Egg再問 所以看了一下 但我只有用過樂觀鎖就是了….

麻煩有誰玩一下悲觀鎖在分享看看

* 樂觀鎖 使用@version

<http://docs.oracle.com/javaee/6/tutorial/doc/gkjhz.html#gkjjc>

<http://openhome.cc/Gossip/EJB3Gossip/Lock.html>

* 好處是DB不會有很大effert

缺點就是version版本號不對的就是給你錯誤訊息而以

* 使用者需要常常的刷新資料

及面對要更新時才發現被人給更新了

要重拿資料再來編輯

<<<如何使用>>>

簡單來說就是table要多加一個欄位

該欄位可以是數字 數值 日期…

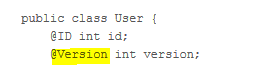
如下

int, Integer, long, Long, short, Short, or java.sql.Timestamp

entity對應的地方

要加上@version annotation

如下圖



<<<效果如下>>>

@version所註記的欄位 update資料前

會先檢查目前DB 版本號

“是否一致”

如果version號碼一致

例如version 99(手上資料) = version 99(DB資料)

就會給更新

更新之後呢 就會自動的把version號碼+1

以上情境呢就會變成version 100號

如果手上持有的資料呢去更新資料時 發現版本號不一致

例如

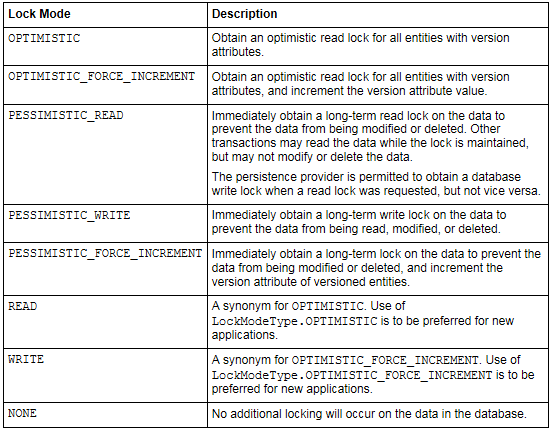
手上是100號 但要更新前 發現已經被變成了105號

JPA就會丟出OptimisticLockException

* Lockmode介紹 與 悲觀鎖

<http://docs.oracle.com/javaee/6/tutorial/doc/gkjiu.html>

下圖是LOCK MODE在JPA有哪些



JPA lockmode分兩塊

* 樂觀(optimistic)

1. Optimistic
2. Optimistic\_force\_increment
3. Read(早期之前寫法 等價第一個 新案子請改用第一個)
4. Write(早期之前寫法 等價第二個 新案子請改用第二個)

* 悲觀(pessimistic)

1. pessimistic\_read
2. pessimistic\_write
3. pessimistic\_force\_increment

* 再來看Pessimistic Read / write / force\_increment 是講甚麼

Pessimistic Read基本上是講給read 但不給update/delete

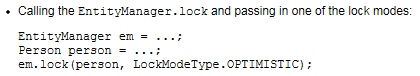
Pessimistic Write 基本上是講 不給read/update/delete

Force\_increment基本上是講 不給update/delete然後直接增加version號碼

(不過前提是要實作optimistic lock機制)

