

# 資深Java微服務工程師實作試題

開發人員:	
Assignment日期:	
Assignment 提交日期:	
Github URL:	頁將程式碼上傳到 並將網址提供於此欄位

## 一、引用相關技術說明

本試題實作必須使用 Spring, Hibernate, junit, Maven 及 Github 來完成本試題。  
相關技術、框架說明參考「已提供程式碼說明」一節。

## 二、概述

在本試題你必須實作數個API，這些API提供服務來操作學生系統，此系統為multiple tiers，分別為 Persistence Layer、Service Layer 及 API Layer。

你必須依照不同的Layer 實作已提供的 Java interfaces，如：DAO, Service，並依照需求完成O-R Mapping及RESTful API，有關 Java interface及RESTful API 參考「已提供程式碼說明」一節。

## 三、案例說明

本試題要實作的學生系統是一個非常簡單系統，其目的主要用來記錄學生、課程及課程修業成績，並且提供相關的Restful API 以供調用。

一個學生可以參加多個課程且每一個課程都一定會有一個修業成績的評等，其修業成績的評等有 A, B, C, D。

為使本試題簡單明瞭，此學生系統並無時間面向的需求，例如：課程的時間、學年....等等，資料模型參考「已提供程式碼說明」一節。

## 四、已提供程式碼說明

1. 從下面的github的專案，創建一個 branch，將 poseitech-java-assignments 匯入 eclipse。  
<https://github.com/poseitechtw/assignments>
2. 學生系統資料庫模型文件位置：poseitech-java-assignments/docs/index.html
3. poseitech-java-assignments 已定義的 Java classes, interfaces 及 package 說明
  - a. Data Access Object (DAO)  
com.poseitech.assignment.dao - 已定義本試題需要實作的 DAO 相關函式，自行參考相關 java interfaces 說明，如有不足可自行增加。不得移除已定義的函式，但可變更函式之輸出入參數。  
  
com.poseitech.assignment.dao.imp - DAO 實作類別必須存放此package中。
  - b. Services  
com.poseitech.assignment.service - 已定義本試題需要實作的 Service 相關函式，自行參考 java interfaces 說明，如有不足可自行增加。不得移除已定義的函式，但可變更函式之輸出入參數。  
  
com.poseitech.assignment.service.imp - Service 實作類別別必須存放此 package中。
  - c. Entity  
com.poseitech.assignment.entity - 本試題僅提供 Java POJO 部分類別，O-R mapping 需依照學生系統資料庫模型將其實作出來。
  - d. Data Transfer Object (Dto)  
com.poseitech.assignment.dto - 本試題使用 J2EE design pattern DTO，DTO 使其與Hibernate OR-Mapping 架構解離，並作為 API layer 與 Service layer 溝通的資料媒介。
  - e. Spring MVC View Model  
com.poseitech.assignment.mvc.model - Spring MVC view model 類別，現無提供任何 view model 類別，請依照試題需求設計相關 view model 類別。
  - f. Spring MVC Controller  
com.poseitech.assignment.mvc.controller - 依試題需求實作 RESTful APIs
4. maven pom.xml  
pom 已定義的相關函式庫版本不得變更，但可依照需求自行追加。

## 五、試題需求

1. 資料庫必須使用 HQLDB。
2. 此應用程式必須使用 spring boot web 執行，其 HTTP 埠號為 10080，HTTPS埠號為10443。
3. Spring, Hibernate必須使用 Java Annoation來完成相關設定。
4. 完成 com.poseitech.assignment.dao.SutdentDao, com.poseitech.assignment.service.AssignmentService com.poseitech.assignment.mvc.controller.AssignmentController 的實作
5. DAO, Service, Controller 必須撰寫 junit test case，且必須可以正確執行測試。
6. AssignmentController 必須完成下列 RESTful API  
格式說明：  
{ } 表示 path parameter

a. 查詢特定的學生

RESTful URL template	/assignments/api/v1/students/{studentId}
Support HTTP Methods	GET
RESTful Response	{ data: { }, // 學生資料 error: "" // 若發生錯誤，顯示錯誤訊息。 }

b. 依條件查詢學生

RESTful URL template	/assignments/api/v1/students?_method=r
Support HTTP Methods	POST
HTTP Post Body	{ id: "student identify number", name: "the name of student", registerDate: "yyyyMMdd" }
RESTful Response	{ data: { [an array of students] }, // 所有符合條件的學生資料 error: "" // 若發生錯誤，顯示錯誤訊息。 }

c. 查詢所有學生

RESTful URL template	/assignments/api/v1/students?start={startRow Number}&limit={limit}
Support HTTP Methods	GET or POST
HTTP Post Body	{ start: "start row number", limit: "fetch size" }
RESTful Response	{ data: { [an array of students] }, error: "" // 若發生錯誤，顯示錯誤訊息。 }

d. 新增一個學生

RESTful URL template	/assignments/api/v1/students?_method=c
Support HTTP Methods	POST
HTTP Post Body	{ name: "the name of student", registerDate: "yyyyMMdd", remark: "some note" }
RESTful Response	{ data: { uri: http://server-name:port/assignments/api/v1/students/"student identify number" id: "student identify number", name: "the name of student", registerDate: "yyyyMMdd", remark: "some note" }, error: "" // 若發生錯誤，顯示錯誤訊息。 }

e. 查詢各科成績的學生人數

RESTful URL template	/assignments/api/v1/students/grades
Support HTTP Methods	GET
RESTful Response	<pre>{   date: {     [       {         level: "A",         course: "課程名稱",         count: "總計獲得A的學生"       },       .....     ]   },   error: "" // 若發生錯誤，顯示錯誤訊息。 }</pre>

7. 製作一簡單網頁展示上述的 RESTful API，網頁格式不拘，但必須使用 ajax 調用 RESTful API