* (前言)

早期jaxb 正常流程都是先寫xsd再透過xsd產生對應的VO

然後VO 或 STRING 再透過jaxb做marshall 或 unmarshall動作

基本上會有三塊東西

1. Jaxb 對應物件
2. Xml xsd描述檔案
3. Jaxb factory class

可以見下面的參考文章javaworld寫的

早期沒有直接支援都要command xjc下指令產生

* (現在都是entity居多 可否直接拿來用 marshall / unmarshall )

現在我們通常都是用entity

正常都希望直接entity就直接拿來轉換成json/xml

或是

Json/xml轉成entity

<<<

Entity各屬性使用 “符號”

串起來不太利於 反轉,如果結構又很複雜 更難寫

這邊我就比較喜歡xml格式儲存

>>>

以下就是針對這個目標的研究

* Java ee 7 看起來restful

我目前測試

應該就直接有支援json/xml

Marshall / unmarshall

With entity

但因為預設產生的entitybean 寫的xml @annotation太簡略

有些比較複雜結構的entity

個人是持保持態度

可能要測 或 自己加工

* 環境介紹 jdk 1.7 ,java ee 7

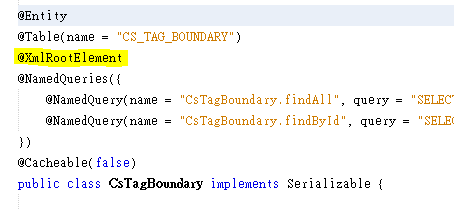
1. 使用eclipselink moxy

基本上moxy本身就包含在eclipselink.jar之中

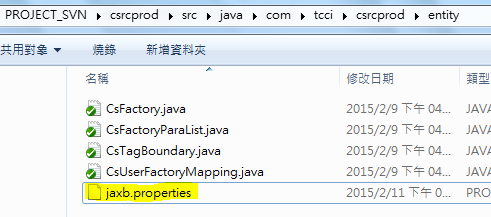
我這邊是使用eclipselink.jar version 2.5.2

1. Entity如果產生時有勾選jaxb xxx選項

會看到下圖黃色處



1. Entity所在目錄記得建立jaxb.properties

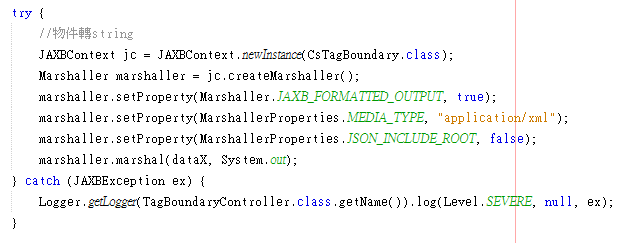


1. jaxb.properties加上一行

javax.xml.bind.context.factory=org.eclipse.persistence.jaxb.JAXBContextFactory

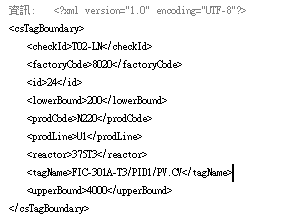
//表示使用eclipselink 作為jaxbfactory

* 作法DEMO如下
* Marshall entity to xml



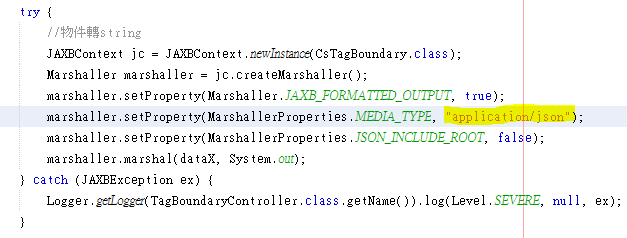
(PS.dataX 是entity)

結果如下



* Marshall entity to json如下

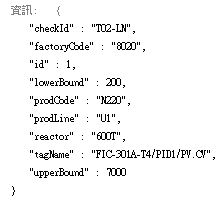
其實把media\_type改為json即可



(PS.dataX 是entity)

