

# Spring Boot 實作班



段維瀚 老師





# 課程安排

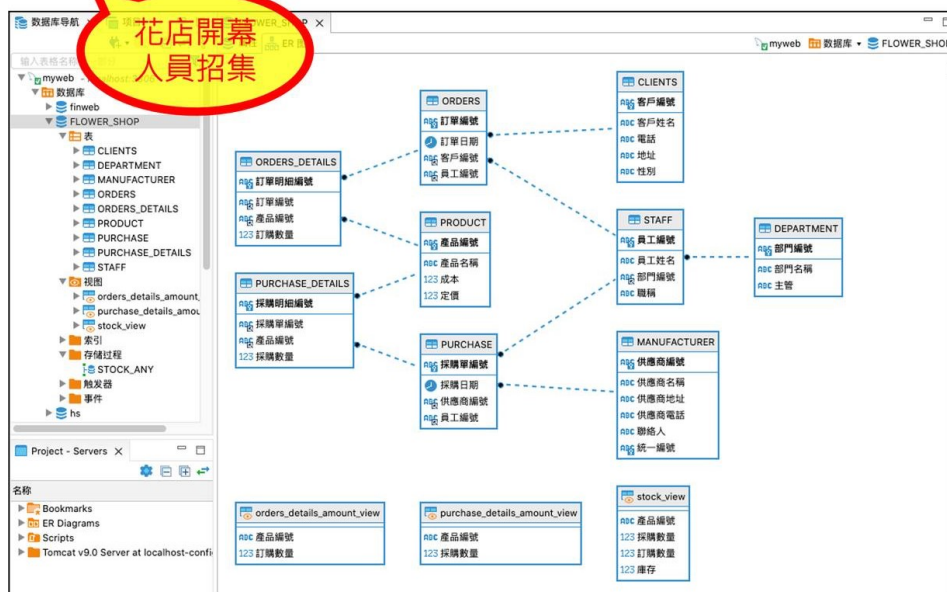
- 1~2 堂 **Springboot**
  - **SpringData JPA** 關聯關係
  - **Thymeleaf** 網頁渲染
- 3~8 實作



# 實作專題~



## 2022~2023 企業產訓實作班-新版專題



spring boot



Spring Data

{ REST:API }

### 專題特色：

1、了解資料表各種關聯關係的處理  
一對多、多對一、多對多

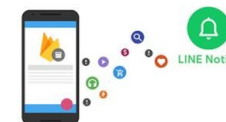


資料表關聯建置  
View  
預儲程序實作

2、結合推播 WebSocket 整合後台訊息



3、前端利用 Line Notify 進行消費通知



4、精美圖表分析





# Spring Boot

- **Spring Boot**是由**Pivotal**團隊提供的全新框架，其設計目的是用來簡化**Spring**應用的建立、執行、除錯、部署等。使用**Spring Boot**可以做到專注於**Spring**應用的開發，而無需過多關注**XML**的配置。**Spring Boot**使用“習慣優於配置”的理念，簡單來說，它提供了一堆依賴打包，並已經按照使用習慣解決了依賴問題。使用**Spring Boot**可以不用或者只需要很少的**Spring**配置就可以讓企業專案快速執行起來。





# Spring Boot

- 習慣 優於 配置
  - Spring Boot 擴展了 Spring framework，透過建立 Spring application 省去了冗長的設定檔(configurations) 能更快更有效率地開發應用系統







