

# 逢甲大學學生報告 ePaper

## Aok - 簡易上手開發的 Web Framework

## Aok - A web Framework for easy development

作者：施冠彰, 楊依辰, 戎宥杰, 黃子騰

系級：資訊三丙

學號：D0745925 , D0746490, D0783026, D0708068

開課老師：薛念林

課程名稱：物件導向軟體工程

開課系所：資訊工程學系

開課學年：109 學年度 第 1 學期



## 中文摘要

### (1) 目的

在撰寫 java 伺服器時，發現並沒有簡單、快速、簡潔建成的 http 服務，我們認為現有的 Java Web Framework 時常會因為需要有充足的功能性，導致框架最終變得肥胖而難以維護，因此我們希望能製造出類似 Koa 的框架，使得在 java 上開發 http 服務能夠輕量化更加簡潔快速。

### (2) 撰寫與方法

首先我們將架構使用 MVC 架構，將 web 框架所需的 Model 與 View、Controller 的部份分開處理，支援多數常見 HTTP Method 的 Request 與 Response，並且設計讓開發者可以很輕易的讀取 request 的資料，並且很輕易的發送 response。不需去處理中間資料型態、資料串流等問題，並且能夠快速建立 Web Server，只要一行指令即可處理，使用 Freemarker Template Engine 來處理 HTML View，可以簡易的開發動態網頁。

### (3) 結果

我們建構出了簡易的、快速的 web server 框架 並且使用框架打造 Aok 框架官網以及班級管家兩項應用服務，雖然目前框架的功能覆蓋沒有很全面，但我們未來會將 Aok 打造成更加完善的框架。

**關鍵字：**JAVA、MVC 架構、框架、網頁伺服器、動態網頁

## Abstract

### Object

We found that Java language didn't have a simple and rapid HTTP server for their user when we set up a server in Java. Currently, because Java Web Framework often need so many sufficient functions that making the size of framework became too large to maintain difficulty. Therefore, we hope to produce a useful frame similar to the KOA, and help our users develop a Java's HTTP server in a light and easy way.

### Design and solution

First, we use the MVC structure to process Model, View and View separately. And our framework supports many common HTTP method like Request and Response, these methods lets our user request and response their data easily. When using our framework, the developer won't deal with the problems of data type or data stream, and you just enter a simple instruction that could establish a Web Server. Finally, we use the Freemarker Template Engine to deal with the problem of HTML View, it help the developer to develop the dynamic Web page.

### Result

We invented a unique web server framework and using this framework to design our AOK's official website and a Web page about the housekeeper of class. Although our framework's function didn't cover overall. But I believed that due to our efforts, AOK framework must be advance in future.

## Keyword

dynamic website, framework, JAVA programming language, MVC, web server

## 目 次

|                 |    |
|-----------------|----|
| 一、框架功能 .....    | 4  |
| 二、UML 圖 .....   | 4  |
| 三、設計原理 .....    | 5  |
| 四、測試與執行畫面 ..... | 5  |
| 五、應用程式 .....    | 11 |

