**FEP專案編碼規範**

2023/03/23

Copyright the SYSCOM Computer Engineering Co.2023, All Rights Reserved

Revision History

|  |  |  |
| --- | --- | --- |
| Date | Author | Remark |
| 2021/05/10 | YU, Richard | Initial Version |
| 2021/05/13 | YU,Richard | 增加MethodBase.GetCurrentMethod()章節。  增加RefBase<T>的使用章節。 |
| 2021/05/20 | YU,Richard | 增加BitConverter.ToString章節。  增加Mybatis事務的使用章節  增加NET中struct物件改寫注意事項章節 |
| 2021/05/12 | YU,Richard | 修改StringLib和StringUtil章節 |
| 2021/05/24 | YU,Richard | 增加FeptxnDao的使用章節 |
| 2021/05/28 | YU,Richard | 增加VB.NET中的isNumeric方法章節 |
| 2021/06/24 | YU,Richard | 增加枚舉類的使用注意事項章節 |
| 2021/08/10 | YU,Richard | 增加字串拼接章節 |
| 2021/09/22 | YU,Richard | 增加C#中的decimal變數ToString格式化章節  增加Mybatis使用Map作為查詢結果章節 |
| 2022/08/10 | YU,Richard | 修改[Mybatis事務的使用](#_Mybatis事務的使用) |
| 2023/03/23 | YU,Richard | 修改[Mybatis事務的使用](#_Mybatis事務的使用) |

# Catalog

[Catalog 3](#_Toc97208765)

[Concept 5](#_Toc97208766)

[遵照Java編碼規範 6](#_Toc97208767)

[資料庫欄位元Boolean和Short 6](#_Toc97208768)

[枚舉類 7](#_Toc97208769)

[枚舉類Parse方法 7](#_Toc97208770)

[日期/數值格式化 8](#_Toc97208771)

[SubString 8](#_Toc97208772)

[StringLib和StringUtil 8](#_Toc97208773)

[.NET中參數帶ref\out或者ByRef標示 9](#_Toc97208774)

[String類型的比較 11](#_Toc97208775)

[CalendarUtil 11](#_Toc97208776)

[PolyfillUtil 11](#_Toc97208777)

[C#中的String.IsNullOrEmpty 11](#_Toc97208778)

[RuntimeException 12](#_Toc97208779)

[使用/修改Mybatis的Mapper和XML檔 12](#_Toc97208780)

[存儲過程和自訂SQL 12](#_Toc97208781)

[MyBatis的XML中SQL只查詢返回第一筆資料的寫法 13](#_Toc97208782)

[字串編碼轉換 13](#_Toc97208783)

[Base64轉換 14](#_Toc97208784)

[FEPTXN動態表名 14](#_Toc97208785)

[C#中decimal的格式化 14](#_Toc97208786)

[電文Request對象 15](#_Toc97208787)

[MethodBase.GetCurrentMethod() 17](#_Toc97208788)

[RefBase<T>的使用 18](#_Toc97208789)

[BitConverter.ToString 20](#_Toc97208790)

[Mybatis事務的使用 20](#_Toc97208791)

[.NET中struct物件改寫注意事項 21](#_Toc97208792)

[FeptxnDao的使用 23](#_Toc97208793)

[VB.NET中的isNumeric方法 24](#_Toc97208794)

[枚舉類的使用注意事項 24](#_Toc97208795)

[字串拼接 25](#_Toc97208796)

[Mybatis使用Map作為查詢結果 26](#_Toc97208797)

[C#中的decimal變數ToString格式化 26](#_Toc97208798)