# Spring Tools

# 4

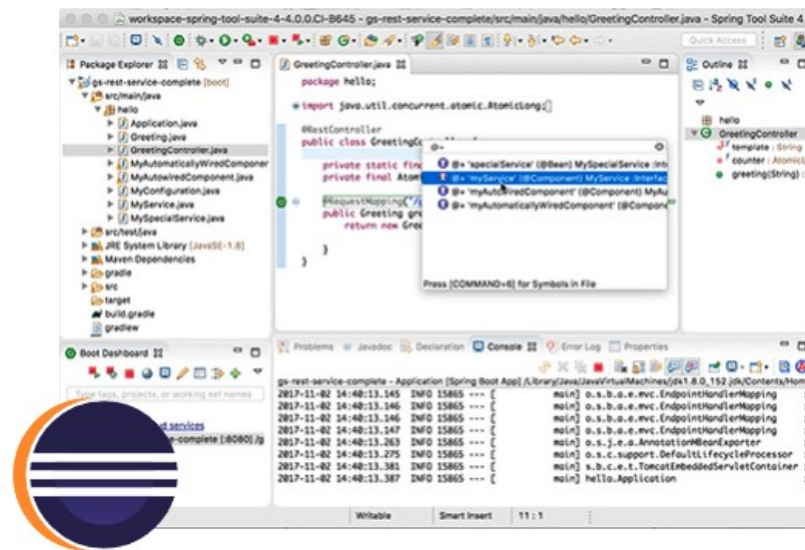
## Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4.  
Free. Open source.

4.12.0 - LINUX X86\_64

4.12.0 - MACOS X86\_64

4.12.0 - WINDOWS X86\_64





# ***pom.xml***

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-tomcat</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-configuration-processor</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```





# YAML 語法

- **YAML 基本語法**

- 使用縮排表示層級關係
- 大小寫區分
- **key: value** (注意「:」後有空格)

- **YAML 數據結構**

- 字面值：鍵值與數值
- 物件：鍵值與數值的集合
- 數組陣列







# *application.yml*

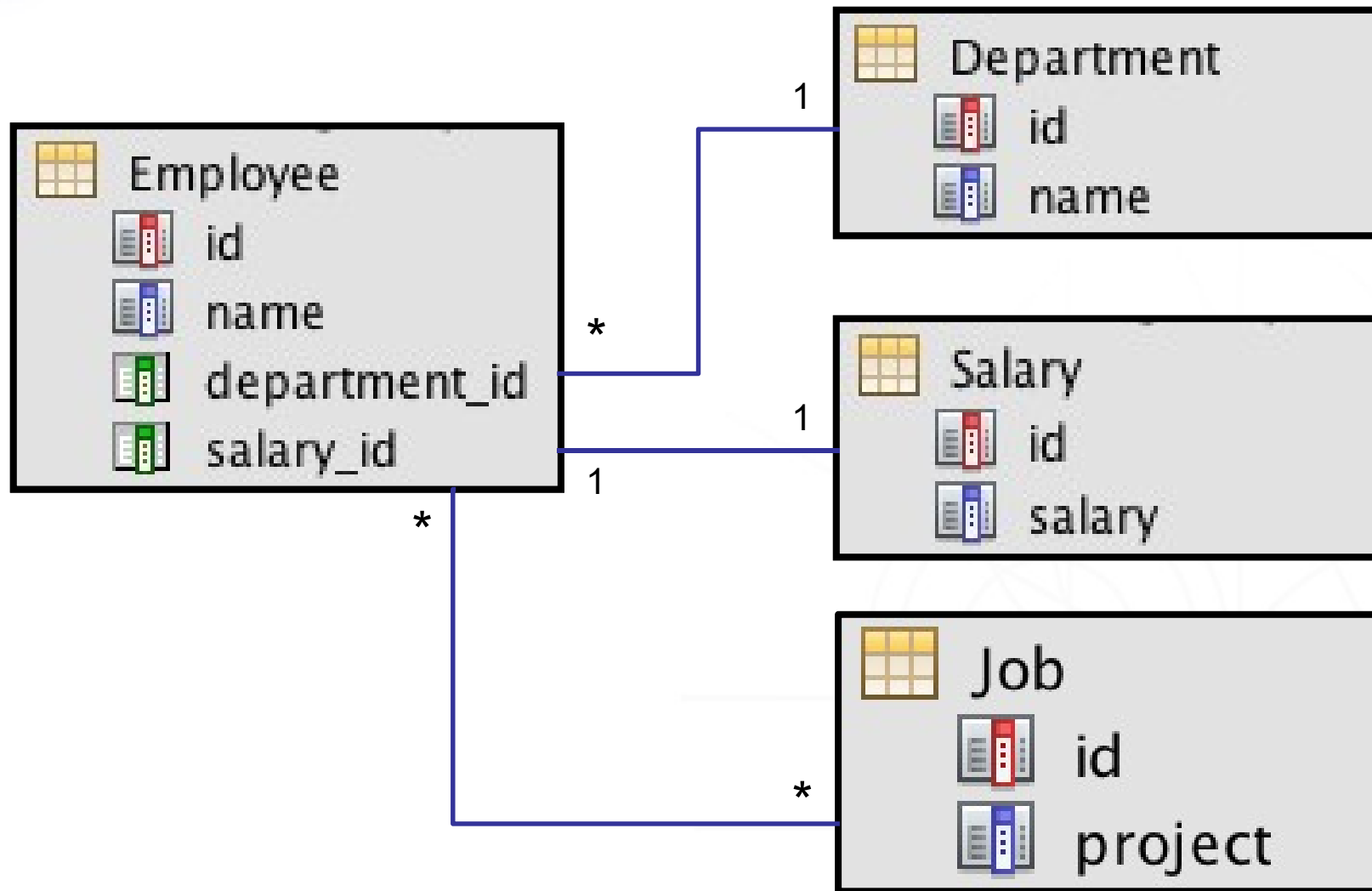
```
server:
  port: 8081
  servlet:
    context-path: /sts
spring:
  thymeleaf:
    cache: false
    mode: HTML
    encoding: UTF-8
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/web?serverTimezone=UTC&characterEncoding=utf-8&useUnicode=true
    username: root
    password: 12345678
  jpa:
    database: MYSQL
    show-sql: true
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MySQL5Dialect
        format_sql: true
```