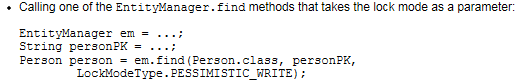
* LOCK MODE 如何被觸發

1. 呼叫em.lock



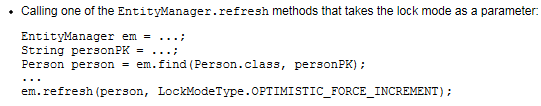
Entity先find出來 再用em.lock鎖定

1. Em.find的時候就順便 決定lock了

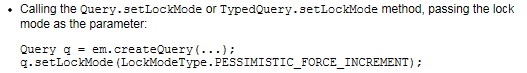


Find還有多一個method就是可以傳參數

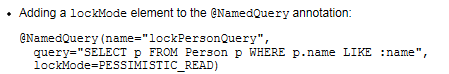
1. Em.refresh 事後 刷撈出來的entity



1. Query條件下 就決定lock mode



1. Namequery時就決定 lock mode



* 悲觀鎖記得要設定過期時間

取資料時 如果搶不到 起碼會timeout掉

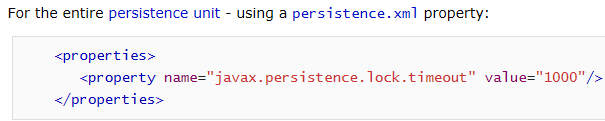
不會一直等在那邊

後面對應的sql 就會變成

select \* from xxx for update wait timeout\_second

哪邊可以設timeout

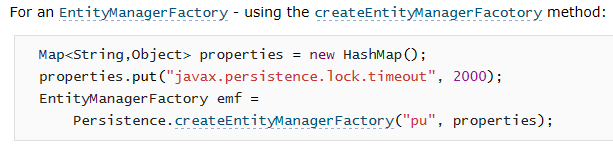
1. persistence.xml



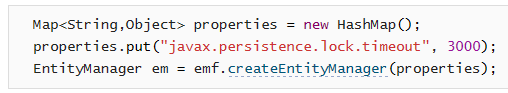
1. EMF (entity manage factory)

不過我們目前的CODE 直接拿到em

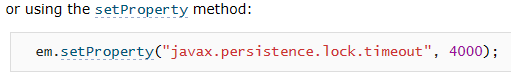
因此不太會這樣用…



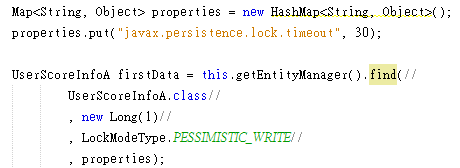
1. EM (entity manager)



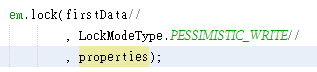
Or



1. 資料讀取時 em.find 給property



1. 先撈資料 再用lock鎖定時 傳入參數



firstData 是先find出來的entity

properties同上面第四點

* 使用心得
* 樂觀鎖

1. 樂觀鎖 如果沒有刻意處理 基本上@version即可
2. 樂觀鎖 如果要刻意處理 可以em.find, em.lock

使用optimistic\_force\_increment

正常只有@version基本上是merge時version才加一

但force\_increment時 就直接對象 加一

* 悲觀鎖

1. 悲觀鎖基本上version可有可無

但version機制有加上時 如果有違反樂觀鎖version 檢查

就會先丟optimistic exception

1. 悲觀鎖 基本上要利用db 來控管

如果有印jpa log 就會發現

後面sql其實是用

(oracle)select \* from xxx where yyy for update

(ms sql server) select \* from xxx where yyy for update nowait

1. 然後在transaction內 如果有別的transaction取用同一個 entity

使用的lockmode是不允許的

就會拋出PessimisticLockException

我這邊理解是 如果有entity 被宣告lock(xxx,LockMode.Pessimistic\_Write)

等於是

Pessimistic\_READ, Pessimistic\_Write如果後面有人拿來用

就會噴錯

1. 悲觀鎖 釋放可以使用

Lock(xxx.LockMode.None)

1. 悲觀鎖 要注意bottleneck,dead lock問題
2. 悲觀鎖 建議設定timeout會比較好

下面幾個地方都可以設定

* Persistence.xml
* Em(entityManager)
* Em.Find or em. lock use property

1. 悲觀鎖 基本上測試不太好寫

因為要

不同的em然後不同的transaction

針對同一個entity lock

如下

Ex.

Em1(entityA1,LockMode.Pessimistic\_xxxx)

Em2(entityA2,LockMode.Pessimistic\_xxxx)

entityA1.setXXX

entityA2.setXXX

但目前專案都是使用JTA控管

JTA不讓使用transaction使用

參考文章

1. <http://docs.oracle.com/javaee/6/tutorial/doc/gkjhz.html#gkjjc>
2. <http://docs.oracle.com/javaee/6/tutorial/doc/gkjiu.html>
3. <http://openhome.cc/Gossip/EJB3Gossip/Lock.html>
4. <http://en.wikibooks.org/wiki/Java_Persistence/Locking>
5. <http://www.coderanch.com/t/645005/OCEJPA/certification/trigger-javax-persistence-PessimisticLockException>
6. <https://blogs.oracle.com/enterprisetechtips/entry/locking_and_concurrency_in_java>
7. <http://www.objectdb.com/java/jpa/persistence/lock>