# Concept

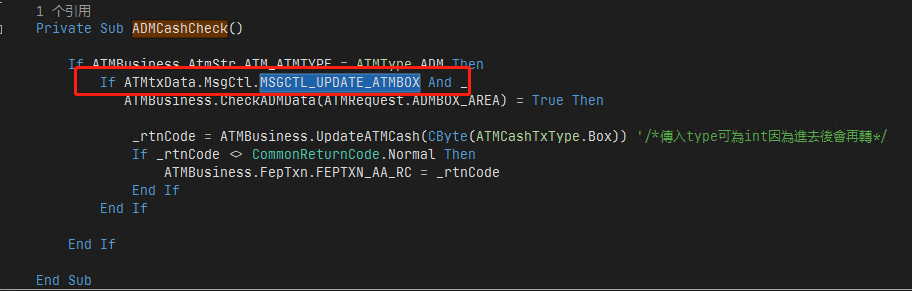
本文檔旨在闡述，本專案的Java編碼規範，以及原.NET版程式轉Java版程式中需要注意的事項，以及實作方式。

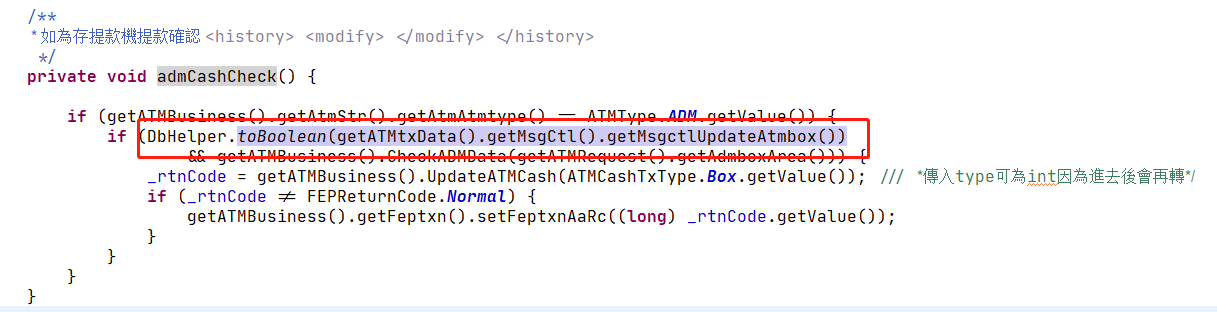
# 遵照Java編碼規範

參考[Code Conventions for the Java TM Programming Language](https://www.oracle.com/java/technologies/javase/codeconventions-contents.html)

# 資料庫欄位元Boolean和Short

.NET中Table對應Entity中欄位是Boolean類型的，在Java中的Mybatis Model是Short型，則需要使用DbHelper進行Boolean和Short的轉換。





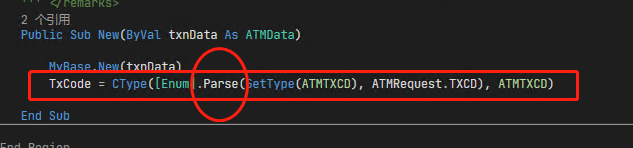
# 枚舉類

每個單獨的枚舉類全部放在fep-vo工程中的，com.syscom.fep.vo.enums下，除了枚舉中定義的變數名外，餘下部分的代碼請改為下圖的寫法(枚舉類的名字請改為對應正確的名字)。



# 枚舉類Parse方法

如下圖，.NET中Enum.Parse方法，在Java中改為使用對應枚舉類的parse方法。



# 日期/數值格式化

.NET中有些DateTime.Now.ToString("yyyy-MM-dd")請使用fep-frmcommon下的FormatUtil。

# SubString

.NET中SubString(Start, Length)該方法第二個參數是Length，所以轉到java中要注意為substring(Start, Start + Length)。例如：