# *SpringData JPA 關聯關係*

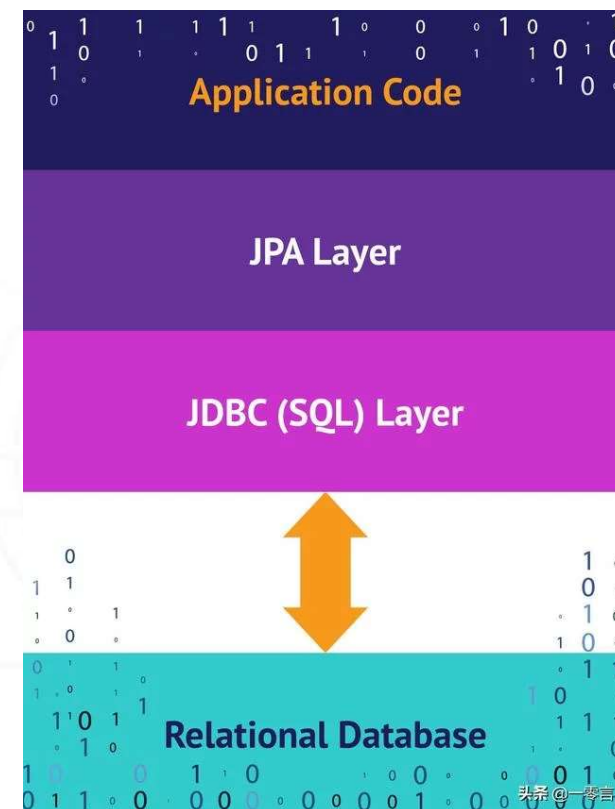




# JPA

- Java Persistence API

- Java 數據持久性
- 保存資料

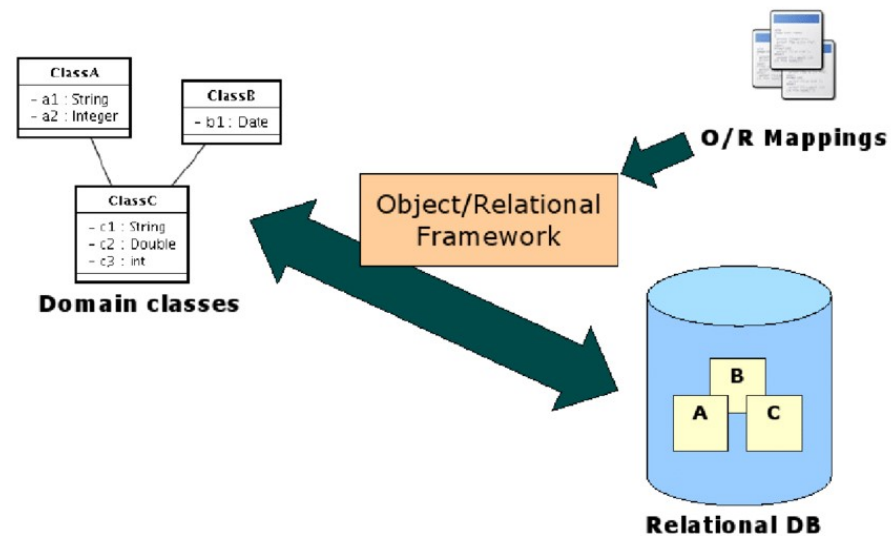


# JPA

- Java Persistence API

- ORM

- Object Relational Mapping
    - 將資料庫資料映射到Java物件(POJO)







# JPA 操作設定

- **@Entity** 映射實體
- **@Table** 映射表
- **@Column** 映射欄位
- **@Id** 映射主鍵
  - **@GeneratedValue** 標示符自動生成
- **@JoinColumn** 關聯欄位
- **@PrimaryKeyJoinColumn** 主鍵關聯
- **@JoinTable** 關聯表的建立
- **@OrderBy** 排序





# JPA 操作設定

- @OneToOne
- @OneToMany
- @ManyToOne
- @ManyToMany
- @Transaction 交易進行
- @JsonIgnoreProperties
  - 防止 json 生成遞迴資料





# JPA 操作設定

- 聯級操作**Cascade**

預設是不使用聯級操作，而可設定的聯級操作如下所示：

CascadeType.PERSIST	在儲存時一併儲存 被參考的物件。
CascadeType.MERGE	在合併修改時一併 合併修改被參考的物件。
CascadeType.REMOVE	在移除時一併移除 被參考的物件。
CascadeType.REFRESH	在更新時一併更新 被參考的物件。
CascadeType.ALL	無論儲存、合併、 更新或移除，一併對被參考物件作出對應動作。



# JPA 操作設定

- FetchType

- FetchType.EAGER

- 立刻將table中的資料注入物件

- FetchType.LAZY

- 除非真正要使用到該屬性的值（自行在Session未關閉前，撰寫程式加入），否則不會真正將資料從table中注入物件

- 預設值

@Basic	FetchType.EAGER
@OneToOne	FetchType.EAGER
@ManyToOne	FetchType.EAGER
@OneToMany	FetchType.LAZY
@ManyToMany	FetchType.LAZY





## 配置 *pom.xml*

```
<!-- JPA -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<!-- MySQL -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.19</version>
</dependency>
```



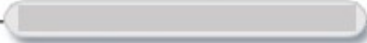






# *application.yml*

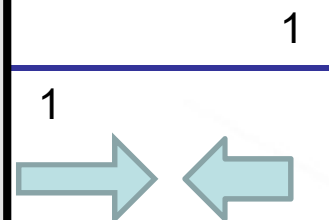
```
server:
  port: 8081
  servlet:
    context-path: /sts
    encoding:
      charset: UTF-8

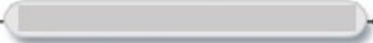

spring:
  thymeleaf:
    cache: false
    mode: HTML
    encoding: UTF-8
  datasource:
    url: jdbc:mysql://localhost:3306/web?serverTimezone=UTC&characterEncoding=utf-8&useUnicode=true
    username: root
    password: 12345678
    driverClassName: com.mysql.cj.jdbc.Driver
  jpa:
    database: MYSQL
    show-sql: true
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MySQL5Dialect
        format_sql: true
```




