

# 中國文化大學教師教學創新暨教材研發 獎勵期末報告

計畫名稱:

個案探討教學模式下結合雲端運算以及物連網課程教材開發計畫

資訊管理學系

賴谷鑫

中華民國一〇七年一月



## 目錄

壹、計畫名稱.....	4
貳、實施課程、授課教師姓名.....	4
參、前言.....	4
肆、計畫特色與具體內容.....	5
伍、實施成效及影響.....	6
陸、結論.....	6
柒、附錄.....	7

## 壹、計畫名稱

本計畫名稱為個案探討教學模式下結合雲端運算以及物連網課程教材開發計畫

## 貳、實施課程、授課教師姓名

本計畫實施的課程名稱為資管所一年級之雲端應用程式開發、實施教師為資管系兼任助理教授賴谷鑫老師。

## 參、前言

雲端應用與物連網為目前最主流的兩項資訊技術，雲端運算可分為 SaaS, PaaS, IaaS。Cloud Computing 分層，由底層往上分別是 Infrastructure（基礎建設）、Platform（平台）以及 Platform 上面的軟體以及應用程式例如 email, games。不同的分層有其不同的應用以及重點。本校資管系發展重點著重於應用程式的開發，而傳統上應用程式的開發有其專屬的開發工具並且要自己架設以及維護伺服器。然而自建系統需要鉅額的建置以及維護成本。因此新型態的雲端服務成為系統開發的新的選擇。在雲端程式設計解決方案部分，最主流的就是 Google 所提供的 GAE(google app engine)以及微軟的 Azure。下圖 1 為 GAE 的系統截圖。

雲端平台提供了很多重要的服務，如自動備份、資訊安全、版本控制、多人協同合作並且整合網頁以及資料庫平台。因此雲端平台上開發系統是未來系統開發的一個重要選項。

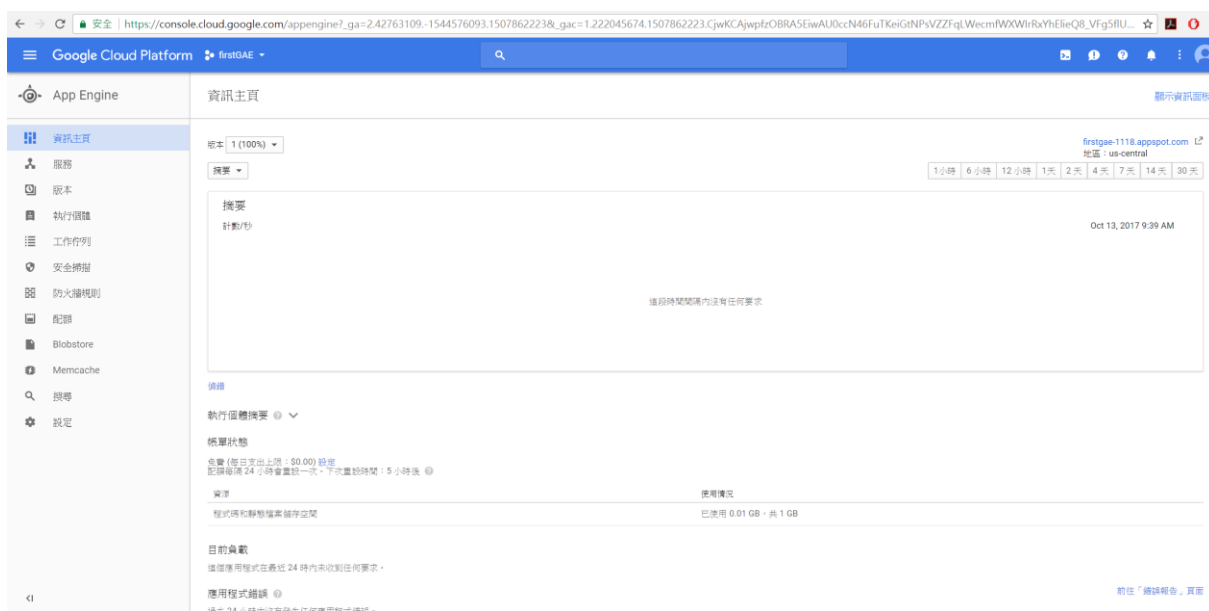


圖 1:GAE 簡單截圖

物連網也是目前目前資訊科技的一大主流，透過微小感測器收集並且分析資訊，藉以達到智慧型的應用系統(如智慧居家、智慧農業、智慧交通等等)的目的。然而在一般的物連網系統上由於物連網裝置本身的計算能力以及電力相對較弱，因此所感測的資料還是要透過電腦或是伺服器的儲存以及運算。一般物連網系統開發者還是得面對到自建伺服器所面對到的成本以及維護心力。因此目前雲端運算以及物連網應用面對到的幾的問題(1)有關雲端應用程式開發目前市面上有深度的教

