

Hw5 書面報告

生機一

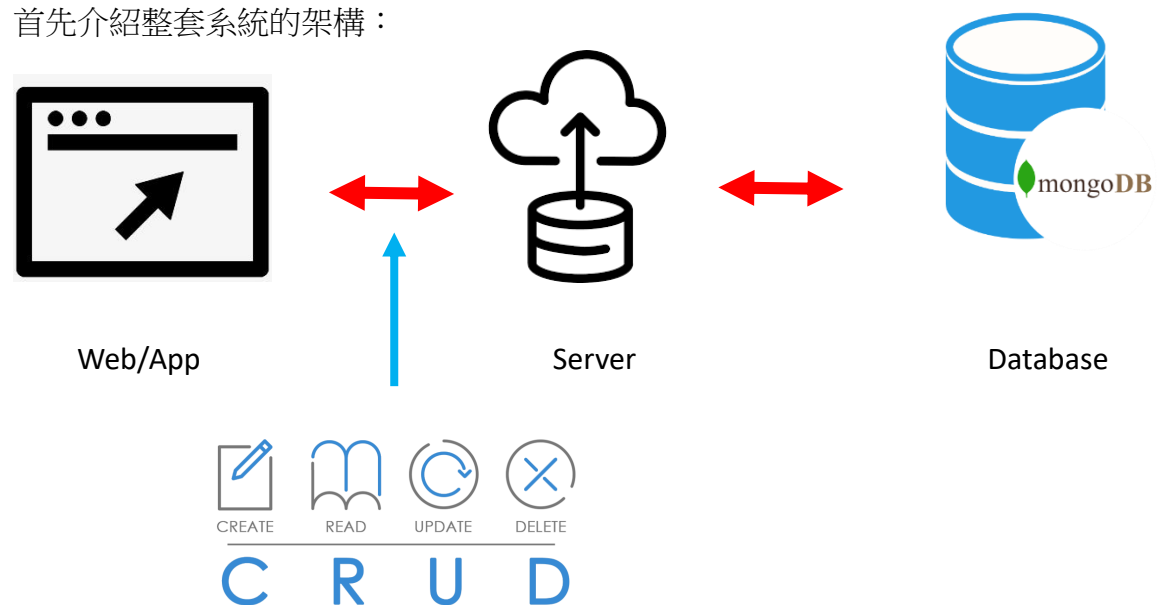
陳玟鉸

R09631011

根據老師給的網址 <https://zellwk.com/blog/crud-express-mongodb/>，學習運用 Node，Express，跟 MongoDB 來實作 CRUD(Create, Read, Put, Delete)的網頁，其中遇到不少問題。而由於不熟悉 json、html 跟 vim，因此花費大量時間在查語法。附上 GitHub 的 URL：

https://github.com/WenAnChen/Cloud_computing/tree/main/Hw5

首先介紹整套系統的架構：



由於是滾動式複寫程式，因此先前的設定由 github 上的程式及教學網頁上的節圖解釋，最後會附上實作結果：

1. Node

首先確認有無 Node，沒有的話需下載：

```
steven@DESKTOP-QSDES0:~$ node -v
v14.15.0
```

接著在指定資料夾中下指令：`$ npm init` 便能創建一個 package.json (config file 的概念)

透過 `$ touch server.js` 來創建 server.js 檔，並在其中加入 `console.log()`，功能像 C++ 的 `cout` 來看是否能夠執行，`$ node server.js` 來執行，若有輸出則代表 Node 功能正常。

2. Express

`$ npm install express --save` 下載 Express，Express 為 Node web 框架，並有大量相容的套件可以使用。

之後在 `server.js` 檔中加入

```
const express = require('express')
const app = express()
```

```
app.listen(3000, function() {
  console.log('listening on 3000')
})
```

使網頁與 node 串聯。

3. CRUD - read

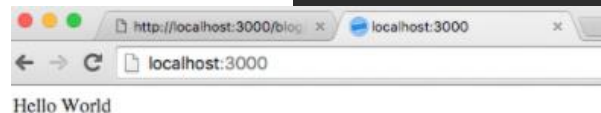
```
app.get('/', function(req, res) {
  res.send('Hello World')
})
```

Get function 主要用來告訴 server 當有請求符合端點時該如何回應，因此有兩個變數：request, response，而這裡會傳送 Hello word 出去。