<b>@OneToOne</b>	一對一
@ManyToOne	多對一
@OneToMany	一對多
@ManyToMany	多對多





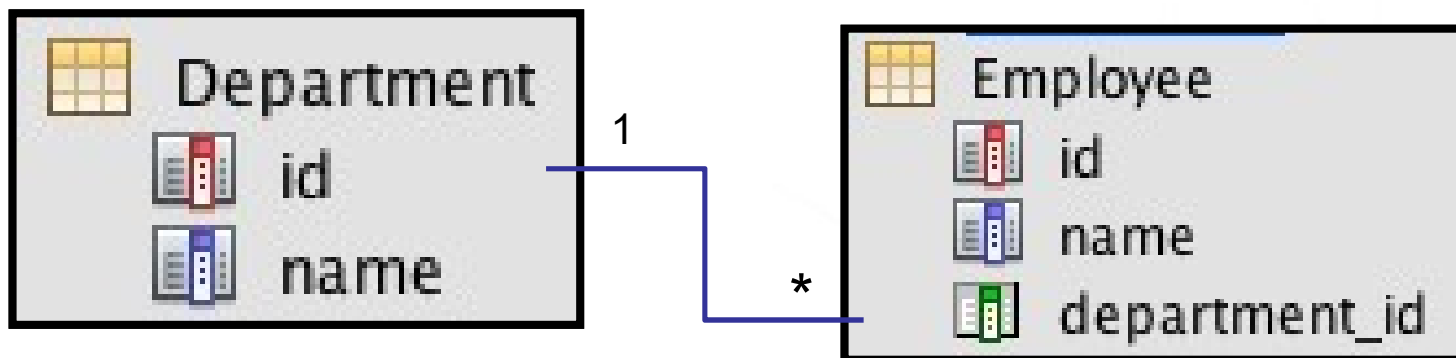


@OneToOne	一對一
@ManyToOne	多對一
@OneToMany	一對多
@ManyToMany	多對多



# Many to One

## 多對一



# One to Many

## 一對多





# mappedBy

- mappedBy = "department"
  - 表示資料變更來自於 Employee 的 department 屬性

```
@Entity // 此類別是資料庫實體表格的對象
public class Department implements Serializable {

    @Id // 主鍵 id
    @GeneratedValue // 自動產生 id 值
    private Integer id;

    @Column(length = 18, nullable = false) // 資料欄
    private String name;

    @OneToMany(cascade = CascadeType.ALL,
              fetch = FetchType.EAGER,
              mappedBy = "department")
    private Set<Employee> employees;

    // 略 ...
}
```

```
@Entity
public class Employee implements Serializable {

    @Id
    @GeneratedValue
    private Integer id;

    @Column(length = 20, nullable = false)
    private String name;

    @ManyToOne(cascade = CascadeType.ALL)
    private Department department;

    // 略 ...
}
```



