

10-1 RESTful軟體風格規範





第十單元大綱

- 10-1 RESTful軟體風格規範
- 10-2 JSON文件規格
- 10-3 資料存取RESTful Service設計
- 10-4 HTTP URL傳遞參數應用
- 10-5 單元測驗



何謂RESTful/REST軟體風格

是一種軟體風格

表現層狀態轉換(Representational State Transfer) Fielding博士於2000年在他的博士論文[1]中提出來的一種全球資訊網軟體架構風格，目的是便於不同軟體/程式在網路（例如網際網路）中互相傳遞資訊

架構在通訊協定HTTP上

允許使用者端發出以統一資源標識符存取和操作網路資源的請求，而與預先定義好的無狀態操作集一致化

採用HTTP Request Method-GET/POST/PUT/DELETE

- 對應資料的新增修改或者刪除與查詢

使用@RestController annotation進行標註註冊

採用POJO方式規劃類別

具有Middleware架構，可將傳輸的JavaBean進行JSON文件序列化或者反序列化

HelloService規劃

```
@RestController  
public class HelloService {  
  
    //回應一個Text/plain 簡單文字串到前端  
    @GetMapping(path="/api/hello/helloworld", produces="text/plain")  
    public String helloWorld() {  
        return "世界和平";  
    }  
}
```

