

# 4-1 @Controller 配置 Fornt-End Controller





## 第四單元大綱

4-1 : @Controller配置Fornt-End Controller

4-2 : thymeleaf Action調用View Page啟動配置

4-3 : Action Dispatcher View Page應用架構

4-4 : 單元四測驗



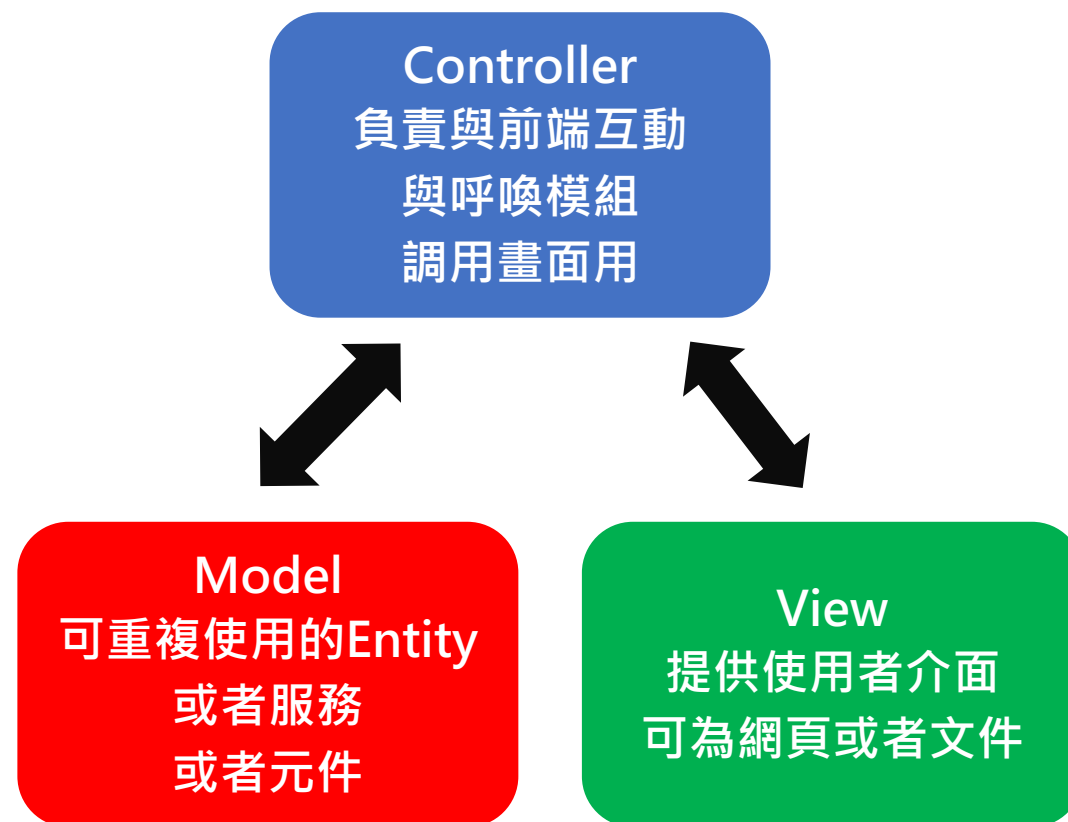
## MVC 設計模式

- Model-View-Controller主要目的是將設計分離關注，並且整合。
- 將網站系統開發切割為控制流程，重複使用的模組與動態調用的頁面或者文件等。

讓系統開發聚焦在特定範圍撰寫  
可讓系統具有高度的可維護性

Spring MVC Web設計控制項類別採用POJO架構  
可以使用xml或者annotation進行配置即可

@Controller標註使用佈署控制器應用



## 規劃好一個package

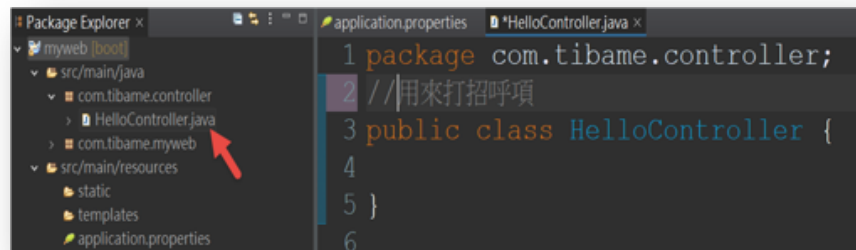
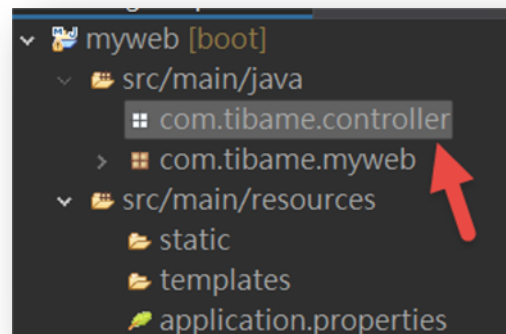
- 用來進行元件掃描註冊的依據