@OneToOne	一對一
@ManyToOne	多對一
@OneToMany	一對多
<b>@ManyToMany</b>	<b>多對多</b>



# @ManyToMany 多對多



\*



\*



#	id	name
1	1	John
2	2	Marry

#	user_id	role_id
1	1	1
2	2	1
3	1	2

#	id	name
1	1	Student
2	2	Player

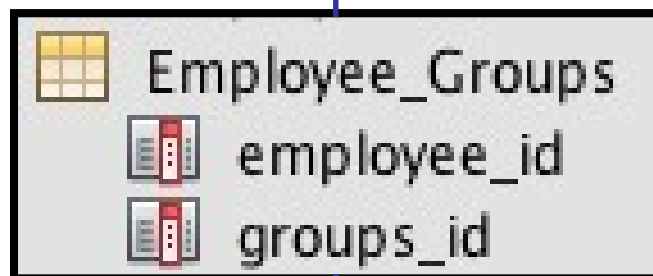


# *Homework*





#	id	name
1	1	John
2	2	Mary
3	3	Tom



1 \*

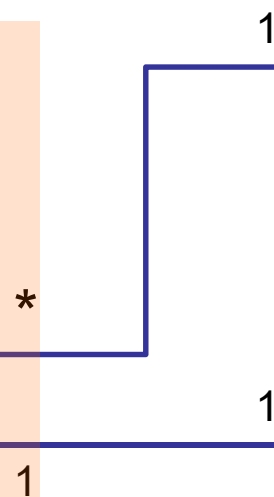
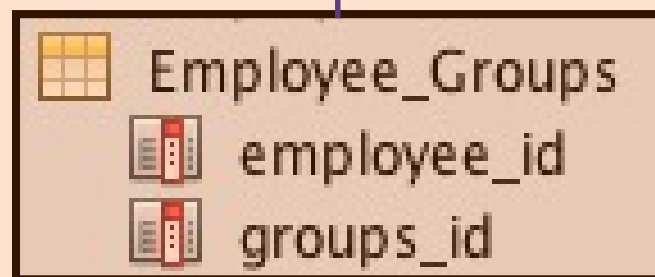
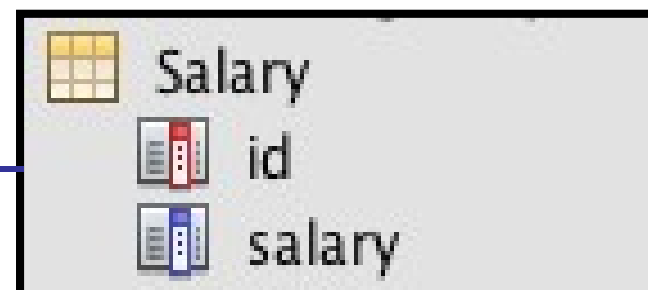
#	employee_id	groups_id
1	1	1
2	2	1
3	3	1
4	1	2
5	2	2



1 \*

#	id	name
1	1	Bikes
2	2	Climbing







# ThymeLeaf



<https://www.thymeleaf.org/>

Thymeleaf是適用於Web和獨立環境的現代伺服器端Java模板引擎