## 一、框架功能

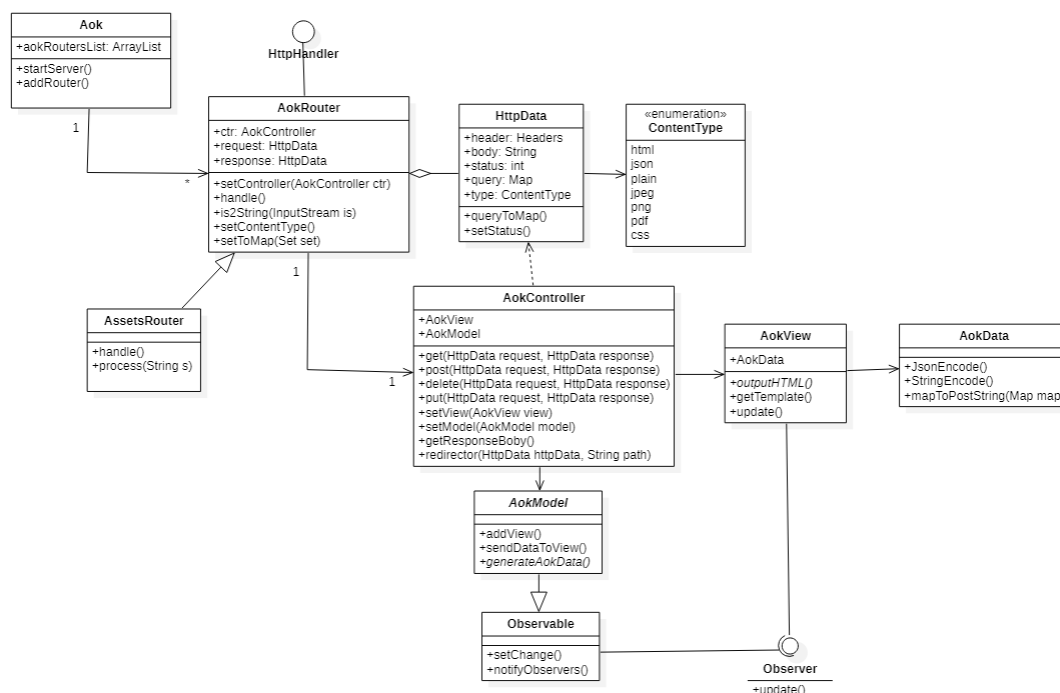
Aok 希望可以提供給開發者一個簡易上手、快速部署、快速開發的一個框架。我們認為現有的 Java Web Framework 框架時常會因為需要有充足的功能性，導致框架最終變得肥胖而難以維護，甚至在效能也大打折扣。因此，我們參考了來自各個程式語言輕量 Web Framework，來做出一個在 Java 上的輕量版 Web Framework。

以下是我們框架實現的功能：

1. 快速建立 Web Server，只要生成 Aok 物件並且 Aok.start(8080)即可建立。
2. 支援多數常見 HTTP Method 的 Request 與 Response，並且設計讓開發者可以很輕易的讀取 req 的資料，並且很輕易的發送 res。不需去處理中間資料型態、資料串流等問題。
3. 採用 MVC 架構，除了分開處理 Model 與 View，在 Controller 的部分，特別拆分成 Router 與 Controller，以此來更有彈性的掌控 Router 定義，並且可以在 Controller 的開發上，有更好的分工體驗。
4. 使用 Freemaker Template Engine 來處理 HTML View，開發者可以更簡易的開發動態網頁。

## 二、UML 圖

### Framework



### 三、設計原理

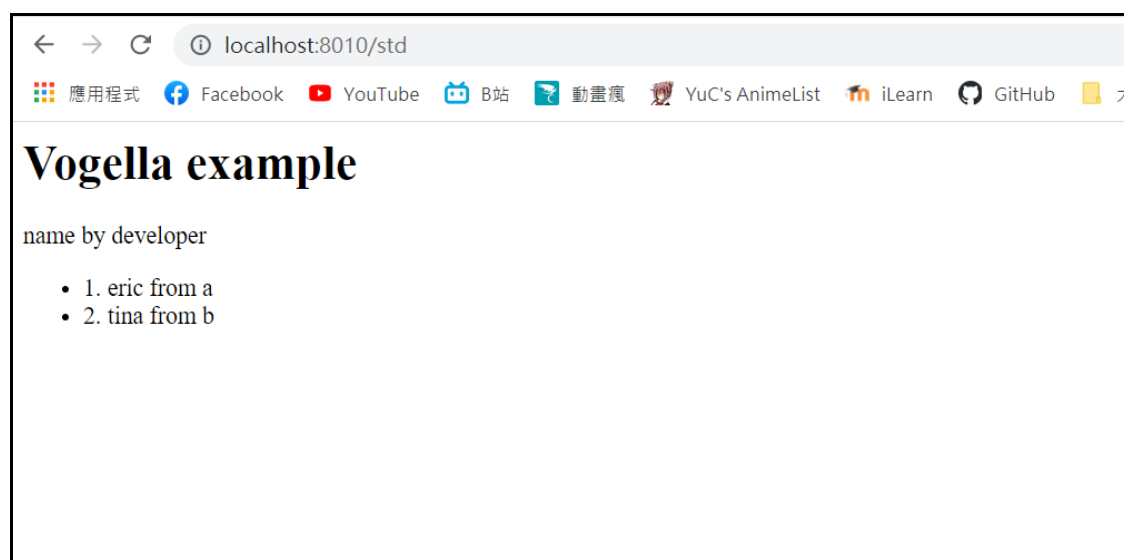
1. 採用 MVC 架構，Model 為 AokModel 介面，使用者可以實作這個介面，來自行定義資料結構與商業邏輯。View 的部分為 AokView，結合 Freemaker 的模板引擎並且可以注入 Model 的資料，獨立處理動態網頁的實作。Controller 的部分分為兩個：AokRouter、AokController，前者為處理 web server router 的工作，可以自行定義 URI 以及其相對應的 controller 物件。而後者即是 Controller，負責處理 client 傳入的 req，中間 model 的資料處理、商業邏輯，最後回傳 res 給 Aok server，回傳給 client。
2. 應用 Observer，由 AokView 實作 Observer 的 update method，AokModel 繼承 Observable，利用 sendDataToView() 將資料即時回傳。如此一來當 data 一有所變更時，可以及時通知 view 做更新。