## 新增一個class

- 採用POJO架構設計

## 新增一個method進行流程處理與調用畫面

- 透過return回應一個調用樣板資料夾頁面的旗標(String)



```
package com.tibame.controller;

import org.springframework.stereotype.Controller;

//用來打招呼項
//採用POJO架構設計
@Controller
public class HelloController {
    |
}
```



```
package com.tibame.controller;

import org.springframework.stereotype.Controller;

//用來打招呼項
//採用POJO架構設計
@Controller
public class HelloController {
    //打招呼方法
    public String helloWorld()
    {
        //處理流程
        return "hello"; //回應調用頁面名稱旗標
    }
}
```

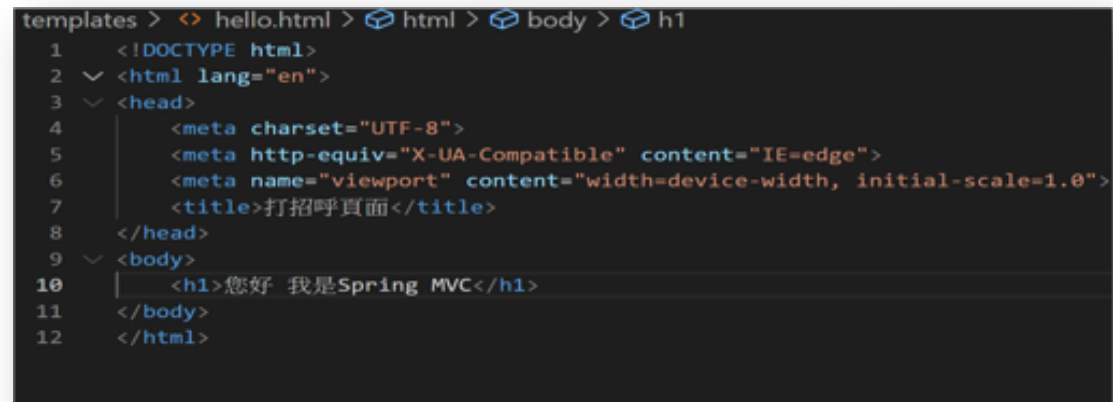
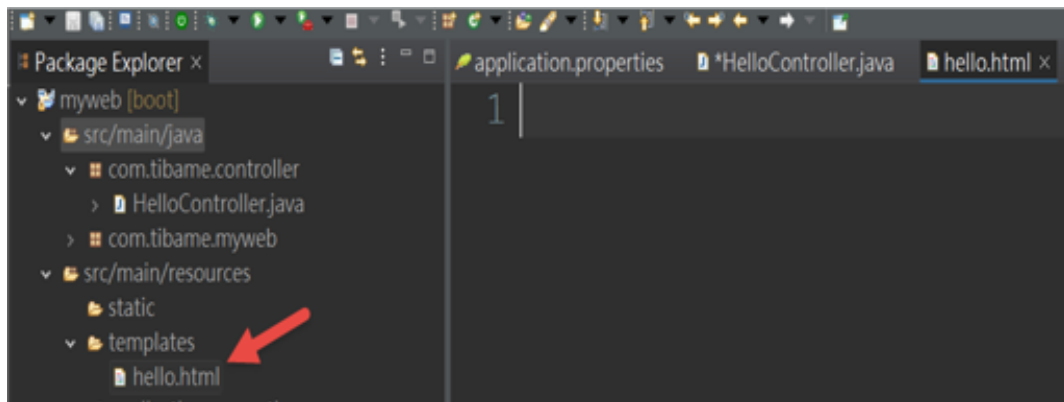
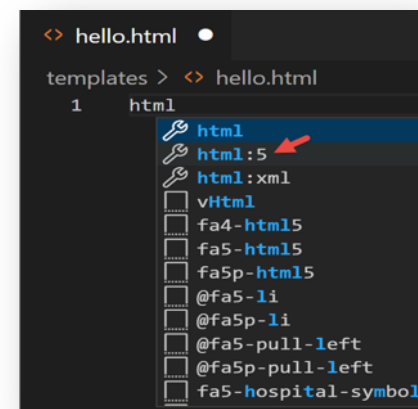
## Controller-Action調用的頁面要及時設計