課書不多；(2)物連網系統的書，缺乏與雲端系統結合的案例。

本計畫的目的則是為了改善上述二個問題，本計畫將撰寫雲端運算平台以及結合物連網應用的教材，其重點有(1)整合雲端平台的課程內容；(2)強調實作之系統開發；(3)利用物連網系統收集資料並且將資料傳輸到雲端平台做運用分析。透過本計畫所提供的教材，學生可以(1)學習到完整雲端平台程式開發；(2)增進問題解決的能力；(3)了解業界所需，增強自我競爭力；(4)了解物連網系統基礎知識；(5)整合雲端以及物連網的能力。

#### 肆、計畫特色與具體內容

本計畫特色為整合雲端應用程式平台以及物連網系統開發。本計畫提供的課程主要在利用物連網系統收集資料上傳到雲端系統，並且利用雲端系統開發相對應的智慧型應用程式。透過真實本計畫所設計的整合式教材，勢必可以增進學生雲端平台運用以及物連網的基本知識。

本計畫的執行方法如下圖 2 所示。

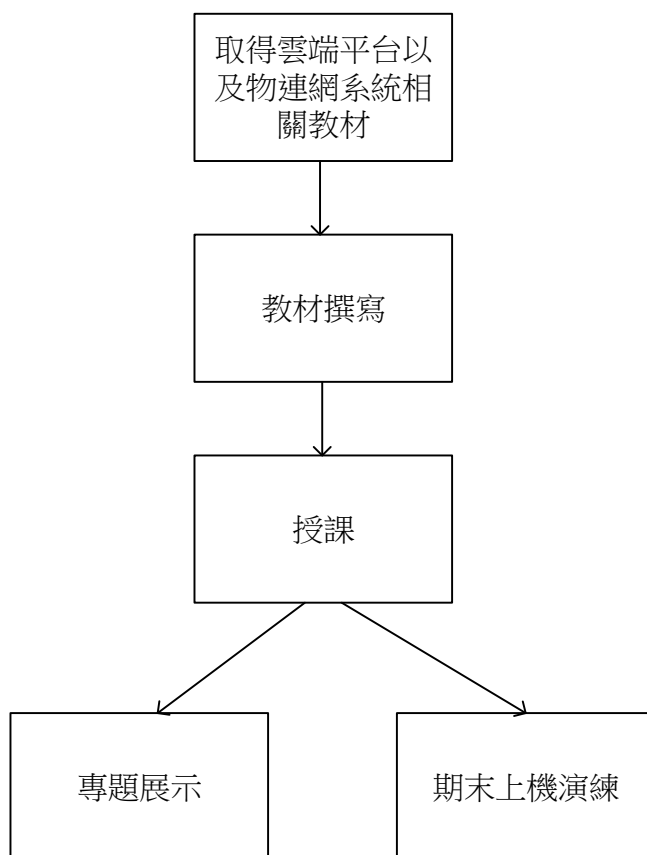


圖 2:專案執流程圖

**Step 1: 取得雲端平台以及物連網系統相關教材:** 本計畫將使用 Google GAE，微軟 Azure 之雲端平台，並且使用樹莓派以及 Arduino 作為物連網硬體設備。上述平台以及硬體都有英文官方網站或是教課書。

**Step 2: 教材撰寫:** 上述的文件皆無法製作成整合教材。本計畫規劃首先以雲端平台系統開發為基礎，首先介紹開發工具以及雲端平台資料庫。隨後介紹物連網系

統。物連網系統教材著重在(1)資料的收集與處理以及(2)資料的傳遞。

**Step 3: 授課:**本計畫所預計採用的授課方式為課堂授課、上機實作以及專題教學。課堂授課為本計畫所編寫的教材；於上機實作部分，本課程會透過分組方式，上機實作有兩個議題，第一個議題為雲端平台程式開發。此議題主要為學生學習雲端平台開發工具的使用、雲端平台開發的實務以及雲端平台伺服器管理。而另一個議題為物連網系統，此議題主要著重在物連網系統的建置，物連網系統的管理。而最重要的一點就是如何透過網路通訊協定將物連網的資料傳到雲端伺服器。所用的方式有(1)藍芽；(2)WIFI 以及(3)行動通訊系統。

**Step 4: 計畫與教學評量:**為了評量學生的學習成果，本課程利分組專題以及上機實作作為教學評量。其中上機實作分為雲端平台系統開發以及物連網系統的建置。上機實作可以了解該組學生對於雲端平台以及物連網有無深入了解。但是上機實作缺乏對問題解決的能力以及整合能力，因此本課程也會採用個案專題的方式針對學生創意、整合力以及問題解決能力。本課程期望達到至少每位學生對於雲端平台系統以及物連網的基礎知識都有所了解而進而了解業界知識。