### 四、測試與執行畫面

```
public class TestMain {  
  
    public static void main(String[] args) throws IOException {  
  
        Aok aok = new Aok(8010);  
        AokRouter testRouter = new AokRouter();  
        TestView tv = new TestView("helloworld.ftl");  
        AokController testStdCtr = new TestStdController();  
        AokModel testModel = new TestDataModel();  
        testModel.addView(tv);  
        testRouter.setController(testStdCtr);  
        testStdCtr.setView(tv);  
        testStdCtr.setModel(testModel);  
        aok.addRouter("/std", testRouter);  
        aok.startServer();  
    }  
}
```

```
public class TestView extends AokView {
    public TestView(String templateFileName) {
        super(templateFileName);
    }
    @Override
    public String outPutHTML() {
        input.put("title", "Vogella example");
        input.put("exampleObject", this.dataObj);
        return getTemplate();
    }
}
```

由 TestView 繼承 AokView，實作 outPutHTML()，再在 TestMain 中設定好 Router、Controller、Model、View 後，就能輕鬆架好一個 Server。



## AoK(Class)

Aok 為整個 Web Server 的核心，負責處理與底層 httpServer 套件溝通、設定、註冊 Router...等工作。開發者只需要在 main 裡面建立一個 Aok 物件，就可以啟動伺服器了。

- **Constructor Detail**

|      |  |
|------|--|
| void | <b>Aok(int port)</b><br>一切準備就緒後，啟動伺服器。<br>建立物件的同時，也會像底層 httpserver 做設定。並且將 Web Server 開啟的 port 傳入。 |
|------|--|

- **Method Detail**

|      |   |
|------|---|
| void | <b>startServer()</b><br>一切準備就緒後，啟動伺服器。  |
| void | <b>addRouter(String path, AokRouter handler)</b><br>註冊新的 Router 到 Aok Server 中。其中，path 為此路由的 URI，例如「/」、「/student」、「/student/scord」等值皆為合法輸入值。handler 則是要傳入 AokRouter 作為此路由的處理物件。 |

- **範例**

```
// 初始化伺服器，輸入 port 號碼
Aok aok = new Aok(8010);

// 註冊 Router
aok.addRouter("/", new AokRouter());

// 啟動伺服器
aok.startServer();
```

### **AokRouter(class, implements HttpHandler)**

主要負責每個路由的資料處理，當 client 傳送請求到某個路由來，將由此路由先前註冊的 AokRouter 來處理後續如 GET、POST 等 Http Method 以及呼叫 AokController 來處理傳入請求。

- **Method Detail**

|      |   |
|------|---|
| void | <b>handle(HttpExchange exchange)</b><br>為實作 HttpHandler 的方法，當路由接收到請求的處理演算法在此方法中定義。其中使用 AokController 作為 Strategy。其中的 HTTP Method 演算法由 AokController 提供。 |
| void | <b>setController(AokController ctr)</b><br>設定 Controller 屬性。  |
| void | <b>setContenttype()</b><br>根據 response 屬性，可以知道此次返回的 Content Type 為何，並且賦予適當的 header，讓 Server 可以成功執行。   |



|                           |   |
|---------------------------|---|
| String                    | <b>is2string(InputStream is) throws IOException</b><br>把 request 中的 body 由 InputStream 轉為 String。 |
| Map<String, List<String>> | <b>setToMap(Set set)</b><br>把 request header 中的 entrySet()轉換成 map 的形式，方便後續操作。                     |