- 執行階段會進行驗證，即使使用者端不進行請求

在thymeleaf預設的樣板資料夾下新增頁面，採用html附檔名，頁面名稱同Method回應的字串名稱

網頁編輯建議採用Visual Studio code工具進行

<https://code.visualstudio.com/download>



控制器類別需要進行佈署到 Spring Container中，才能提供服務

可以透過應用系統啟動(Main Program)時進行元件掃描設定

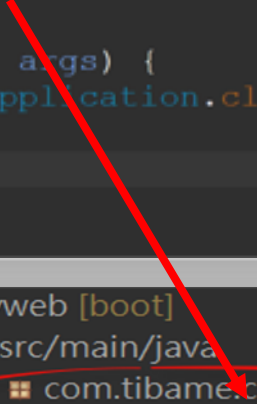
@ComponentScan Annotation配置

- 掃描 package

```
package com.tibame.myweb;

import org.springframework.boot.SpringApplication;
//配置自動掃描元件註冊到Spring Container
@SpringBootApplication
@ComponentScan(basePackages= {"com.tibame.controller"})
public class MywebApplication {

    public static void main(String[] args) {
        SpringApplication.run(MywebApplication.class, args);
    }
}
```

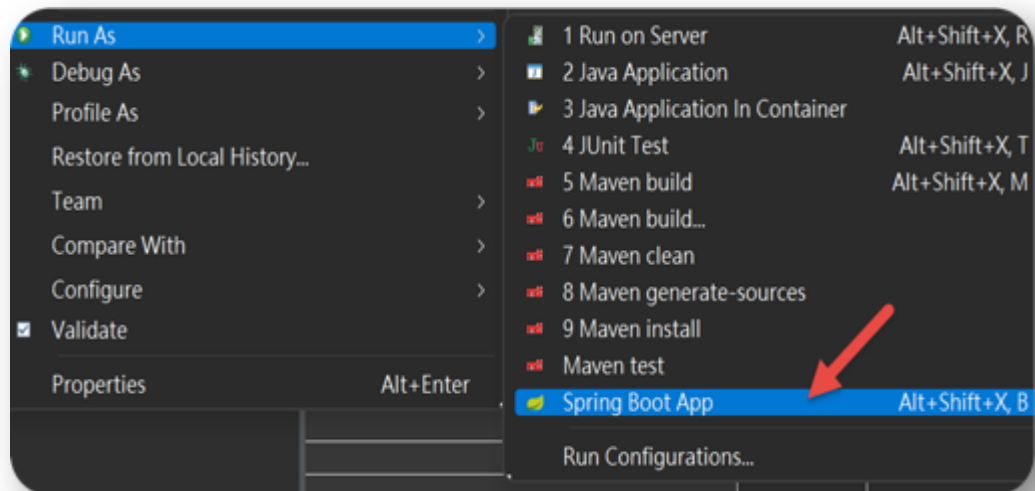


myweb [boot]

- src/main/java
  - com.tibame.controller**
    - HelloController.java
  - com.tibame.myweb
    - MywebApplication.java
- src/main/resources
  - static
  - templates
    - hello.html
  - application.properties

將專案啟動為Spring Boot App

透過 jar 方式(預設設定)  
從CLI視窗啟動網站系統



```
//用來打招呼項  
//採用POJO架構設計  
@Controller  
public class HelloController {  
    //建構子  
    public HelloController() {  
        System.out.println("Hello控制器產生Instance...");  
    }  
}
```

```
2022-07-19 09:12:21.003 INFO 31296 --- [main] com.tibame.myweb.MywebAppl  
2022-07-19 09:12:21.005 INFO 31296 --- [main] com.tibame.myweb.MywebAppl  
2022-07-19 09:12:21.413 INFO 31296 --- [main] o.s.b.w.embedded.tomcat.To  
2022-07-19 09:12:21.417 INFO 31296 --- [main] o.apache.catalina.core.Sta  
2022-07-19 09:12:21.418 INFO 31296 --- [main] org.apache.catalina.core.S  
2022-07-19 09:12:21.463 INFO 31296 --- [main] o.a.c.c.C.[Tomcat].[localh  
2022-07-19 09:12:21.463 INFO 31296 --- [main] w.s.c.ServletWebServerAppl  
Hello控制器產生Instance...  
2022-07-19 09:12:21.615 INFO 31296 --- [main] o.s.b.w.embedded.tomcat.To  
2022-07-19 09:12:21.620 INFO 31296 --- [main] com.tibame.myweb.MywebAppl
```



## 總結：4-1 @Controller 配置Fornt-End Controller

設計起Spring MVC第一個Controlle與配置  
之後，我們如何設計打招呼功能Action對外的  
配置