## 伍、實施成效及影響

本計畫已實行完畢，由學生學習評量高於學校平均可以得知本計畫實施受同學肯定。透過本計畫的實行學生得到的成果如下。

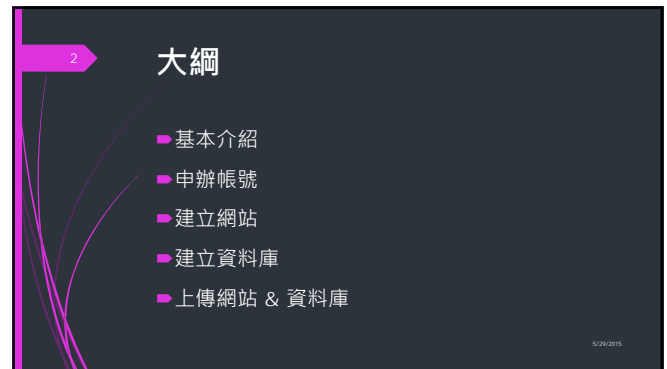
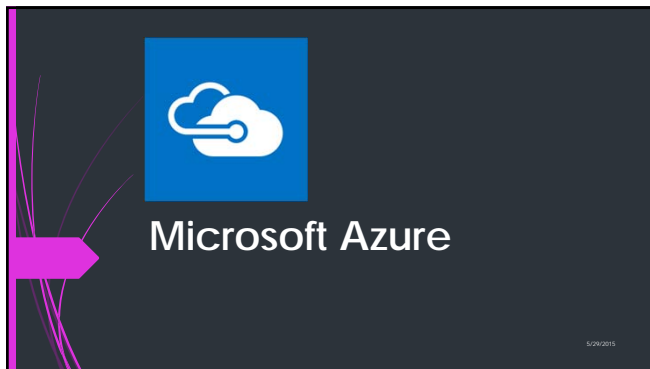
1. 充實基礎程式設計的能力：不論是利用何種平台開發系統，程式設計基礎都是其中最重要的能力。一般學生在初學程式設計時候缺乏範例的指導。因此對於解決問題的能力無法培養。也喪失了學習程式設計的機會。本計畫透過案例教學重新充實學生基礎程式設計的能力。
2. 了解雲端應用程式開發實作：雲端應用為最近流行的技術，任何的應用都被套上雲端以趕上流行。然而就資訊系所畢業的學生而言則必須對雲端的技術有所研究。本計畫配合課程撰寫了真正的雲端平台的教材。學生在本課程中真的實作
3. 了解物聯網基礎：物聯網同時也是最近流行的技術，然而物聯網需要以其他技術整合才可以發揮其最大效用。常見整合物聯網的技術為雲端技術以及大數據分析技術。而透過本計畫的實行可以使學生整合兩項最新流行的技術。

## 陸、結論

本計畫結合了物聯網以及雲端運算兩大主軸技術，對於學習新的技術而言有需要注意的有(1)基礎是否扎實；(2)是否具有範例或是個案探討；(3)是否有整合式的教學案例。本計畫符合以上三大要求。而由於課程學分數的限制，本課程只有兩學分，因此未能有足夠的時間涵蓋大數據分析的議題。因此期望未來有機會可以有這方面的課程。

## 柒、附錄

本計畫預計所開發的教材擷取部分如附錄：



7 環境設置

- 申請 Azure 帳號
- Visual Studio Professional 2013 with Update 4
- Web Platform Installer
  - 安裝 Azure SDK for vs2013

5/29/2015

8 Dreamspark

申請 Dreamspark 帳號  
<https://www.dreamspark.com/>

5/29/2015

9 Dreamspark

5/29/2015

10 Dreamspark

5/29/2015

11 Free Trial

升級為 Azure Free Trial

升級為 Microsoft Azure Free Trial 並取得 \$16,300 美元之 30 天之 Azure 免費試用。您會收到一個新帳戶並繼續執行 30 天。而且在此期間內，您可以使用免費之服務來建置 Azure 方案。

您想要註冊嗎？

注意：升級您的訂閱帳戶，需要升級您的訂閱帳戶，必須為其新增訂閱方式。當 \$16,300 的試用期結束，您的訂閱帳戶將停止，試用期結束後您支付您可以繼續使用服務或選擇新的訂閱帳戶，以您期望的期限繼續使用服務。