.NET中，SubString(4, 6)，轉到Java則是substring(4, 10)，即第二個參數為4 + 6 = 10。

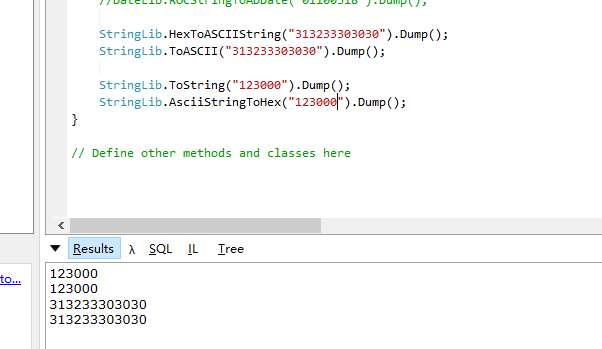
注意，如果是用轉換工具轉出的代碼，會自動轉好，無需修改。

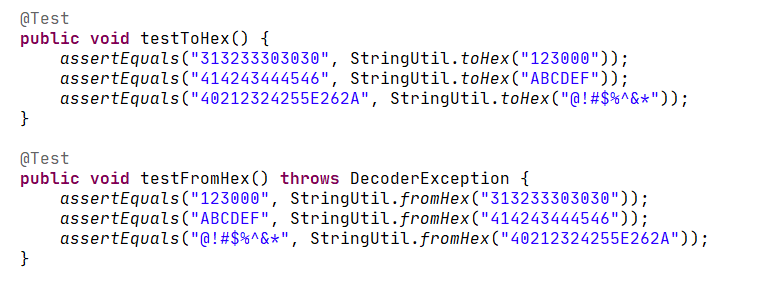
# StringLib和StringUtil

以下.NET版StringLib中的方法，對應使用StringUtil中的方法。

StringLib.ToString/AsciiStringToHex --> StringUtil.toHex

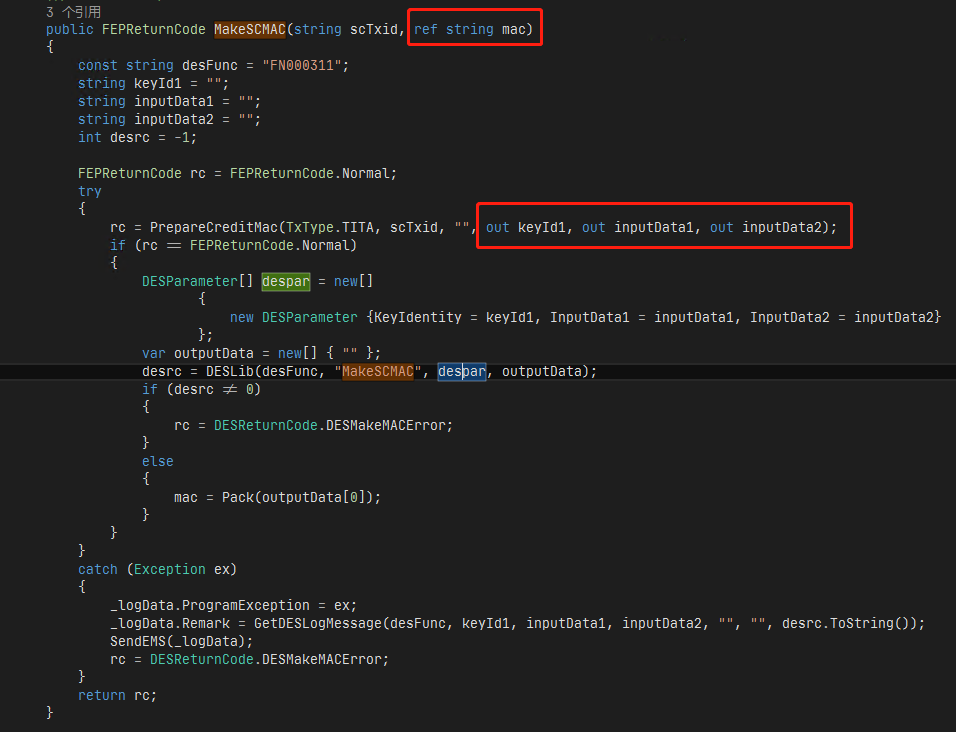
StringLib.ToASCII/HexToASCIIString --> StringUtil.fromHex