### AssetsRouter(class, implements HttpHandler, extends AokRouter)

主要負責處理靜態檔案，檔案路徑預設在該 project 下層的資料夾下 (Aok/src/Assets/檔案名稱)，像是 html、css、jpg....等型態的檔案。

- Method Detail

|        |   |
|--------|---|
| void   | <b>handle(HttpExchange exchange) throws IOException</b><br>在 HttpHandler GET 到靜態檔案時做處理，並將資料 response 給 Server |
| String | <b>process(String s)</b><br>負責做靜態檔案的字串處理  |

### HttpData(class)

儲存所有 response 和 request 得到的資料(ex.header、body)。

- Method Detail

|                    |  |
|--------------------|--|
| Map<String,String> | <b>queryToMap()</b><br>把接收到的 query 轉換成 map 型態。 |
| void               | <b>setStatus(int s)</b><br>設定 status 的值。       |

### AokController(class)

主要負責處理 GET、POST、PUT、DELETE，這些 HttpMethod 被 client 端呼叫時所需做的動作。

- Constructor Detail

|      |   |
|------|---|
| void | <b>AokController(AokModel model, AokView view)</b><br>給定 view 和 model 初值。 |
|------|---|

- Method Detail

|          |   |
|----------|---|
| HttpData | <b>Get(HttpData request, HttpData response)</b><br>處理被呼叫時須執行的動作，使用者可以覆寫此 function。    |
| HttpData | <b>Post(HttpData request, HttpData response)</b><br>處理被呼叫時須執行的動作，使用者可以覆寫此 function。   |
| HttpData | <b>Delete(HttpData request, HttpData response)</b><br>處理被呼叫時須執行的動作，使用者可以覆寫此 function。 |
| HttpData | <b>Put(HttpData request, HttpData response)</b><br>處理被呼叫時須執行的動作，使用者可以覆寫此 function。    |
| void     | <b>setView(AokView v)</b><br>設定 view 的值。  |
| void     | <b>setModel(AokModel m)</b><br>設定 model 的值。   |
| void     | <b>redirector(HttpData httpData, String path)</b><br>用來將網頁重新導向。                       |

### AokModel(class)

處理相關在網頁上所需要的相關 Data Type 以及商業邏輯，實作 observable 被 View 所觀察。

- Method Detail

|        |   |
|--------|---|
| void   | <b>addView(AokView view)</b><br>將觀察者 View 與 Model 綁定。                     |
| void   | <b>sendDataToView()</b><br>當 Model 更動時透過 sendDataToView 呼叫 View 的 update。 |
| Object | <b>generateAokData()</b><br>生成一個 AokData 物件。                              |

## AokView(class)

呼叫使用者定義的 html 並且透過 out 將 html 與 AokModel 做合並輸出，並且實作 Observer 的 update。

- Constructor Detail

|      |  |
|------|--|
| void | <b>AokView(String templateFileName)</b><br>設定文件格式跟 templateFileName。 |
|------|--|

- Method Detail

|        |  |
|--------|--|
| String | <b>getTemplate()</b><br>拿取 template 資料。                        |
| void   | <b>update(Observable o, Object arg)</b><br>當 Module 更動時，呼叫的函式。 |
| String | <b>outPutHTML()</b><br>負責把 Model 跟 View 融合並實作出來。               |

## AokData(class)

主要負責傳輸資料編碼與解碼的動作。

- Method Detail

|        |  |
|--------|--|
| String | <b>mapToPostString()</b><br>把 POST 接收到的 map 資料轉換成可處理的 String 型態。 |
| String | <b>jsonEncode()</b><br>把使用者使用到的資料轉換成 json 型態。                    |
| String | <b>StringEncode()</b><br>把使用者使用到的資料轉換成 String 型態。                |

