1. 尚未建立Spring Tool Suite(STS)開發環境請參考協作平台，開發環境之『開發環境建置』

<http://172.17.240.93/pages/viewpage.action?pageId=199819281>

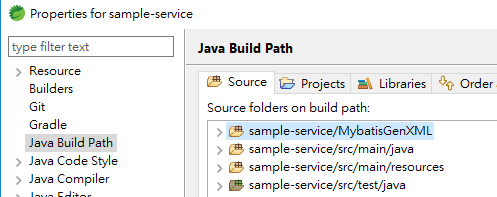
1. 建立新專案，參考協作平台，開發環境之『Spring Boot Web API 開發專案建立』  
   <http://172.17.240.93/pages/viewpage.action?pageId=199819281>
2. 此範例以portal專案當sample，請依實際專案調整檔案及設定檔內容名稱
3. Spring boot預設專案結構為三個  
   src/main/java：程式碼

src/main/resources：應用程式設定

src/test/java：測試碼

另外自行增加的目錄

MybatisGenXML：mybatis generator產出xml的位置，需自行將此目錄加入專案Source路徑，否則mybatis generator會找不到輸出目錄



1. src/main/java：系統服務化後，程式碼的概念是以模組為概念，跟以往在第一層分層controller/service/dao概念不同，可參考此篇文章<http://www.javapractices.com/topic/TopicAction.do;jsessionid=0BF4844350780B6F55476E1137FF4893?Id=205>
   1. 若是單一服務專案，通常即為單一模組，如sample範例是portal web api，則參考此範例之結構

com.fubon.portal：屬於此膜組之controller、service程式檔都放在此即可

com.fubon.portal.repository：放資料庫相關程式碼，Mybatis generator產出之程式碼在此，還是會把資料部分多放一層，其原因是model經常很多，Mybatis是以table base產出model跟mapper。

* 1. 若是系統專案，以多模組來表示package，相關controller/service都放在該package中，若多到需要再分類，也是依模組概念細分package，如
* com.fubon.doctor
* com.fubon.drug、com.fubon.drug.病種
* com.fubon.patient

1. src/main/resources：

|  |  |
| --- | --- |
| 目錄 | 說明 |
| / | 應用程式執行設定檔 bootstratp.yml：啟動設定，決定啟動環境  sample.yml：不同環境共用設定  sample -xxx.yml：不同環境各自設定  logback.yml：log設定 |
| /config.maps | * 專案執行時mybatis參考之xxxMapper.xml * mybatis generator產出之mapper放在MybatisGenXML目錄，且可能會因table調整重複generate * 人工從MybatisGenXML目錄複製內容到此(config.maps)目錄 * 因/config.maps /xxxMapper.xml還會人工修改，所以不能將mybatis generator產出目錄設在同一個路徑(所以放MybatisGenXML目錄)，否則generate時已修改內容會被蓋掉。 |
| /WEB-INF/classes/bootstrap-xxx.xml | CICD時依環境xxx包版的啟動設定檔 |

1. Web API開發方式，並整合swagger資訊
   1. Controller

* 參考PortalController分別有GET/CREATE/UPDATE/DELETE Method寫法
* 參考PortalController.findAnnounceInfoList()使用到查詢分頁PageInfo，實際請參考此程式範例寫法
  1. Service  
     PortalService承接controller要處裡的實際邏輯
  2. com.fubon.portal.repository
* Agent.java：mybatis generator產出的model
* AgentMapper.java：mybatis generator產出的跟AgentMapper.xml對應的方法介面，內容可以加入自己客製的method。

1. Mybatis使用方式
   1. 於src/main/resources/generatorConfig.xml編輯mybatis產出Mapper之設定，參考網路教學文件<https://blog.csdn.net/Thousa_Ho/article/details/70919831>
   2. 此generator設定產出之Mapper.xml存放於/MybatisGenXML中(標紅色為各專案名稱自行修改)  
      <sqlMapGenerator targetPackage=*"config.maps"* targetProject=*"portal-service/MybatisGenXML"* />
   3. 程式碼(model, mapper class)產出於(標紅色為各專案名稱自行修改)

<javaModelGenerator targetPackage=*"com.fubon.portal.repository"* targetProject=*"portal-service/src/main/java"* />

<javaClientGenerator targetPackage=*"com.fubon.portal.repository"* targetProject=*"portal-service/src/main/java"* type=*"XMLMAPPER"*>

* 1. table產出設定請參考範例及教學文件

1. 程式設定檔詳細說明
   * bootstrap.yml：設定專案名稱, 內容為dev環境, config server位址, 並依此處設定去載入相對應環境之設定檔

WEB-INF\classes\bootstrap-xxx.yml為各環境(sit, uat, prod…)之bootstrap檔，在佈署時會置換到(replace) 前述之bootstrap.yml

* + sample.yml：所有環境之共用設定，不會因環境而異動
  + sample -xxx.yml：各xxx環境之設定，如資料庫連線會不同

1. config server用途
   1. bootstrap.yml有設定config server位址
   2. 開發階段之所有設定都放在專案檔之bootstrap.yml, sample.yml, sample -dev.yml
   3. SIT/UAT/PROD環境之設定若是跨專案共用, 如murex DB連線設定, 此設定是放在config server上application.yml上
   4. 各專案也可以把自己的sample.yml, sample-sit.yml, sample-prod.yml放到config server上集中管理，目前不強制規定，但若是較多instance佈署則建議集中，才可以集中在config server上改設定。
2. Spring Boot注入之相關服務

相關服務註冊需將library(jar)加入build.gradle，請自行參考此檔，有新增library需點選build.gradle按右鍵refresh gradle即會自動下載

* 1. Health Check：library加入即可

'org.springframework.boot:spring-boot-starter-actuator'

* 1. Spring Boot Admin整合：library加入即可，可以讓Spring Boot Admin監控此服務相當多的資訊，參考協作平台『金市應用服務架構說明.pptx』有很多圖

'de.codecentric:spring-boot-admin-starter-client'

* 1. Config Server：library加入即可，可存取config server設定

'org.springframework.cloud:spring-cloud-starter-config'

* 1. Consul服務註冊：在主程式上加入@EnableDiscoveryClient，加入library為'org.springframework.cloud:spring-cloud-starter-consul-discovery'
  2. Logstash整合：library加入即可，並在logback.xml加入appender

'net.logstash.logback:logstash-logback-encoder:6.1'

* 1. 安全性：library加入即可，加入後即可使用annontation或程式碼控管web api權限，及不合法存取alert等，參考協作平台『金市應用服務架構說明.pptx』

'org.springframework.boot:spring-boot-starter-security'

* 1. Jaeger http trace：library加入即可  
     'io.opentracing.contrib:opentracing-spring-jaeger-web-starter:2.0.0'