* Config Server: <https://ithelp.ithome.com.tw/articles/10193676>

1. Mave dependency:

|  |
| --- |
| spring-cloud-config-server |

1. application.properties

server.port=8000

spring.cloud.config.server.git.uri= file://D:/Bill/workspace/Spring/Spring Cloud/config-repo

1. bootstrap.properties

spring.application.name=config-service

1. 加上 @EnableConfigServer Tag 即可啟用

* Config Client: <https://ithelp.ithome.com.tw/articles/10193676>

1. Mave dependency:

|  |
| --- |
| spring-cloud-starter-config |
| spring-boot-starter-actuator |
| spring-boot-starter-web |

1. application.properties

management.endpoints.web.exposure.include=\*

\*加上這段才能執行 curl -X POST <http://localhost:8001/actuator/refresh>

1. bootstrap.properties

spring.application.name=config-client-with-bus

spring.cloud.config.uri=http://localhost:8000

設定config server 路徑與設定檔名稱

1. Application.java

@SpringBootApplication public class Application {

@Autowired

void setEnvironment(Environment e) {

System.out.println(e.getProperty("configuration n.projectName"));

}

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

}

1. ProjectNameRestController.java

@RestController

@RefreshScope

public class ProjectNameRestController {

@Value("${configuration.projectName}")

private String projectName;

@RequestMapping("/project-name")

public String projectName() {

return this.projectName;

}

}

1. 本機設定檔D:\Bill\workspace\Spring\Spring Cloud\config-repo

config-client-with-bus.properties

server.port=${PORT:8001}

configuration.projectName=Spring Cloud

1. 向Server端確認參數

curl -X GET 'http://localhost:8000/config-client-with-bus/master'

1. 向Client取得project-name

curl -X GET 'http://localhost:8001/project-name'

回覆 Spring Cloud

1. 更新設定檔參數內容

configuration.projectName=Spring Cloud Test

1. 用Postman觸發Client更新

curl -X POST '<http://localhost:8001/actuator/refresh>'

[

    "configuration.projectName"

]

畫面顯示異動過的參數名稱

1. 再檢查可以發現已經更新了

curl -X GET 'http://localhost:8001/project-name'

回覆 Spring Cloud Test

**但是這樣子還是必須每一台 Client 去觸發 Refresh，所以 Spring Cloud 提供一個做法透過 RabbitMQ 來觸發**

* 修改 Config Client

1. 增加Mave dependency:

|  |
| --- |
| spring-cloud-starter-bus-amqp |

1. application.yml 增加 rabbitmq 部分
2. 安裝 RabbitMQ Server

## 安裝 RabbitMQ 的注意事項

## 安裝流程參考: <https://medium.com/@mybaseball52/install-rabbitmq-on-win-10-a039d48e1c80>

## RabbitMQ 是在 Erlang 環境下執行的，版本對應可以參考。 <https://www.rabbitmq.com/which-erlang.html>

## 安裝Erlang: <https://www.erlang.org/downloads/23.0>

## 設定環境變數

## 並在Path底下加入%ERLANG\_HOME%\bin即可

## 安裝RabbitMQ Server: <https://www.rabbitmq.com/install-windows.html>

## 安裝完畢之後進到sbin資料夾，打開終端機設置一下 RabbitMQ 的套件

## .\rabbitmq-plugins.bat enable rabbitmq\_management

## 

完成設置後，即可啟動 RabbitMQ Server。

.\rabbitmq-server.bat

## 輸入<http://localhost:15672/>，即可進入登入畫面，帳密皆為guest(預設)

## 完成登入後，就可以看到 RabbitMQ 的管理畫面了。

1. 之後啟動兩個Client(Port:8001,8002)  
   <http://localhost:8001/project-name>  
   <http://localhost:8002/project-name>
2. 更新設定檔config-client-with-bus.properties  
   configuration.projectName=Spring Cloud Bus

並執行Postman

curl -X POST '<http://localhost:8001/actuator/refresh>'

1. 觀看結果

<http://localhost:8001/project-name>  
<http://localhost:8002/project-name>