以下，您應該選擇好要升級。

訂閱帳戶名稱  
Free Trial

5/29/2015

12 安裝 Visual Studio Professional 2013 with Update 4

Microsoft DreamSpark

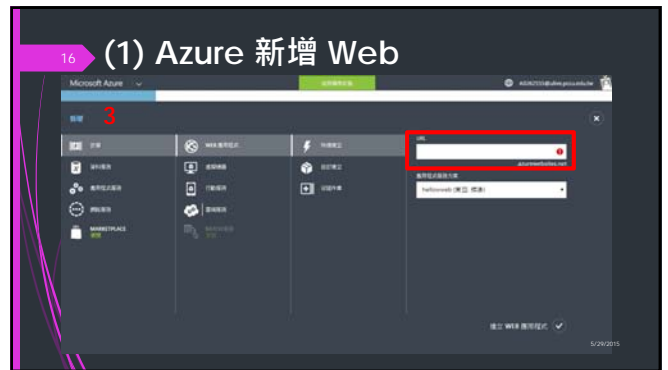
免費。零風險。

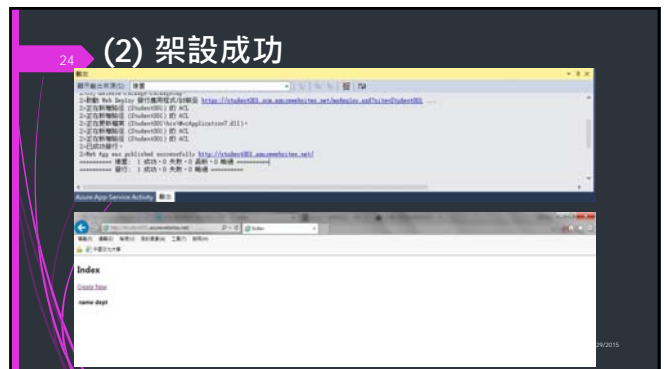
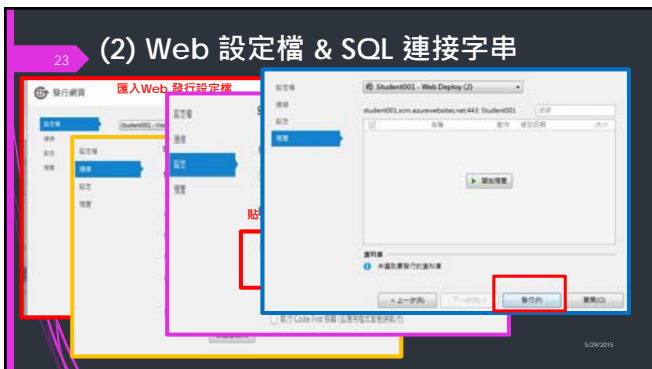
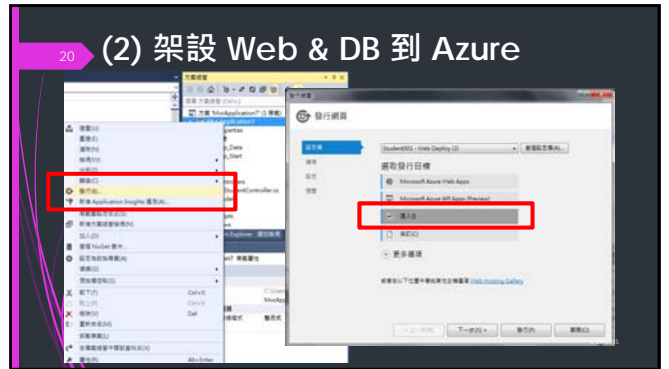
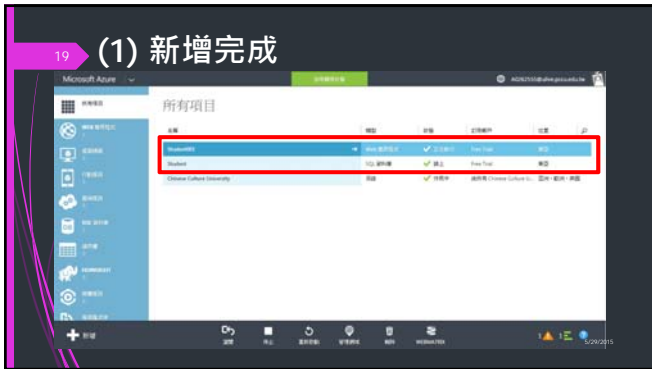
學生免費下載最新 Visual Studio Professional 2013 Update 4

下載

5/29/2015









## Google App Engine for JAVA

2018/1/28

1

## Outline

1. 申請Google Cloud MySQL
2. 使用 MySQL Workbench 連線
3. Eclipse Google Plugin 介紹
4. Eclipse Google Plugin
5. Eclipse建立JAVA網頁
6. 開發簡易登入系統
7. 專案佈署至GAE

2018/1/28

2

## MySQL(免費60天)

1. 以Google帳號 登入 Google Developers Console : <https://console.developers.google.com/project>

2. 點選需要申請MySQL的GAE專案



2018/1/28

3

## MySQL(免費60天)

3. 左邊列的儲存空間 → Cloud SQL → 申請免費試用 → 填寫基本資料以及信用卡資訊



註：一定要輸入信用卡資訊，並且試刷大約30塊以驗證資料正確性

2018/1/28

4

## MySQL(免費60天)

4. 建立Cloud SQL 執行個體
  - 填寫執行個體的ID (註：只能使用小寫字母、數字和連字號。)
  - 區域 (請選擇與此專案相同的區域)
  - 層級 (依需求做選擇)



2018/1/28

5

## MySQL(免費60天)

5. 點擊「顯示進階選項」做進一步的設定
6. 打勾「IPv4位址」：便於之後使用MySQL連線
7. 「允許的網路」：列在這邊的網路才可以進行連線
8. 「App Engine 應用程式 ID」：為此專案的ID，假如沒有或是無法新增，需回去檢查「區域」的部分。

(圖在下一張)