## Natural templates

HTML templates written in Thymeleaf still look and work like HTML, letting the actual templates that are run in your application keep working as useful design artifacts.

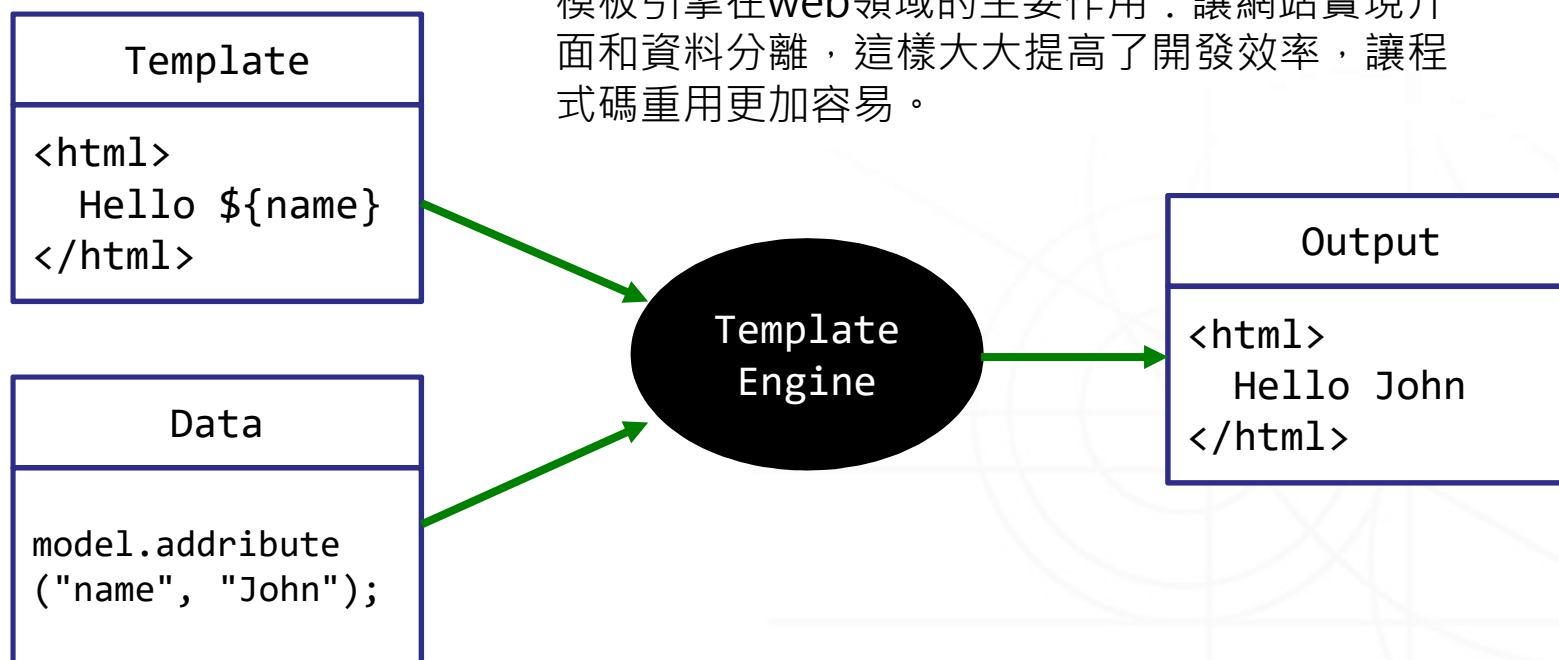
```
1 <table>
2   <thead>
3     <tr>
4       <th th:text="#{msgs.headers.name}">Name</th>
5       <th th:text="#{msgs.headers.price}">Price</th>
6     </tr>
7   </thead>
8   <tbody>
9     <tr th:each="prod: ${allProducts}">
10      <td th:text="${prod.name}">Oranges</td>
11      <td th:text="${#numbers.formatDecimal(prod.price, 1, 2)}">0.99</td>
12    </tr>
13  </tbody>
14 </table>
```



