

      if(page<1 || page>output.totalPages){
        return res.redirect('/products'); // 若查看的分頁超出範圍，則重新轉向
      }
      return db.queryAsync(sql2);
    })
    .then(results=>{
      output.rows = results;
      res.json(output);
    })
    .catch(ex=>{
      console.log(ex);
    });
});
```

2. 以 nodemon 啟動 index.js 並使用瀏覽器連至「<http://localhost:3000/products>」或「<http://localhost:3000/products/2>」查看結果。



```
localhost:3000/products

{
  perPage: 6,
  page: 1,
  total: 8,
  totalPages: 2,
  - rows: [
    - {
      sid: 1,
      name: "口罩",
      price: 22,
    },
    - {
      sid: 2,
      name: "衛生紙",
      price: 20,
    },
    - {
      sid: 3,
      name: "帽子",
      price: 300,
    },
    - {
      sid: 4,
      name: "雨傘",
      price: 200,
    },
    - {
      sid: 5,
      name: "鉛筆",
      price: 10,
    },
    - {
      sid: 6,
      name: "雨衣",
      price: 350,
    },
  ],
}
```

```
localhost:3000/products/2

{
  perPage: 6,
  page: 2,
  total: 8,
  totalPages: 2,
  - rows: [
    - {
      sid: 7,
      name: "鞋子",
      price: 1000,
    },
    - {
      sid: 8,
      name: "手帕",
      price: 120,
    },
  ],
}
```