2018/1/28

6

### MySQL(免費60天)

9. 等Google建立完成，點進來可以看到這個畫面

10. 進入「存取控制」修改登入者的資訊

2018/1/28 7

### MySQL(免費60天)

9. 等Google建立完成，點進來可以看到這個畫面

10. 進入「存取控制」修改登入者的資訊

2018/1/28 8

### MySQL(免費60天)

9. 等Google建立完成，點進來可以看到這個畫面

10. 進入「存取控制」修改登入者的資訊

2018/1/28 9

### MySQL(免費60天)

11. 新增MySQL使用者，便於連線時使用。

→「新增使用者」→「填寫使用者名稱」→「密碼」→「新增」

2018/1/28 10

### MySQL(免費60天)

12. 進入 IP 位址，查看 IPv4 Address，接下來要使用 MySQL 連線到 Google Cloud SQL

2018/1/28 11

### MySQL Workbench

1. 下載MySQL Installer  
<https://dev.mysql.com/downloads/windows/installer/>


2. 安裝完成後，開啟MySQL Workbench建立連線

1. Connection Name：此連線的名稱
2. Hostname：連線的IPv4位址
3. Username：輸入使用者名稱
4. 測試連結  
「Test Connection」

2018/1/28 12

### MySQL Workbench

- 輸入該MySQL帳號的密碼 & 測試連結成功的訊息




- 最後按下OK，建立此Connection
- 連結Google Cloud SQL


2018/1/28 13

### MySQL Workbench

- 新增 DataBase，「輸入DB Name」→「按下 Apply」→「Apply」→「Finish」



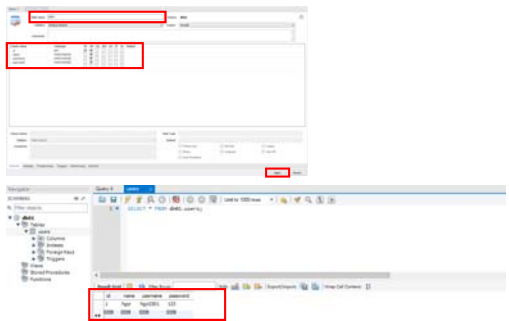
- 建立Table前，先將db01設定為Default Schema



2018/1/28 14

### MySQL Workbench

- 建立Table及欄位&新增一筆資料




2018/1/28 15

### Eclipse Google Plugin

安裝Eclipse Google Plugin

- 可直接在Eclipse執行Google Plugin

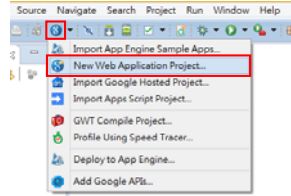
- Eclipse開啟後，從選單按Help → Install New Software
- 在Work with輸入網址→<https://dl.google.com/eclipse/plugin/4.4>
- 勾選下列五項→「Next」→「Next」→勾選I accept...→「Finish」→進行安裝→「OK」→「Restart Now」重新啟動Eclipse



2018/1/28 16

### Eclipse建立JAVA網頁

- 建立New Web Application Project：
  - Google圖示→ New Web Application Project



2018/1/28 17

### Eclipse建立JAVA網頁

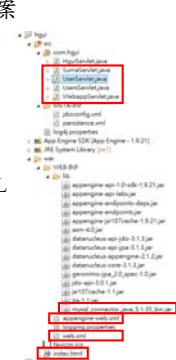
- Project設定：
  - Project name：可自訂與應用程式 ID 不同的名稱
  - Package：
  - Use Google Web Toolkit打勾取消



2018/1/28 18

### 開發簡易登入系統

1. 建立一個簡易登入系統需要八個檔案
  - UserServlet.java — 建立使用者
  - UsersServlet.java — 觀看所有使用者
  - WebappServlet.java — 登入後第一頁做一個加總
  - SumaServlet.java — 呈現加總的數字
  - Mysql\_connector\_java\_5.1.35\_bin.jar — 使用Msql的jdbc來連結Google Cloud SQL
  - appengin-web.xml — 設定環境變數，當專案上傳至GAE後可以正常的讀取SQL
  - web.xml — 網頁路徑轉換設定
  - index.html — 登入畫面



2018/1/28 19

### 開發簡易登入系統


1. 建立一個簡易登入系統需要八個檔案
  - 新增UserServlet.java、UsersServlet.java、WebappServlet.java、SumaServlet.java四個檔案
    - 專案src資料夾 → [com.專案名稱]點選右鍵 → New → File
    - 在「File name」輸入檔案名稱(含副檔名) → Finish。



2018/1/28 20

### 開發簡易登入系統

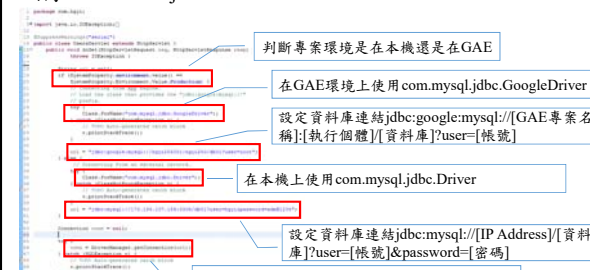
1. 建立一個簡易登入系統需要八個檔案
  - 加入Mysql\_connector\_java\_5.1.35\_bin.jar檔案
    - 網址：<http://dev.mysql.com/downloads/connector/j/>
    - 安裝完後會在Java檔案裡(預設：C:\Program Files (x86)\MySQL\MySQL Connector J)
    - 將jar檔複製起來貼到專案檔裡war → WEB-INF → lib 右鍵 Paste 貼上