# 模板引擎

- JSP、Thymeleaf、Freemarker

模板引擎在web領域的主要作用：讓網站實現介面和資料分離，這樣大大提高了開發效率，讓程式碼重用更加容易。





# Thymeleaf

- Thymeleaf作為被Springboot官方大力推薦的模板引擎
  - 動靜分離
    - Thymeleaf選用html作為模板頁，這是任何一款其他模板引擎做不到的！
  - 開箱即用：
    - Thymeleaf提供標準和Spring標準兩種方言，可以直接套用模板來實現。
  - 目前預設版本：**2.x**





# Thymeleaf

- 編寫框架
  - **pom.xml** 加入 **Thymeleaf** 的依賴
  - **application.yml** 整合配置內容
  - **controller** 用來繫結 **Thymeleaf html** 頁面
  - **static** 存放靜態資源
    - **css**、**javascript**、靜態/傳統 **html**
  - **templates** 存放模板引擎
    - **Thymeleaf** 的 **html** 檔案





# Thymeleaf

```
<tr th:each="user, stat : ${users}">
  <td th:text="${stat.index}"></td>
  <td th:text="${stat.count}"></td>
  <td th:text="${user.id}"></td>
  <td th:text="${user.name}"></td>
  <td th:text="${user.password}"></td>
  <td th:text="${user.birth}"></td>
  <td th:text="${user.height}"></td>
  <td th:text="${user.weight}"></td>
  <td>
    <a th:href="@{/user/{id}(id=${user.id})}">編輯</a>
  </td>
  <td>
    <a th:href="@{/user/delete/{id}(id=${user.id})}">刪除</a>
  </td>
</tr>
```

# Thymeleaf

- 宣告

- `<html xmlns:th="http://www.thymeleaf.org">`

標籤	作用	示例
th:id	替換id	<code>&lt;input th:id="\${user.id}"/&gt;</code>
th:text	文字替換	<code>&lt;p text:="\${user.name}"&gt;big sai&lt;/p&gt;</code>
th:utext	支援html的文字替換	<code>&lt;p utext:="\${htmlcontent}"&gt;content&lt;/p&gt;</code>
th:object	替換物件	<code>&lt;div th:object="\${user}"&gt;&lt;/div&gt;</code>
th:value	替換值	<code>&lt;input th:value="\${user.name}" &gt;</code>
th:each	迭代	<code>&lt;tr th:each="student:\${user}" &gt;</code>
th:href	替換超連結	<code>&lt;a th:href="@{index.html}"&gt;超連結&lt;/a&gt;</code>
th:src	替換資源	<code>&lt;script type="text/javascript" th:src="@{index.js}"&gt;&lt;/script&gt;</code>