也可以使用箭頭函式表示：

```
app.get('/', (req, res) => {
  res.send('Hello World')
})
```

結果：



3. `$touch index.html` 創建一個 html 檔來設計前端網頁，`index.html` 的程式碼也在 github 的 Url 可以找到。

4. `$ npm install nodemon --save-dev` 下載 Nodemon，Nodemon 可以自動的重啟 server 避免因網路斷訊時網頁就從此當機。此後若要啟動 server 時，指令需更改為：`$ npm run dev`

5. CRUD - create

在 `server.js` 中增加 post 的函式，可以接受 html 上的字串並發送給 server。

```
app.post('/quotes', (req, res) => {
  console.log('Hellooooooooooooooooooooo!')
})
```

測試結果：啟動 `server.js` 時會輸出

```
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
listening on 3000
Hellooooooooooooooooooooo!
```

6. MongoDB

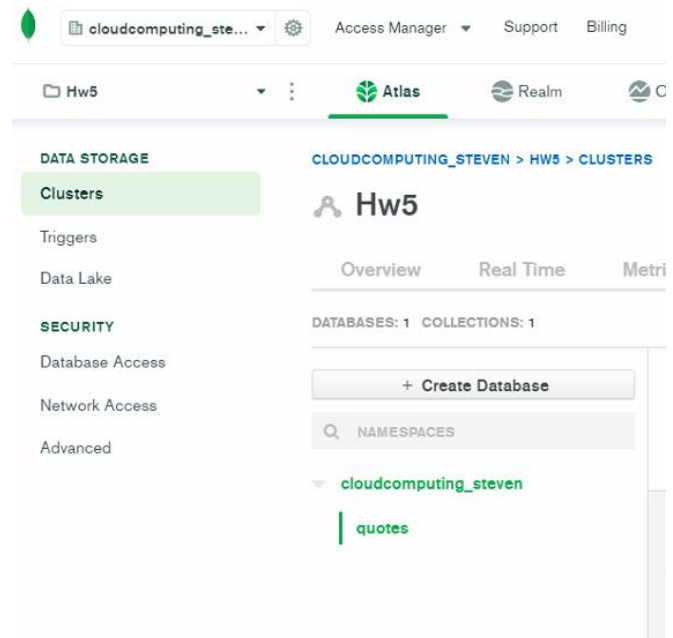
先在 ubuntu 上下載支援 npm 的 mongoDB

`$ npm install mongodb --save`

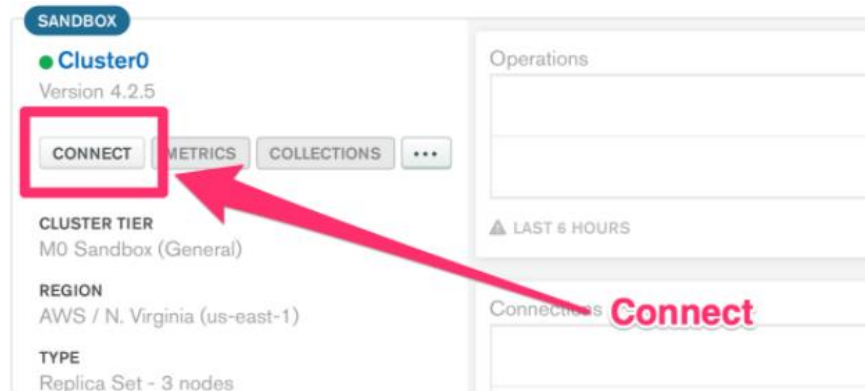
之後便能使用其中的函式連線

```
const connectionDB = "mongodb+srv://stevenDB:1996y1005@Hw5.yuqkq.mongodb.net/Hw5?retryWrites=true&w=majority"
MongoClient.connect(connectionDB, { useUnifiedTopology: false }).then(client => {
  console.log('Connected to Database')
  const db = client.db('cloudcomputing_steven')
  const quotesCollection = db.collection('quotes')
```

7. 創建 MongoDB 的 database 帳號：
Project 名稱為
cloudcomputing_steven，Cluster 名
稱為 Hw5



創建 Cluster 之後點選 connect，並
輸入允許存取的 IP 位址(由於再創建一個 cluster 需要收費，因此擷取教學
網站上的圖案示意)



並創建使用者名稱及密碼：

An IP address has been whitelisted. Add another whitelist entry in the IP Whitelist tab.