2018/1/28 21

### 開發簡易登入系統

2. 編寫程式碼 UsersServlet.java、appengin-web.xml、web.xml、index.html
  - 編寫UsersServlet.java
    - 判斷專案環境是在本機還是在GAE
    - 在GAE環境上使用com.mysql.jdbc.GoogleDriver
    - 設定資料庫連結jdbc:google:mysql://[GAE專案名稱]:[執行個體]/[資料庫]?user=[帳號]
    - 在本機上使用com.mysql.jdbc.Driver
    - 設定資料庫連結jdbc:mysql://[IP Address]/[資料庫]?user=[帳號]&password=[密碼]
    - 將連線字串給 Connection



2018/1/28 22

### 開發簡易登入系統

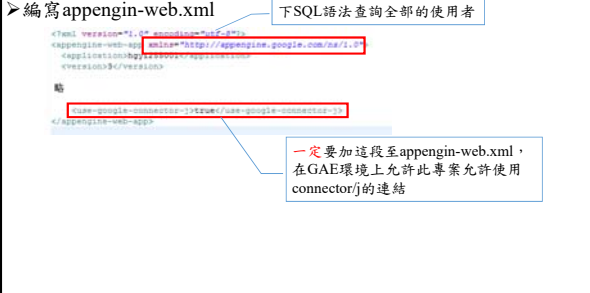
2. 編寫程式碼 UsersServlet.java、appengin-web.xml、web.xml、index.html
  - 編寫UsersServlet.java
    - 下SQL語法查詢全部的使用者
    - 使用rs.next()尋找使用者



2018/1/28 23

### 開發簡易登入系統

2. 編寫程式碼 UsersServlet.java、appengin-web.xml、web.xml、index.html
  - 編寫appengin-web.xml
    - 下SQL語法查詢全部的使用者
    - 一定要加這段至appengin-web.xml，在GAE環境上允許此專案允許使用connector/j的連結



2018/1/28 24

### 開發簡易登入系統

2. 編寫程式碼 UsersServlet.java、appengin-web.xml、web.xml、index.html

► 編寫web.xml

```

18 <web-app>
19 <context-param>
20 <param-name>contextConfigLocation</param-name>
21 <param-value>/WEB-INF/classes/com.hqyl.UsersServlet</param-value>
22 </context-param>
23 <filter>
24 <filter-name>appengineFilter</filter-name>
25 <filter-class>com.google.appengine.api.impl.servlets.AppEngineFilter</filter-class>
26 </filter>
27 <filter-mapping>
28 <filter-name>appengineFilter</filter-name>
29 <url-pattern>/</url-pattern>
30 </filter-mapping>
31 <listener>
32 <listener-class>com.hqyl.UsersServlet</listener-class>
33 </listener>
34 </web-app>

```

2018/1/28

25

### 開發簡易登入系統

2. 編寫程式碼 UsersServlet.java、appengin-web.xml、web.xml、index.html

► 編寫index.html

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html>
4 <!-- The HTML 4.01 Transitional DOCTYPE declaration
5 <!-- above sets at the top of the file will set
6 <!-- the browser's rendering engine into
7 <!-- "Quirks Mode", rendering with quirks and
8 <!-- with a "Standards Mode" header is suggested,
9 <!-- but may lead to some differences in layout. -->
10 </html>
11 <head>
12 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
13 </head>
14
15 <body>
16 <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">
17 <div style="text-align: center; border-bottom: 1px solid black; padding-bottom: 5px;">
18 <span style="font-weight: bold; font-size: 1.2em; display: inline-block; margin-right: 10px;">登入介面
19 <input type="text" value="帳號" style="width: 100%; border: none; border-bottom: 1px solid black; padding: 2px 5px; margin-bottom: 5px;" />
20 <input type="password" value="密碼" style="width: 100%; border: none; border-bottom: 1px solid black; padding: 2px 5px; margin-bottom: 5px;" />
21 <input type="button" value="登入" style="width: 100%; border: 1px solid black; padding: 5px; margin-bottom: 5px;" />
22 <input type="button" value="註冊" style="width: 100%; border: 1px solid black; padding: 5px; margin-bottom: 5px;" />
23 </div>
24 </div>
25 </body>
26 </html>

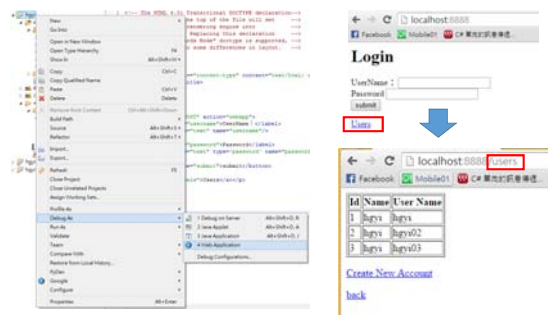
```

2018/1/28

26

### 開發簡易登入系統

3. 對著專案右鍵 → Debugs As → Web Application



2018/1/28

27

### 開發簡易登入系統

2. 編寫程式碼UserServlet.java、WebappServlet.java、SumaServlet.java、web.xml

► 編寫UserServlet.java