描述Class註冊於Spring Container
下為一個RESTful Service控制項

Response Content-Type設定

Spring Boot主程式類別進行ComponentScan配置

指定Service Package

配置到Spring Container中

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

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

}
```

```
2022-08-27 07:29:40.001 INFO 13392 --- [
2022-08-27 07:29:40.025 INFO 13392 --- [
2022-08-27 07:29:40.104 INFO 13392 --- [
2022-08-27 07:29:40.167 INFO 13392 --- [
2022-08-27 07:29:40.434 INFO 13392 --- [
2022-08-27 07:29:40.439 INFO 13392 --- [
com.tibame.domain.Engine@1fba3fd6
1. 公司物件建構...
2. 員工物件建構...
Hello控制器產生Instance...
2. 員工物件建構...
Hello Service產生個體物件...
```

安裝RESTful前端測試工具postman

<https://dl.pstmn.io/download/latest/win64>

測試HelloService第一個Hello服務

Body	Cookies	Headers (5)	Test Results	Status: 200
KEY		VALUE		
Content-Type		①	text/plain;charset=UTF-8	
Content-Length		①	12	
Date		①	Fri, 26 Aug 2022 23:14:37 GMT	
Keep-Alive		①	timeout=60	
Connection		①	keep-alive	

Overview GET http://localhost:8080/

http://localhost:8080/api/hello/helloworld

GET http://localhost:8080/api/hello/helloworld

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize Text

1 世界和平

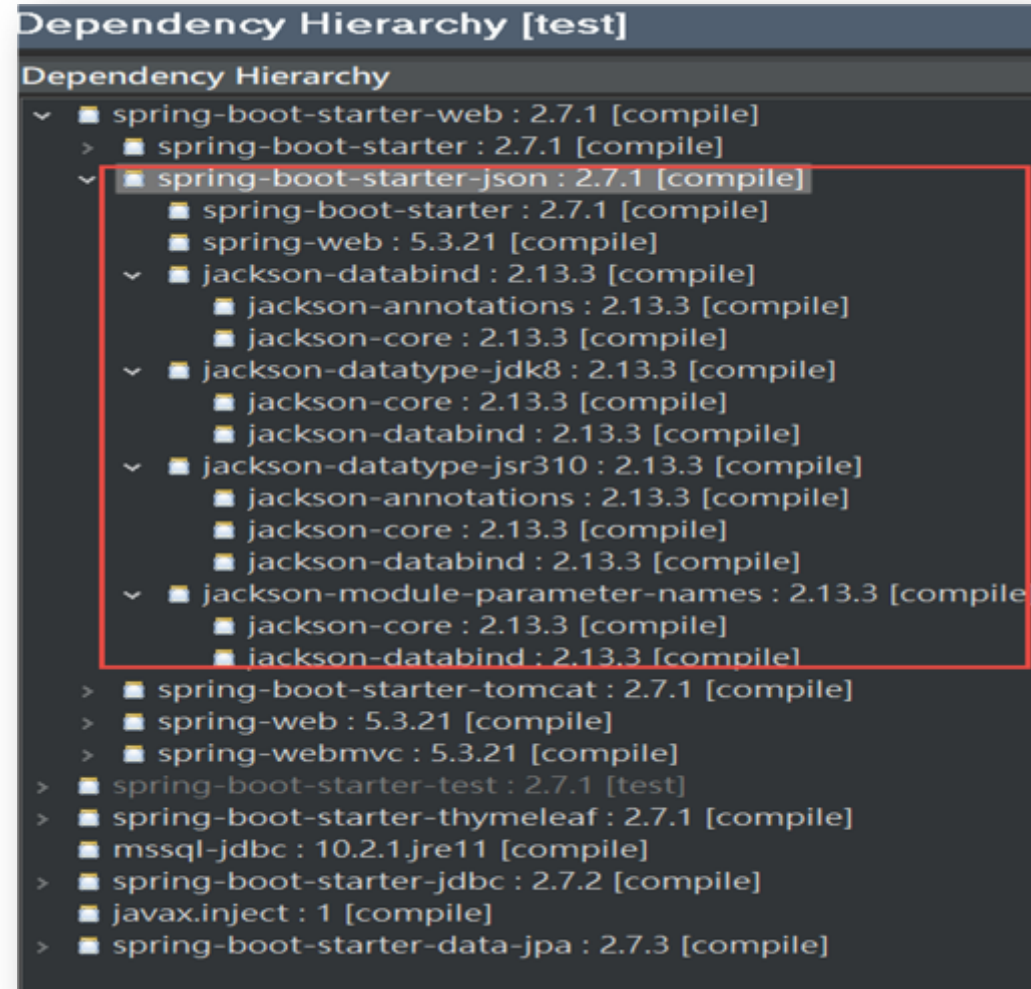
Spring Boot RESTful Service回應物件主動序列化為 JSON

其中機制採用 Jackson解析框架

架構在Middleware(Filter)上進行物件序列化機制

Spring Boot Starter Web封裝Jackson

可透過@ResponseBody透過Jackson機制序列化物件為JSON



規畫一個JavaBean類別

透過Controller-Action回應一個實體物件

透過@ResponseBody描述方法回應時透過Jackson序列化物件

```
public class Message implements java.io.Serializable{
    private int code;
    private String msg;
    public int getCode() {
        return code;
    }
    public void setCode(int code) {
        this.code = code;
    }
    public String getMsg() {
        return msg;
    }
    public void setMsg(String msg) {
        this.msg = msg;
    }
}
```

```
//回應物件序列化成Json內容
@GetMapping(path="/api/hello/helloworldobject", produces="application/json")
@ResponseBody
public Message helloWorldObject() {
    //建構JavaBean物件
    Message message=new Message();
    message.setCode(200);
    message.setMsg("Hello World");
    return message;
}
```


http://localhost:8080/api/hello/hellowobject

GET ⌵ http://localhost:8080/api/hello/hellowobject

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ⌵ ↺

```
1 {  
2   "code": 200,  
3   "msg": "Hello World"  
4 }
```

Body Cookies Headers (5) Test Results ⊕ Status: 200 OK

KEY	VALUE
Content-Type	① application/json
Transfer-Encoding	① chunked
Date	① Fri, 26 Aug 2022 23:46:22 GMT
Keep-Alive	① timeout=60
Connection	① keep-alive



總結：10-1 RESTful軟體風格規範

了解RESTful風格之後，請且設計一個簡單的Hello Service回應簡易訊息到前端之後，我們需要更進一步了解JSON文件格式與類別規劃之間的關係。