## ContentType(enumeration)

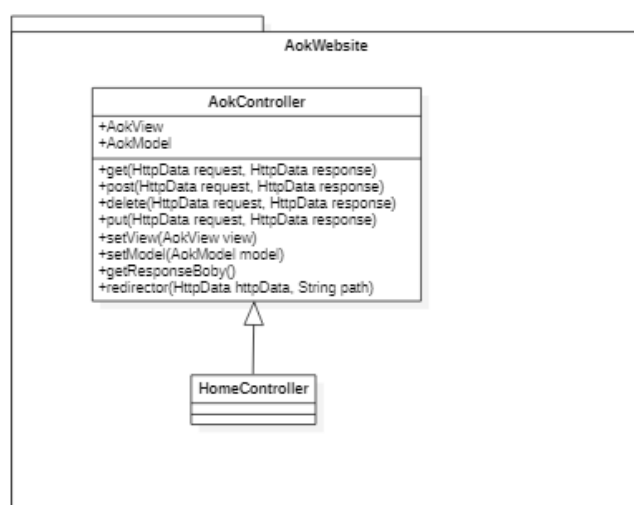
宣告各個可能用到的檔案的 content type。

## 五、應用程式

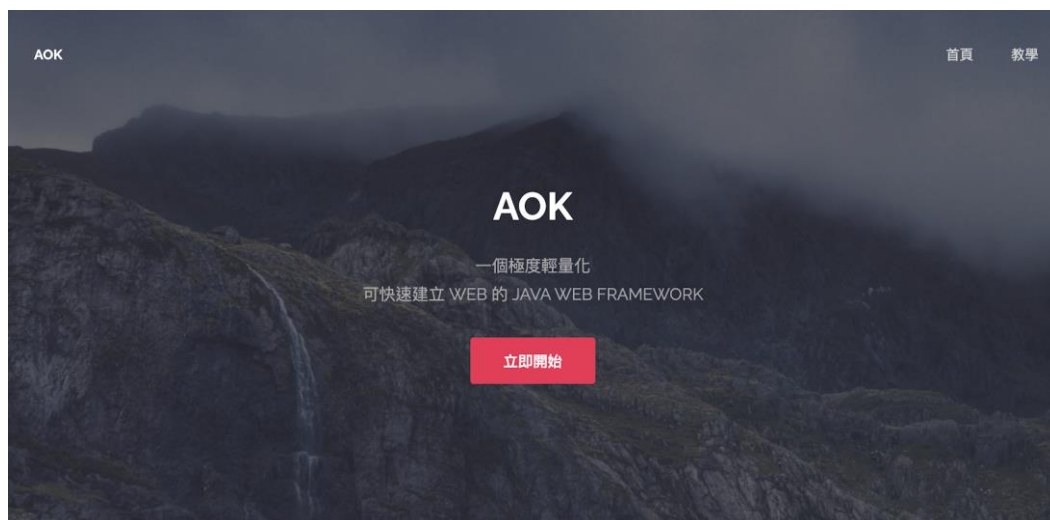
兩個皆為使用 Aok 來開發，Aok 框架網站為靜態網頁，Class Manager 則是動態網頁。