```

1 package com.hqyl;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 public class UserServlet extends HttpServlet {
12     private static final long serialVersionUID = 1L;
13
14     // The doGet method of the HttpServlet class implements
15     // servlet <code>GET</code> requests.
16     // You can now return data from the HTTP request.
17     // You should return the result as a character stream.
18     protected void doGet(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         response.setContentType("text/html");
21         PrintWriter out = response.getWriter();
22         out.println("<h1>登入成功</h1>");
23         out.println("<br/>");
24         out.println("<h2>登入失敗</h2>");
25         out.println("<br/>");
26         out.println("<h3>登入失敗</h3>");
27         out.println("<br/>");
28         out.println("<h4>登入失敗</h4>");
29         out.println("<br/>");
30         out.println("<h5>登入失敗</h5>");
31         out.println("<br/>");
32         out.println("<h6>登入失敗</h6>");
33         out.println("<br/>");
34         out.println("<h7>登入失敗</h7>");
35         out.println("<br/>");
36         out.println("<h8>登入失敗</h8>");
37         out.println("<br/>");
38         out.println("<h9>登入失敗</h9>");
39         out.println("<br/>");
40         out.println("<h10>登入失敗</h10>");
41     }
42 }

```

2018/1/28

28

### 開發簡易登入系統

2. 編寫程式碼UserServlet.java、WebappServlet.java、SumaServlet.java、web.xml

► 編寫UserServlet.java

```

1 package com.hqyl;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 public class UserServlet extends HttpServlet {
12     private static final long serialVersionUID = 1L;
13
14     // The doGet method of the HttpServlet class implements
15     // servlet <code>GET</code> requests.
16     // You can now return data from the HTTP request.
17     // You should return the result as a character stream.
18     protected void doGet(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         response.setContentType("text/html");
21         PrintWriter out = response.getWriter();
22         out.println("<h1>登入成功</h1>");
23         out.println("<br/>");
24         out.println("<h2>登入失敗</h2>");
25         out.println("<br/>");
26         out.println("<h3>登入失敗</h3>");
27         out.println("<br/>");
28         out.println("<h4>登入失敗</h4>");
29         out.println("<br/>");
30         out.println("<h5>登入失敗</h5>");
31         out.println("<br/>");
32         out.println("<h6>登入失敗</h6>");
33         out.println("<br/>");
34         out.println("<h7>登入失敗</h7>");
35         out.println("<br/>");
36         out.println("<h8>登入失敗</h8>");
37         out.println("<br/>");
38         out.println("<h9>登入失敗</h9>");
39         out.println("<br/>");
40         out.println("<h10>登入失敗</h10>");
41     }
42 }

```

2018/1/28

29

### 開發簡易登入系統

2. 編寫程式碼UserServlet.java、WebappServlet.java、SumaServlet.java、web.xml

► 編寫UserServlet.java

```

1 package com.hqyl;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 public class UserServlet extends HttpServlet {
12     private static final long serialVersionUID = 1L;
13
14     // The doGet method of the HttpServlet class implements
15     // servlet <code>GET</code> requests.
16     // You can now return data from the HTTP request.
17     // You should return the result as a character stream.
18     protected void doGet(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         response.setContentType("text/html");
21         PrintWriter out = response.getWriter();
22         out.println("<h1>登入成功</h1>");
23         out.println("<br/>");
24         out.println("<h2>登入失敗</h2>");
25         out.println("<br/>");
26         out.println("<h3>登入失敗</h3>");
27         out.println("<br/>");
28         out.println("<h4>登入失敗</h4>");
29         out.println("<br/>");
30         out.println("<h5>登入失敗</h5>");
31         out.println("<br/>");
32         out.println("<h6>登入失敗</h6>");
33         out.println("<br/>");
34         out.println("<h7>登入失敗</h7>");
35         out.println("<br/>");
36         out.println("<h8>登入失敗</h8>");
37         out.println("<br/>");
38         out.println("<h9>登入失敗</h9>");
39         out.println("<br/>");
40         out.println("<h10>登入失敗</h10>");
41     }
42 }

```

2018/1/28

30

### 開發簡易登入系統

2. 編寫程式碼UserServlet.java、WebappServlet.java、SumaServlet.java、web.xml

➢編寫WebappServlet.java

```
1 package com.hgyis;
2 @SuppressWarnings("unchecked")
3 public class WebappServlet extends HttpServlet {
4     private void doGet(HttpServletRequest req, HttpServletResponse resp)
5         throws ServletException, IOException {
6         String username = req.getParameter("username");
7         String password = req.getParameter("password");
8         String url = null;
9         if (StringUtil.isEmpty(username) ||
10             StringUtil.isEmpty(password)) {
11             resp.setContentType("text/html");
12             resp.getWriter().println("請輸入帳號密碼!");
13             return;
14         }
15         User user = UserUtil.getUserByUsername(username);
16         if (user == null) {
17             resp.setContentType("text/html");
18             resp.getWriter().println("帳號不存在!");
19             return;
20         }
21         boolean isValid = UserUtil.isValid(username, password);
22         if (!isValid) {
23             resp.setContentType("text/html");
24             resp.getWriter().println("密碼錯誤!");
25             return;
26         }
27         HttpSession session = req.getSession(true);
28         session.setAttribute("user", user);
29         resp.setContentType("text/html");
30         resp.getWriter().println("登入成功!");
31     }
32 }
```

2018/1/28

31

### 開發簡易登入系統

2. 編寫程式碼UserServlet.java、WebappServlet.java、SumaServlet.java、web.xml

➢編寫WebappServlet.java

