技術筆記

筆記更新紀錄

|  |  |  |
| --- | --- | --- |
| 日期 yyyy/MM/dd | 更新人員 | 更新備註 |
| 2019/04/03 | Miles | 建立文件 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目錄

[1 Spring Cloud 基本簡介 4](#_Toc5194313)

[2 Spring Cloud 優缺點 4](#_Toc5194314)

[3 Spring Cloud 練習 4](#_Toc5194315)

## 1 Spring Cloud 基本簡介

|  |
| --- |
| Spring Cloud 基本簡介 |
|  |
|  |

## 2 Spring Cloud 優缺點

|  |
| --- |
| Spring Cloud 優缺點 |
|  |
|  |

## 3 Spring Cloud 練習

|  |
| --- |
| Spring Cloud 練習 |
| 使用技術 |
| 1. 專案管理:Gradle 2. 框架: Spring Cloud |
| 使用架構 |
| 1. Microservice |
| 步驟 |
| 1. 建立Gradle專案，並且建立四個子專案(Sub Project) 2. 父專案目錄結構大致如下:      1. 子專案目錄結構大致如下:      1. 因為使用專案管理Gradle框架，所以每一個專案(不管是父專案還是子專案)都需要有一個build.gradle的配置檔，而settings.gradle 主要做注入專案的動作      1. 父專案的build.gradle 尚未研究，但是可以使用一些語法，例如allProject、subProject進行專案管理。 2. 首先，Spring Cloud 有分為 Spring Cloud Config Client 與 Spring Cloud Config Server   兩端，顧我們可以建立基本的Server配置   1. 我們先在SpringCloud-Config專案建立build.gradle配置，並且倒入基本套件與注入依賴         以上build.gradle部分內容尚未釐清目的所以需要再花時間研究  其中，    第一個是說明依賴Spring Cloud Config Server端需要使用的套件  第二個是說明依賴Spring Cloud Eureka starter啟動需要使用的套件  以上可以說明就是專案(SpringCloud-Config)為一個配置專案，並且使用到了Eureka。   1. 建立第一個Spring Cloud Config Server 配置專案入口     其中，@SpringBootApplication 是一個SpringBoot 的入口，@ComponentScan 是自動掃描，@EnableConfigServer是一個Spring Cloud 的配置路口，@EnableDiscoveryClient是代表這是一個Eureka的Clinet端口。  以上配置代表該專案的主程式是一個微服務入口。   1. 建立一個Eureka的配置專案 |
| 說明: |
| 1. Eureka:   Eureka是微服務(microservice)架構中最為核心與基礎的模塊，它主要是用來實現各種微服務實例的自動化和發現。  Spring Cloud Eureka是Spring Cloud Netfilx微服務套件中的一部分，它是建立在Netfilx Eureka基礎上二次封裝，主要負責完成微服務架構中的服務治理功能。 |