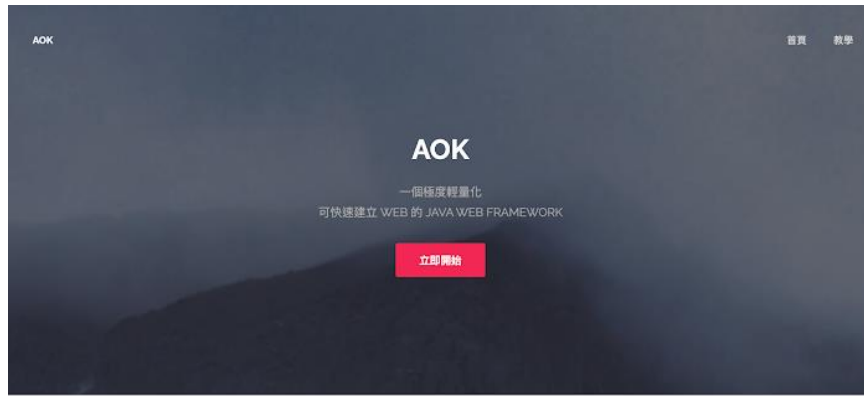
### 1. Aok 框架官網

建構一個簡單精美的使用手冊，除了介紹外還有詳細的教學，讓使用者可以快速學會 Aok 框架的操作。



### 實作結果

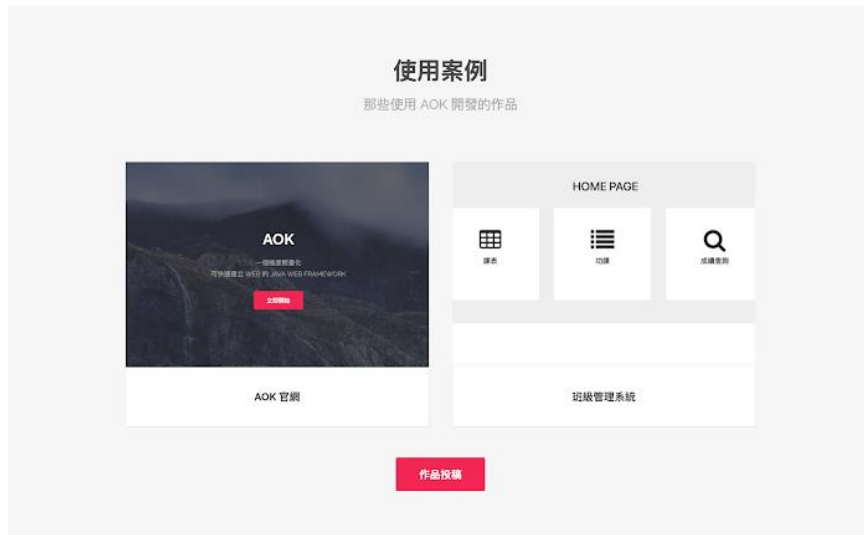




### 為什麼是「極度輕量化」 框架滿街跑，為何我們比較優秀？

市面上的 Web 框架很多，支援 Java 語言的也高山流水，但牠們都有一些通病，就是為了彈性，為了高擴充性，造成框架龐大、不易快速開發、不能理解。

Aok 就不一樣，我們的出發點是希望開發者能在一個下午可以寫一個功能管理系統出來（相信吧，我們測試過了，沒問題）。我們希望開發者可以快速上手這個框架，並且可以在很短的時間，就能開發出屬於自己的 APP 服務、動態網頁。從這一來，開發者可以更專注在使用者需求上，而不是框架怎麼用。