# .NET中參數帶ref\out或者ByRef標示

Java中除了物件可以作為參數按照位址傳遞，其他基本物件包括String都只能按照值傳遞。所以，類似下圖.NET中有使用ref，或者out標示的參數，在Java中請使用fep-frmcommon中com.syscom.fep.frmcommon.ref下的類。





# String類型的比較

.NET中String類型直接用比較符號進行比較的，在Java中要改為equals，並且注意將被比較的字串物件，或者是字串常量放在equals的左邊，這樣可以避免出現NullPointException。例如：

C#：atmRes.Substring(0, 6) != "0F0F0F"

java： !"0F0F0F".equals(atmRes.substring(0, 6))

# CalendarUtil

位於fep-frmcommon下，主要是Calendar日期操作相關方法，同時提供西元年🡨🡪民國年相互轉換的方法，對應.NET中的DateLib下的西元年🡨🡪民國年轉換方法。

注意，CalendarUtil中rocStringToADDate和adStringToADDate，當傳入的參數不合法時，轉出來的值為null，所以在使用這兩個方法時，要注意判斷方式

# PolyfillUtil

位於fep-frmcommon下，主要是建立.NET中特有的方法。

# C#中的String.IsNullOrEmpty

對應使用org.apache.commons.lang3下的，StringUtils.isBlank，或者StringUtils.isNotBlank。

# RuntimeException

如果是用轉換工具轉出的代碼，會自動將.NET的Exception轉為Java的RuntimeException，故請改為Exception。

# 使用/修改Mybatis的Mapper和XML檔

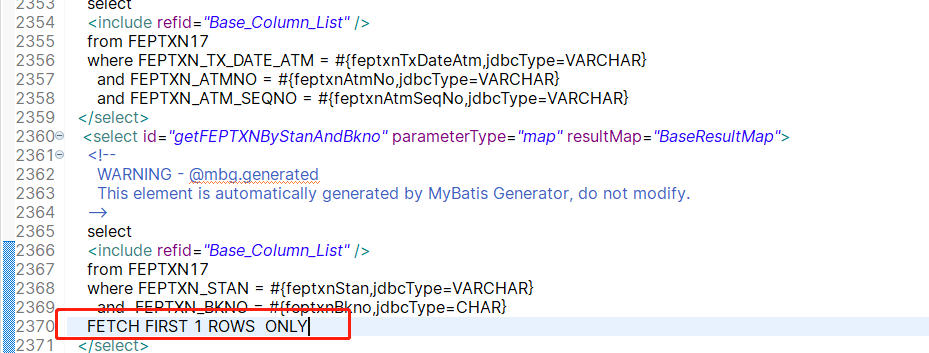
原有生成的代碼請不要修改，如果需要請自行添加新的代碼，並append到檔案的最後，並添加上注釋，例如2021-05-10 Richard add

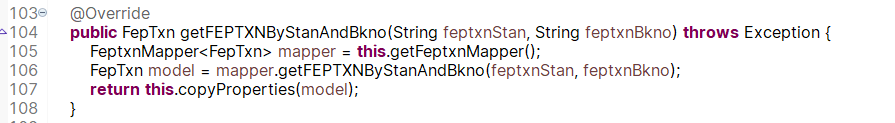
Mybatis的mapper中，注意insert和update，還各自有一個尾碼是Selective的方法，區別是，帶Selective尾碼的方法，會判斷每個insert或者update的欄位是否為null，如果不為null才insert或者update