Create a Database User

This first user will have **atlasAdmin** permissions for this project.
Keep your credentials handy, you'll need them for the next step.

Username
stevenDB

Password
19c...05 [Autogenerate Secure Password](#) [HIDE](#)




[Create Database User](#)

[Choose a connection method](#)

選擇連線項目：

Choose a connection method [View documentation](#)

Get your pre-formatted connection string by selecting your tool below.

-  **Connect with the mongo shell**
Interact with your cluster using MongoDB's interactive Javascript interface
-  **Connect your application**
Connect your application to your cluster using MongoDB's native drivers
-  **Connect using MongoDB Compass**
Explore, modify, and visualize your data with MongoDB's GUI

最後會產生 Url，要複製下來貼到 server.js 中的 MongoClient connect

Connection String Only **Full Driver Example**

```
mongodb+srv://yoda:<password>@cluster0-rmp3c.mongodb.net/test?ret
```

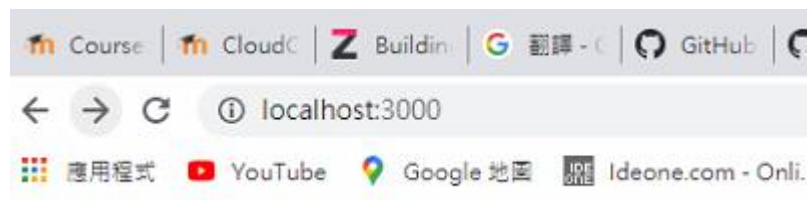
[Copy](#)

Replace **<password>** with the password for the user, **yoda**, and ensure all special characters are **URL encoded**.

- 改寫 server.js 中的程式，接收網頁上 html 中的字串(results)，並傳給 mongoDB 做儲存。
而當整套系統能成功運作時，會輸出 log file。

結果：

```
[node:4168] starting `node server.js`
(node:4168) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in
a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to
the MongoClient constructor.
(Use `node --trace-deprecation ...` to show where the warning was created)
Connected to Database
listening on 3000
CommandResult {
  result: {
    n: 1,
    opTime: { ts: [Timestamp], t: 2 },
    electionId: 7fffffff000000000000000002,
    ok: 1,
    '$clusterTime': { clusterTime: [Timestamp], signature: [Object] },
    operationTime: Timestamp { _bsontype: 'Timestamp', low_: 2, high_: 1607415363 }
  },
  connection: Connection {
    _events: [Object: null prototype] {
      error: [Function (anonymous)],
      close: [Function (anonymous)],
      timeout: [Function (anonymous)],
      parseError: [Function (anonymous)],
      message: [Function (anonymous)]
    },
    _eventsCount: 5,
    _maxListeners: undefined,
    id: 0,
    options: {
      host: 'hw5-shard-00-01.yuqkq.mongodb.net',
      port: 27017,
      size: 5,
      minSize: 0,
      connectionTimeout: 10000,
      socketTimeout: 0,
      keepAlive: true,
      keepAliveInitialDelay: 120000,
      noDelay: true,
      ssl: true,
      checkServerIdentity: true,
      ca: null,
      crt: null,
      cert: null,
      key: null,
      passphrase: null,
      rejectUnauthorized: true,
      promoteLongs: true,
    }
  }
}
```



Online money bank

<input type="text" value="name"/>	<input type="text" value="price"/>	<input type="button" value="Submit"/>
-----------------------------------	------------------------------------	---------------------------------------

網頁：輸入項目及價格，使 mongoDB 能記錄花費

此時從 mongoDB 的網頁中也能看到在 Web 所輸入的字串：

Name：陳玟銓，qoute：15684

DATABASES: 1 COLLECTIONS: 1

+ Create Database

Q NAMESPACES

▼ cloudcomputing_steven

quotes

cloudcomputing_steven.quotes

COLLECTION SIZE: 59B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes

FILTER {"filter": "example"}

QUERY RESULTS 1-1 OF 1

```
{
  "_id": ObjectId("5fcf3643aa0e8610486e5f1a"),
  "name": "陳玟銓",
  "quote": "15684"
}
```

代表此 Node，Express，跟 MongoDB 及 CRUD 中的 CR 有成功串聯。