### 框架特色 給你用 AOK 的四個理由

**框架極度輕量**  
少於半打文件，輕鬆安裝與使用  
如此一來，未來如果有什麼 Bug 或者需要新增功能、擴充功能，都可以很輕鬆的更新框架。甚至，你也可以一起來維護



**簡易上手**  
一個下午就能學會框架全部  
一般學習新框架，要花多少時間？三天？一個星期？一個月？這可以跟你保證，學習 Aok，你只需要一個下午，甚至更短，就可以成為 Aok 高手了

**快速開發**  
一個下午開發出管理系統及小工具  
經過我們的實測，只要充分使用框架，一個下午就可以開發一個功能管理系統，外加部署。讓開發者少一點時間寫程式，多一點時間給創意、需求



**MVC架構**  
MODEL - VIEW - CONTROLLER 多人開發更舒適  
大學學過更多人開發？別擔心，使用 Aok，你可以放心的多人開發，將前端的部分完全分開，不互相干擾，未來調整也相當容易。git merge 就搞定



## 快速開始

只需要短短的時間，就可以學會AOK框架，實現你的作品與專題



AOK - 快速建個伺服器



AOKROUTER



AOKCONTROLLER



AOKVIEW



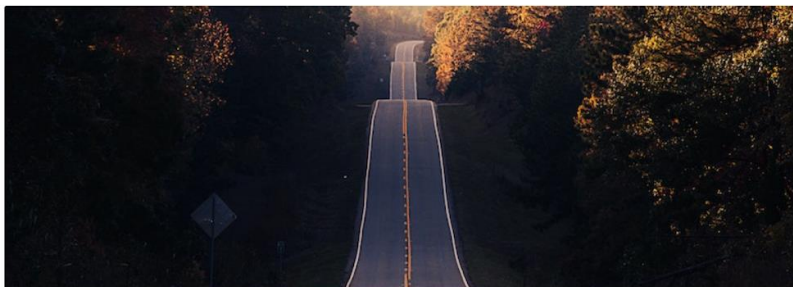
AOKMODEL



AOKDATA

## AOKCONTROLLER

重新掌握你的框架



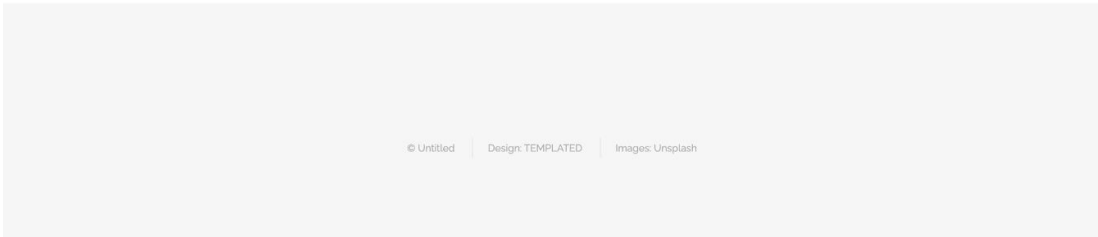
主要負責處理 GET、POST、PUT、DELETE，這些 HttpMethod 被 client 端呼叫時所做的動作。

建構子詳細定義：

| 建構子   | 說明                  |
|---|---------------------|
| AokController(AokModel model, AokView view) | 給定 view 和 model 初值。 |

方法詳細定義：

| 方法   | 說明                              |
|--|---------------------------------|
| Get()                                      | 處理被呼叫時須執行的動作，使用者可以覆寫此 function。 |
| Post()                                     | 處理被呼叫時須執行的動作，使用者可以覆寫此 function。 |
| Delete()                                   | 處理被呼叫時須執行的動作，使用者可以覆寫此 function。 |
| Put()                                      | 處理被呼叫時須執行的動作，使用者可以覆寫此 function。 |
| setView(AokView v)                         | 設定 view 的值。                     |
| setModel(AokModel m)                       | 設定 model 的值。                    |
| redirector(HttpData httpData, String path) | 用來將網頁重新導向。                      |

