

Spring Boot 開發規範

Spring Boot 資料夾結構

```
|--com.example.demo
    |--configuration
    |--controller
        |--advice
    |--dto
    |--entity
    |--exception
    |--model
    |--repository
    |--service
        |--impl
```

Spring Boot 檔案命名規則

每一支 API 都有相對應的交易名稱，以下範例用 XXX-C-ETNTRANST003

DTO

檔案放在 dto 資料夾內管理。

1. 上行：ETNTRANST003**Tranrq**.java
2. 下行：ETNTRANST003**Tranrs**.java
3. 自定義：

```

{
  "TRANRS": {
    "Subject": "string",
    "Content": "string",
    "FlowsList": [
      {
        "EmpId": "string",
        "FlowSeq": 0,
        "BasicInfo": {
          "BranchId": "string",
          "BranchName": "string"
        },
      },
    ]
  }
}

```

- 若records的key為 XXXList ，命名時省略List，例如：FlowsList 只要取 Flows 即可。
- 若為多層record，命名時請依序往下接，例如：
 - ETNTRANST003**Tranrs**.java
 - ETNTRANST003**TranrsFlows**.java
 - ETNTRANST003**TranrsFlowsBasicInfo**.java

Controller

檔案放在 controller 資料夾內管理。

API 的 URL 規則會依照功能分成不同 param：**col/id**、**col/name**、**list/id**、**list/name**。這四個 URL 就可以拆成兩組，Controller 也會拆成不同的檔案。

- **Col**Controller.java
- **List**Controller.java

Service 和 Service Implement

注意資料夾，Implement 是放在 Service 裡面，而非獨立出來。

- ETNTRANST003**Svc**.java
- ETNTRANST003**SvcImpl**.java

Repository 和 Entity

主要是和資料庫對接，對應 Table 名稱：EVENT_TRANS_REC

- Repository :
檔案放在 Repository 資料夾內管理。
 - EventTransRecRepository.java
- Entity :
檔案放在 Entity 資料夾內管理。
 - Entity : EventTransRecEntity.java
 - PK : EventTransRecPKEntity.java

SQL

檔案放在src/main/resources/sql

ETNTRANST003_方法名稱_001.sql

Spring Boot 開發注意事項

共通項目：

1. 程式碼撰寫規範，亦可參照 Java 開發規範
2. 程式碼務必排版
3. 全域變數上方需有註解，方法亦須標註 **JavaDoc**，並確認參數及回傳值是否正確，只有 **@Override** 不用 **JavaDoc**
4. 程式碼撰寫完成後，需要掃描 **PMD** 及 **sonarLint**

Controller

1. **@Autowired** 宣告 private
2. 使用 **@PostMapping** and **@GetMapping**

Dto

1. Inner class獨立出來
2. 檢核
 - 字串 **@NotBlank**(message = "CaseNbr 不得為空")
 - 數字 (物件型別) **@NotNull**
 - 固定長度 **@Size**(message = "UserId 長度不得超過8", max = 8)
 - 物件檢核需加上**@Valid**

Entity

1. Class annotation

- @Entity
- @Table(name = "ELC_ASSET_FACILITY_LINK_TMP")
- @IdClass(ElcAssetFacilityLinkTmpPKEntity.class) (非必要，有複合PK才需要)
- @Data
- @EqualsAndHashCode(callSuper = true)(非必要，不寫的話本身預設為true，有加@Data者也不用寫)
- 欄位 @JsonProperty必要性(非必要)

2. Entity注意要檢核 Table 中不可為空的欄位

Service Implement

1. 對資料庫有異動，須注意交易控制 @Transactional(rollbackFor = Exception.class)
2. 注意成員變數要執行緒安全(例：SimpleDateFormat非執行緒安全只能在方法中新，或改用DateTimeFormat宣告成員變數)
3. BigDecimal轉成字串需用toString
4. 丟出的 Exception 要明確，不是都丟最大的 Exception
5. 字元用單引號，不需寫成字串雙引號
6. 善用 StringUtils (判斷參數/給預設值/equals 等)
7. 查詢方式

- findById
- JPA
- Join or [] or count or in：另寫.sql檔，使用SqlUtil產生sql 給SqlAction執行
 - 多筆：getResultList()
 - 單筆：getResultList.get(0)
 - 單欄位：getSingleResult()

7. Update：findById 後直接操作 Entity，要下save

8. Insert：要先檢查ID是否存在(existById or exist(entity))，要下save

9. Delete

- deleteById (JPA)
- 目前沒有包try catch，讓底層拋Exception出來