# 存儲過程和自訂SQL

AamybatisMapper類，專門用於添加處理一些複雜的sql，以及存儲過程

# MyBatis的XML中SQL只查詢返回第一筆資料的寫法

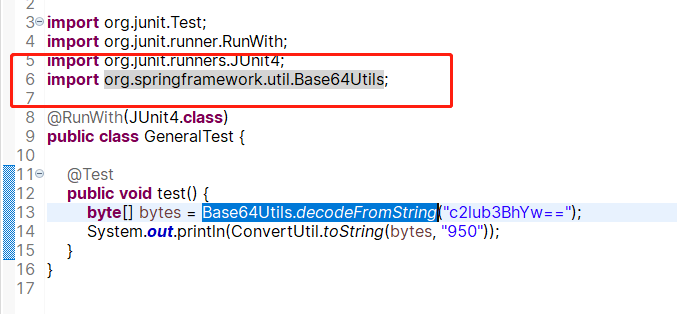




# 字串編碼轉換

net中有Encoding.GetEncoding("Unicode")，或者Encoding.Unicode，在java中要對應使用PolyfillUtil.toCharsetName("Unicode")方法取charsetName，用於轉string或者是轉bytes，因為.net中unicode編碼，相當於是java中的UTF-16LE

# Base64轉換



# FEPTXN動態表名

Feptxn表的操作，原本31個mapper和xml，現在合併為一個，採用的是動態傳入表名的方法，後面再新增方法時，請看參考已經寫好的mapper和xml

# C#中decimal的格式化

\_fepTxn.FEPTXN\_TX\_AMT.ToString("0.00").Replace(".", "")

可以使用

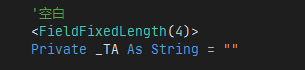
MathUtil.toString(this.feptxn.getFeptxnTxAmt(), "0.00")

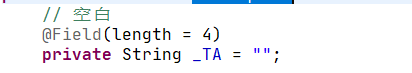
或者

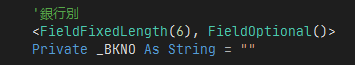
MathUtil.toString(this.feptxn.getFeptxnTxAmt(), 2)

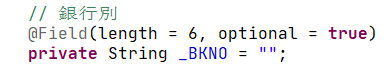
# 電文Request對象

注意以下C#和Java欄位元注入方式寫法不同。

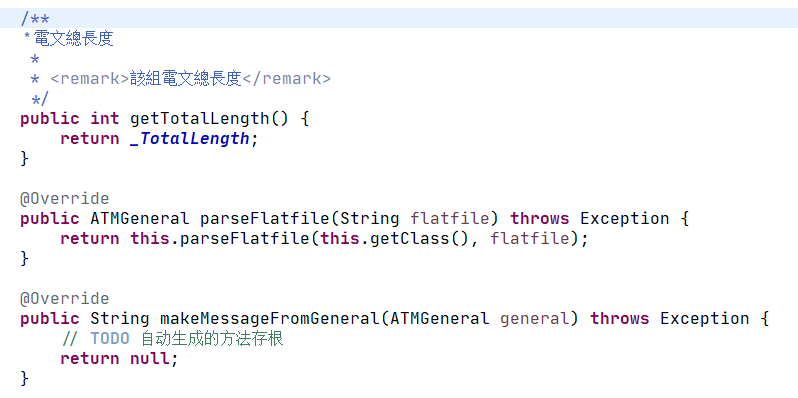








如下圖，前兩個方法必須實作，makeMessageFromGeneral理論上不需要實作，因為Request物件基本上是由其他系統接收到的16進制字串形式的電文轉換後的物件，所以基本上是不需要再由物件轉成字串。



toGeneral方法必須實作，但是要注意電文中部分欄位元需要根據電文規格書中的描述，調整為實際的資料類型。因為在C#程式中，數值類的欄位元全部都定義為了decimal，而用工具轉換後都變成了BigDecimal，所以Java中需要調整為int或者double。另外，注意以下寫法：

|  |
| --- |
| C#：  .RATE = CDec(IIf(Decimal.TryParse(StringLib.ToASCII(\_RATE), .RATE), .RATE / 10 ^ 5, 0)) '匯率  .CHARGE = CDec(IIf(Decimal.TryParse(StringLib.ToASCII(\_CHARGE), .CHARGE), .CHARGE / 10 ^ 2, 0)) '手續費 (台幣) |