結果

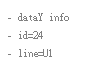
以下case因為不包含root所以不會看到csTagBoundary物件



* Unmarshall xml to entity



結果



* Unmarshall json to entity

Unmarshaller 預設是jaxb 因此json要刻意寫指定格式



結果



* 延伸閱讀1:參數JSON\_INCLUDE\_ROOT

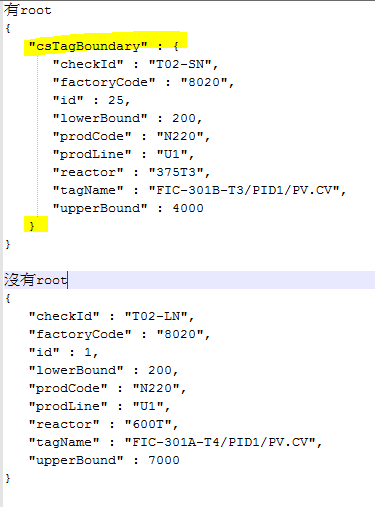
差異可以看物件轉json

發現物件root會沒有

當JSON\_INCLUDE\_ROOT設為false

這邊會連到影響到unmarshall也要清楚目前用的json

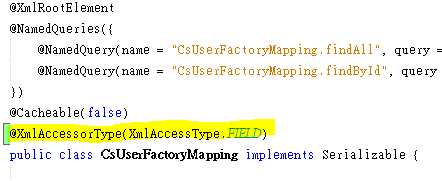
到底有沒有root

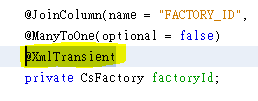


* 延伸閱讀2 :如果entity裡面有不想要在xml/json出現的東西

1. Class上面加上 @XmlAccessorType(XmlAccessType.FIELD)
2. Property上面加上 @XmlTransient

這個屬性就會在xml/json 產生時忽略



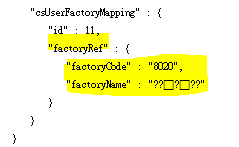


* 延伸閱讀3:如果entity裡面如果還有entity

使用XmlElement註記資訊



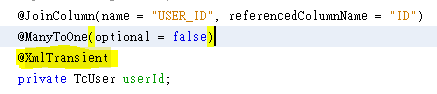
[PS.被註記的entity要有@XmlElementRoot](mailto:PS.被註記的entity要有@XmlElementRoot)宣告 ,沒有宣告 加了也沒用...



上圖黃色區塊就是entity跟它裡面的屬性 變成json物件

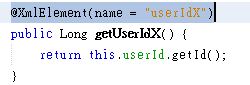
* 延伸閱讀4: 如果entity下的sub entity有cycle問題

被disable 該如何顯示資訊

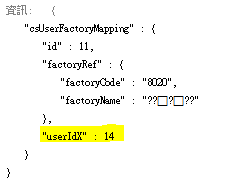


自己寫getMethod取得自己想要的資訊格式

記得method上面加上@XmlElement(name = "userIdX")



結果



* (補充說明)目前我們慣用的結構 tc\_user很容易 造成cycle

Master=>detail=>detail.creator=>tc\_user.creator => loop

變成jaxb/json要注意cycle狀況

* (補充說明)Netbeans’s目前generate entity xml annotation只有很簡單的宣告

@XmlRootElement

基本上比較陽春一點

Xml這邊其實也有檢核功能 預設值

當物件時 轉換成 xml時就可以派上用場

基本上要看寫的用得如何

* 參考文章

1. <http://www.javaworld.com.tw/jute/post/view?bid=19&id=293455>
2. <https://netbeans.org/kb/74/websvc/jaxb.html>
3. <http://www.developer.com/java/xml-processing-in-java-ee-7.html>
4. <http://www.techferry.com/articles/jaxb-annotations.html#Using-JAXB-JPA-Annotations-in-conjunction>

* JAXB要注意結構 ref 不然很容易導致循環錯誤

A=>B=>C 然後C=>A

這樣的問題 會導致xml/json會產生不出來

<http://www.ibm.com/developerworks/rational/library/resolve-jaxb-cycle-errors/index.html>

* 如有興趣 建議看api

<http://docs.oracle.com/javaee/7/api/>

[javax.xml.bind.annotation](http://docs.oracle.com/javaee/7/api/javax/xml/bind/annotation/package-summary.html)

<http://docs.oracle.com/javaee/7/api/javax/xml/bind/annotation/package-summary.html>

* JPA ENTITY有@專門處理entity 與entity關係

Jaxb基本也有類似的東西 要稍微注意一下

* (應用)

正常我們紀錄資料都比較喜歡用log table

Mirror資料過去

長得很像的額外table

優點是好處理

但如果一堆功能都要log也就會導致要建一堆的log table

這邊如果我們把entity

單純使用符號串起來 有沒有結構處理還不知道

事後反轉會很不好處理

就變成真的是紀錄而已

但如果我們能夠直接entity轉成 “有結構性”的xml/json字串

反轉也就不會太困難

* (思考問題)

基本上entity轉換成xml/json某些情況下

雖然可以往下整個攤開(全部相關資訊結構)

但基本上其實可以考慮就第一層帶個id就好了

如果只是參照使用…不一定要資料那麼大