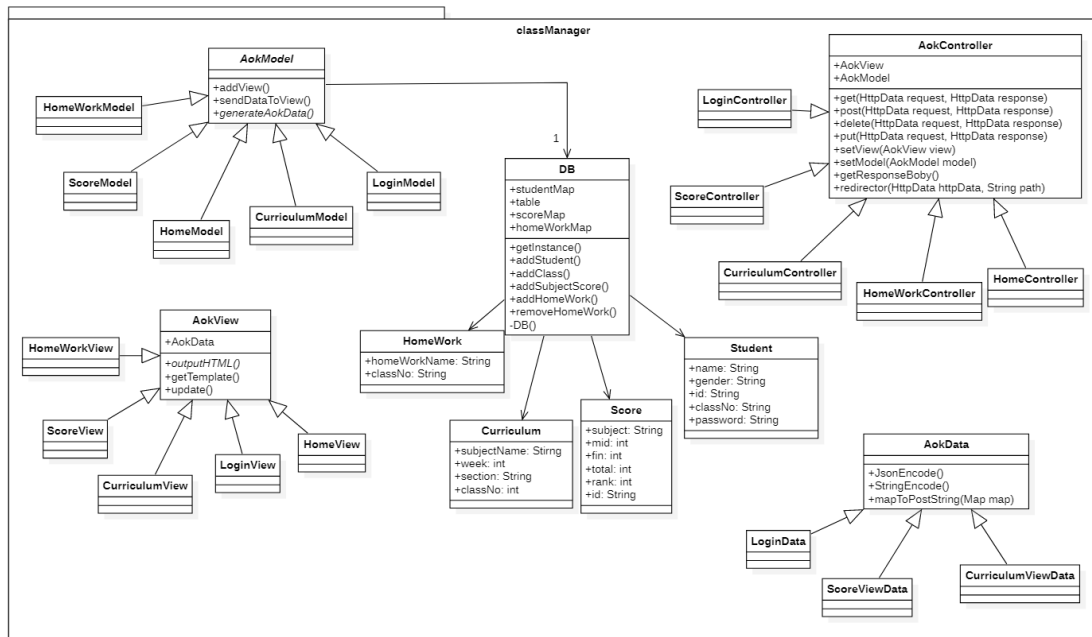


© Untitled | Design: TEMPLATED | Images: Unsplash

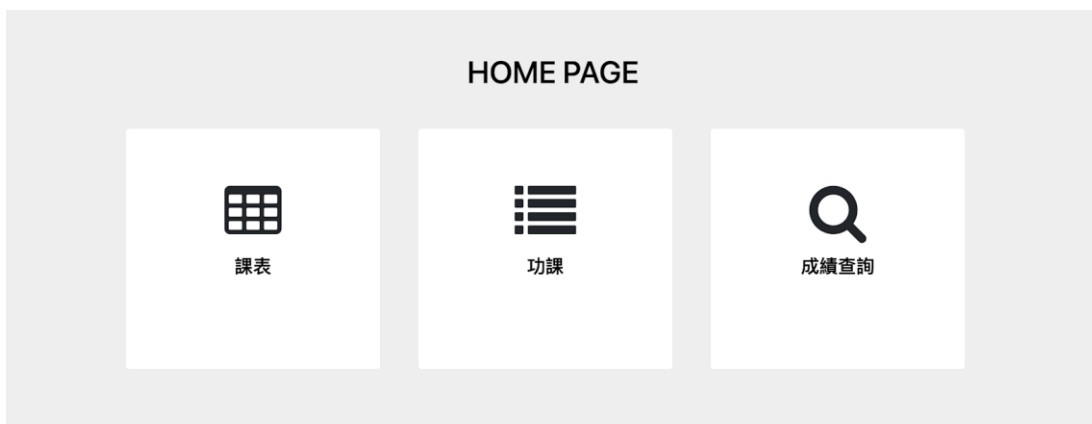
## 2. Class Manager

讓 Aok 接上 DB，讓資料儲存在資料庫中。並且實作班級管理的相關功能：顯示課表、作業查詢、成績查詢等一系列的功能。並且皆透過 Aok Web Framework 來實作。

這裡的 DB class 採用 Singleton 設計方法，用 getInstance() 判斷是否為唯一物件。



實作結果





### 課表

|     |   |
|-----|---|
| 星期一 | ▼ |
| 星期二 | ▼ |
| 星期三 | ▼ |
| 星期四 | ▼ |
| 星期五 | ▼ |

### 課表

|     |          |
|-----|----------|
| 星期一 | ▼        |
| 星期二 | ▼        |
| 1   |          |
| 2   | 物件導向軟體工程 |
| 3   | 物件導向軟體工程 |
| 4   |          |
| 5   |          |
| 6   |          |
| 7   |          |
| 8   |          |
| 9   |          |
| 10  |          |
| 星期三 | ▼        |
| 星期四 | ▼        |
| 星期五 | ▼        |

### 功課列表

- 物件導向專題
- Design Pattern
- 程式設計
- iOS

更新清單

### 成績查詢

**Student Information**

姓名：施冠彰  
學號：eric  
班級：資訊三丙

| 編號 | 科目名稱     | 期中考成績 | 期末考成績 | 學期總成績 | 排名  |
|----|----------|-------|-------|-------|-----|
| 0  | 習題       | 100   | 100   | 100   | 100 |
| 1  | 物件導向程式設計 | 100   | 100   | 100   | 100 |
| 2  | 程式設計     | 100   | 100   | 100   | 100 |
| 3  | 設計       | 100   | 100   | 100   | 100 |

|     |      |         |
|-----|------|---------|
| 系排名 | 累計系排 | 累計系排百分比 |
| 1   | 1    | 1%      |

Class Manager  
Login Page  
Login or register from here to access.

User Name

Password

Login

## 參考文獻

Apache FreeMarker. (2021, February 17). *What is Apache FreeMarker™?* Apache FreeMarker™. <https://freemarker.apache.org/>

Lars Vogel, (c) 2012 - 2020 vogella GmbH. (2012). *Using FreeMarker templates (FTL)- Tutorial*. Vogella.com.

<https://www.vogella.com/tutorials/FreeMarker/article.html>

Tibor. (2012, July 24). *How to obtain the query string in a GET with Java HttpServer/HttpExchange?* Stack Overflow.

<https://stackoverflow.com/questions/11640025/how-to-obtain-the-query-string-in-a-get-with-java-httpserver-httpexchange>

Mana, U. (2020, January 11). *A Simple HTTP Server in Java*. Dzone.com; DZone.

<https://dzone.com/articles/simple-http-server-in-java>