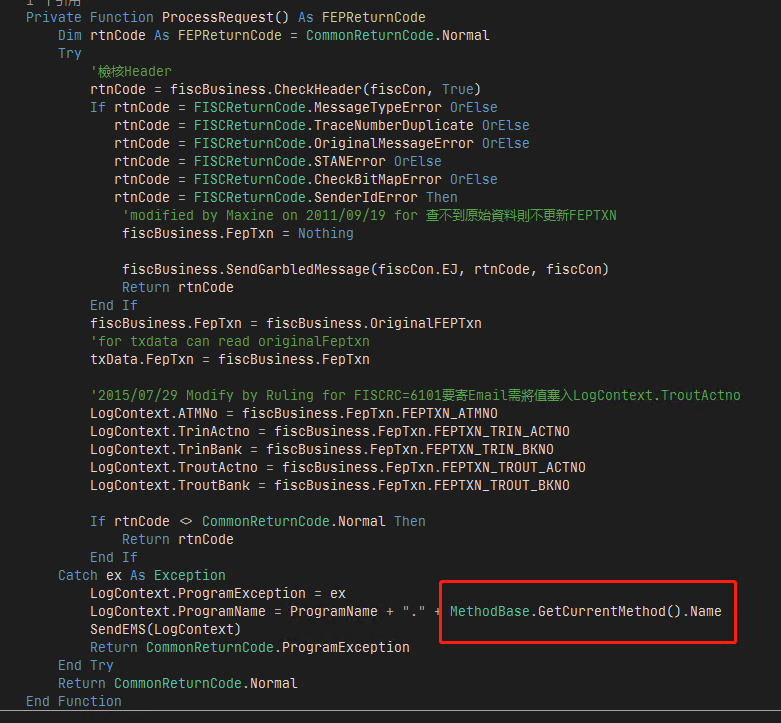
|  |
| --- |
| Java：  request.setRATE(this.toDouble(this.getRATE(), 5));// 匯率  request.setCHARGE(this.toDouble(this.getCHARGE(), 2)); // 手續費 (台幣) |

具體請參考fep-vo下的com.syscom.fep.vo.text.atm.request.IWDRequest，對應C#下的Common\Text\ATMText\Request\IWD\_Request。

# MethodBase.GetCurrentMethod()

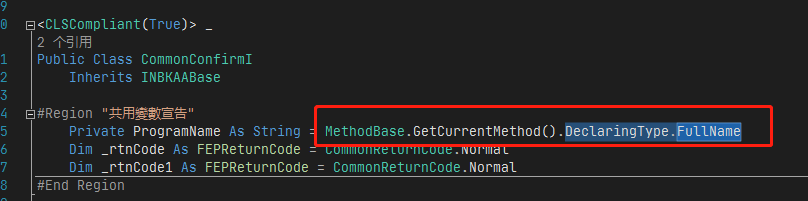
在.NET方法中，遇見MethodBase.GetCurrentMethod().Name這個代碼，在Java裡請改成當前所在的方法的名字。

例如下圖，紅框部分的代碼，在java中直接改為"ProcessRequest"。



在.NET方法中，遇見MethodBase.GetCurrentMethod().DeclaringType.FullName這個代碼，在Java裡請改成當前類的名稱。

例如下圖，紅框部分的代碼，在java中直接改為private static final String ProgramName = CommonConfirmI.class.getSimpleName();。