# ThymeLeaf

- 表達式

- $\${}$  變數表達式
- $*\{\}$  可選變數表達式
- $\#\{\}$  資源變數表達式
- $@\{\}$  鏈結表達式
- 字串連接、數學運算、布林運算、?: 運算







# ThymeLeaf

- 迭代狀態參數配置
  - **index**: 當前迭代索引，從**0**開始
  - **count**: 當前迭代序號，從**1**開始
  - **size**: 元素的總量
  - **current**: 當前正在迭代的元素
  - **even/odd**: 迭代是奇數列還是偶數列
  - **first**: 當前迭代是否是第一個
  - **last**: 當前迭代是否是最後一個





# ThymeLeaf

- 條件判斷表達式可以是如下的形式：
  - 布林類型返回 **true/false**
  - 數值類型且值不是0，返回 **true**
  - 字元類型且值不是0，返回 **true**
  - 字串類型且值不是
    - **"false"**、**"off"**、**"no"**，返回 **true**
  - 之外其他類型其值不是 **null**，返回 **true**
  - 物件對象的值是**null**，返回 **false**



# Thymeleaf 3 新語法

```
<dependency>
  <groupId>org.thymeleaf</groupId>
  <artifactId>thymeleaf</artifactId>
  <version>3.0.11.RELEASE</version>
</dependency>
<dependency>
  <groupId>org.thymeleaf</groupId>
  <artifactId>thymeleaf-spring5</artifactId>
  <version>3.0.11.RELEASE</version>
</dependency>
```

```
<dependency>
  <groupId>org.thymeleaf.extras</groupId>
  <artifactId>thymeleaf-extras-java8time</artifactId>
  <version>3.0.4.RELEASE</version>
</dependency>
```

支援 java 8 Date

- **#dates** : *java.util.Date* 對象的實用方法
- **#calendars** : 類似於 **#dates** , 用於 *java.util.Calendar* 對象
- **#numbers** : 格式化數字對象的實用方法
- **#strings** : *String* 對象的實用方法
- **#objects** : Java *Object* 類的通用
- **#bools** : 布爾評估的實用方法
- **#arrays** : 數組的實用方法
- **#lists** : 列表的實用方法
- **#sets** : 集合的實用方法
- **#maps** : 地圖的實用方法
- **#aggregates** : 用於在數組或集合上創建聚合的實用方法
- **#messages** : 用於在變量表達式中獲取外部化消息的實用方法

學習參考網址 : <https://www.baeldung.com/spring-thymeleaf-3-expressions>  
<https://www.baeldung.com/dates-in-thymeleaf>



# ***Bootstrap***

使用全球最流行的前端  
開發工具 Bootstrap，  
快速設計及自定義響應  
式網站





# ***Bootstrap***

於網頁<header>中配置：

```
<link rel="stylesheet"  
href="https://maxcdn.bootstrapcdn.com/bootstrap/  
3.3.7/css/bootstrap.min.css"/>
```



# Bootstrap 網格

## 柵格参数

通过下表可以详细查看 Bootstrap 的柵格系统是如何在多种屏幕设备上工作的。

	超小屏幕 手机 (<768px)	小屏幕 平板 (≥768px)	中等屏幕 桌面显示器 (≥992px)	大屏幕 大桌面显示器 (≥1200px)
柵格系统行为	总是水平排列		开始是堆叠在一起的, 当大于这些阈值时将变为水平排列C	
<code>.container</code> 最大宽度	None (自动)	750px	970px	1170px
类前缀	<code>.col-xs-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>
列 (column) 数	12			
最大列 (column) 宽	自动	~62px	~81px	~97px
槽 (gutter) 宽	30px (每列左右均有 15px)			
可嵌套	是			
偏移 (Offsets)	是			
列排序	是			