```
33 String contextPath = req.getContextPath();
34 String url = contextPath + "/webapp";
35 HttpServletResponse res = com.hgyis.servlet.WebappServletUtil.getResponse(res);
36 res.setContentType("text/html");
37 while (true) {
38     String username = req.getParameter("username");
39     String password = req.getParameter("password");
40     if (StringUtil.isEmpty(username) ||
41         StringUtil.isEmpty(password)) {
42         res.getWriter().println("請輸入帳號密碼!");
43         continue;
44     }
45     User user = UserUtil.getUserByUsername(username);
46     if (user == null) {
47         res.setContentType("text/html");
48         res.getWriter().println("帳號不存在!");
49         continue;
50     }
51     boolean isValid = UserUtil.isValid(username, password);
52     if (!isValid) {
53         res.setContentType("text/html");
54         res.getWriter().println("密碼錯誤!");
55         continue;
56     }
57     HttpSession session = req.getSession(true);
58     session.setAttribute("user", user);
59     res.setContentType("text/html");
60     res.getWriter().println("登入成功!");
61 }
```

2018/1/28

32

### 開發簡易登入系統

2. 編寫程式碼UserServlet.java、WebappServlet.java、SumaServlet.java、web.xml

➢編寫SumaServlet.java

```
1 package com.hgyis;
2 import java.io.IOException;
3 @SuppressWarnings("serial")
4 public class SumaServlet extends HttpServlet {
5     public void doGet(HttpServletRequest req, HttpServletResponse resp)
6         throws ServletException, IOException {
7         PrintWriter pw = resp.getWriter();
8         resp.setContentType("text/plain");
9         String num1 = req.getParameter("num1");
10         String num2 = req.getParameter("num2");
11         pw.println("Total: " + (Integer.parseInt(num1) + Integer.parseInt(num2)));
12     }
13 }
```

2018/1/28

33

### 開發簡易登入系統

2. 編寫程式碼UserServlet.java、WebappServlet.java、SumaServlet.java、web.xml

➢編寫web.xml

```
<web-app>
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/classes</param-value>
</context-param>
<listener>
<listener-class>com.hgyis.WebappServlet</listener-class>
</listener>
<listener>
<listener-class>com.hgyis.WebappServlet</listener-class>
</listener>
<listener>
<listener-class>com.hgyis.WebappServlet</listener-class>
</listener>
<listener>
<listener-class>com.hgyis.WebappServlet</listener-class>
</listener>
<listener>
<listener-class>com.hgyis.SumaServlet</listener-class>
</listener>
<listener>
<listener-class>com.hgyis.SumaServlet</listener-class>
</listener>
<listener>
<listener-class>com.hgyis.SumaServlet</listener-class>
</listener>
</web-app>
```

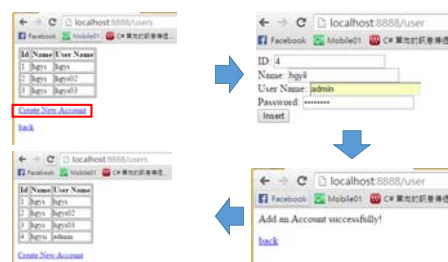
2018/1/28

34

### 開發簡易登入系統

2. 編寫程式碼UserServlet.java、WebappServlet.java、SumaServlet.java、web.xml

➢UserServlet.java 畫面



2018/1/28

35

### 開發簡易登入系統

2. 編寫程式碼UserServlet.java、WebappServlet.java、SumaServlet.java、web.xml

➢WebappServlet.java 畫面



➢SumaServlet.java 畫面



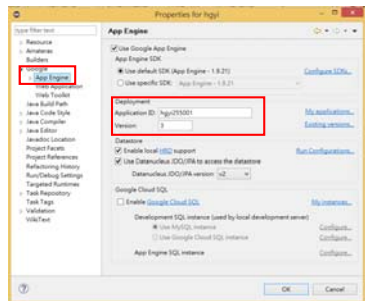
2018/1/28

36



## 專案佈署至GAE

1. 先設定GAE application屬性及version屬性  
 ▶專案右鍵 → Properties → Google → App Engine

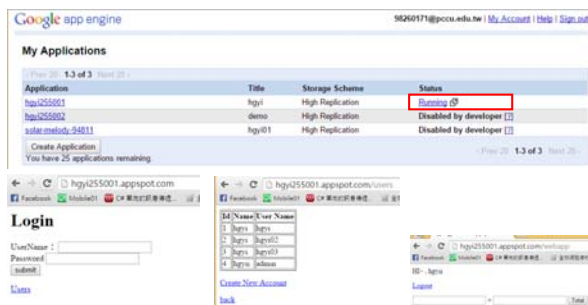


2018/1/28

37

## 專案佈署至GAE

2. Google App Engine平台 :  
 • 輸入Google帳號登入平台 <https://appengine.google.com>



2018/1/28

38