# RefBase<T>的使用

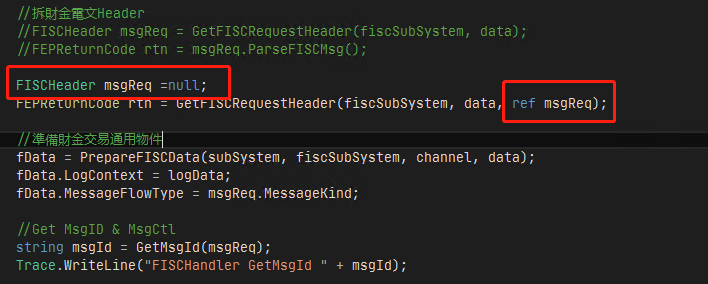
前面[.NET中參數帶ref\out或者ByRef標示](#_.NET中参数带refout或者ByRef标示)有簡單介紹，.NET中方法參數按照位址傳遞的處理方式，這裡將講述如何使用RefBase<T>實現.NET中，方法中的物件參數按照位址傳遞的處理方法。

如下圖，msgReq是按照位址傳遞的，那麼方法GetFISCRequestHeader中會給msgReq賦值。

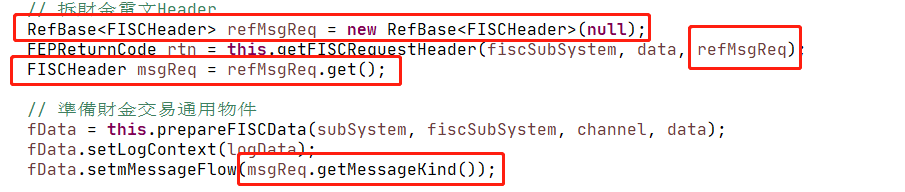
在Java中，當JavaBean作為方法參數時，實際上是將JavaBean的引用進行傳遞，也就是說，方法內只可以改變JavaBean的內部屬性的值，而不能改變JavaBean本身。

簡單的說：

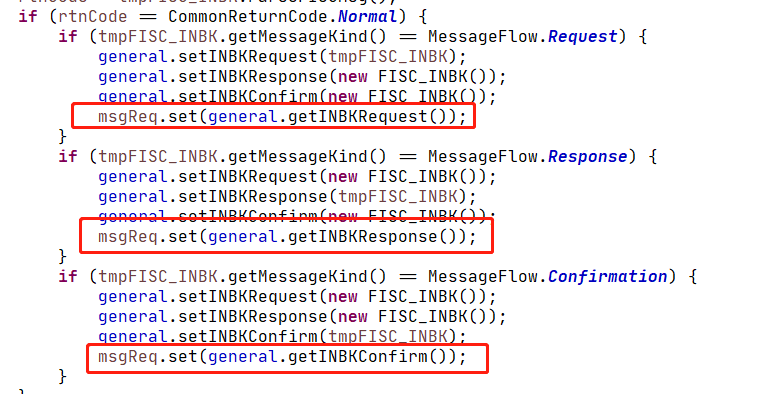
1. 如果沒有初始化JavaBean物件，也就是JavaBean以null傳入到方法中，則該方法內無論再怎麼給JavaBean賦值，則經過該方法後，該JavaBean還是null。
2. 如果有初始化JavaBean物件，則傳入方法中，該方法內部可以修改JavaBean的屬性值，但是無論怎麼重新給JavaBean賦值，則該JavaBean是不會改變的。



如下圖，是Java修改後的代碼，定義一個RefBase，由於.NET中初始化是null，那麼這裡RefBase的構建函數中也是塞入null。然後傳入方法中，方法過後第三個紅框處，將原本的FISCHeader msgReq通過RefBase.get()取出，繼續供後面的代碼使用。



然後，下圖是傳入的方法內部，紅框內通過RefBase.set()方法，將需要重新賦值的物件塞入(.NET中是直接用=賦值)。



# BitConverter.ToString

如果看到.NET程式中有用到：

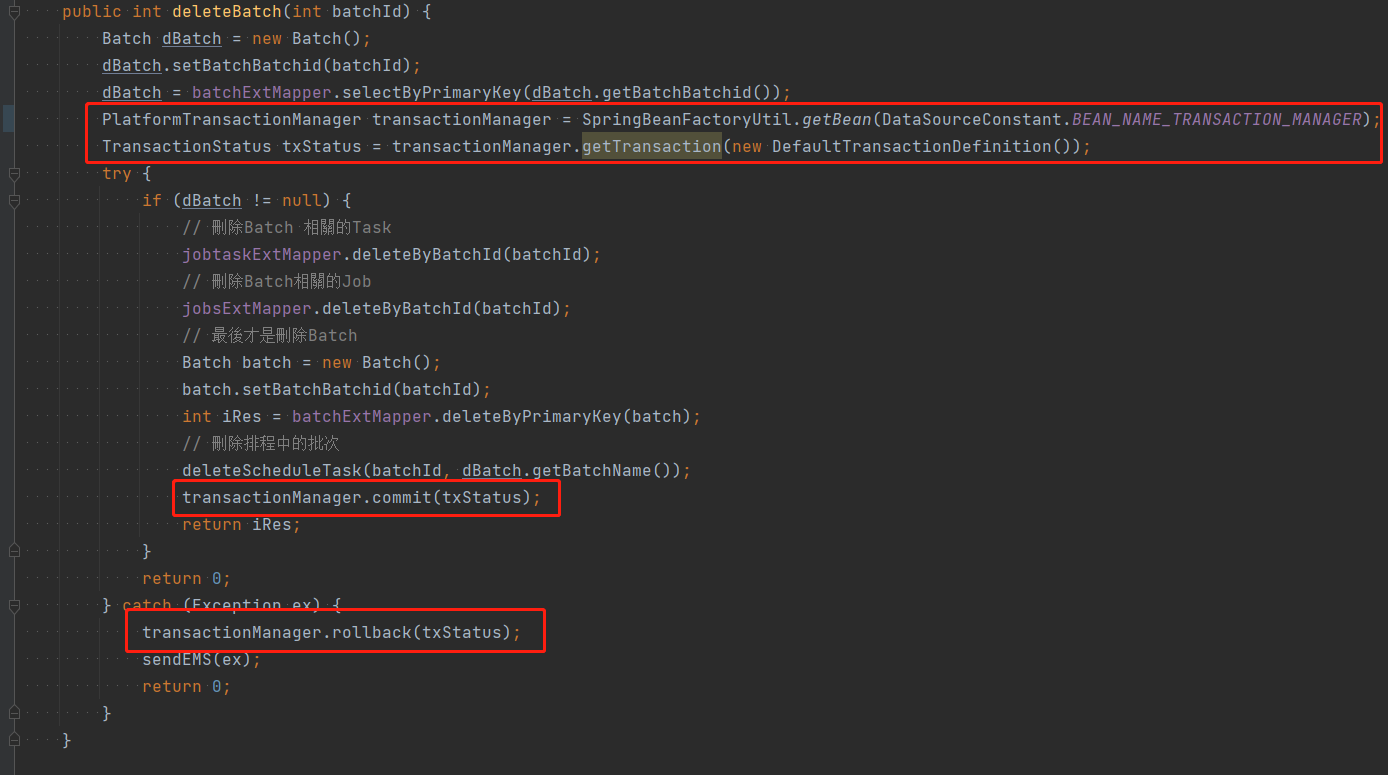
BitConverter.ToString(bytes, 0, length).Replace("-", "")

Java版請使用fep-frmcommon：

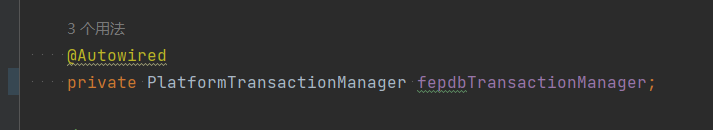
ConvertUtil.toHex()方法。

# Mybatis事務的使用

如果需要使用Mybatis的事務，可以參考以下示例，注意紅框中的三行代碼和位置：



当然，在Spring Bean物件中，可以通过下面的方式获取fepdbTransactionManager。

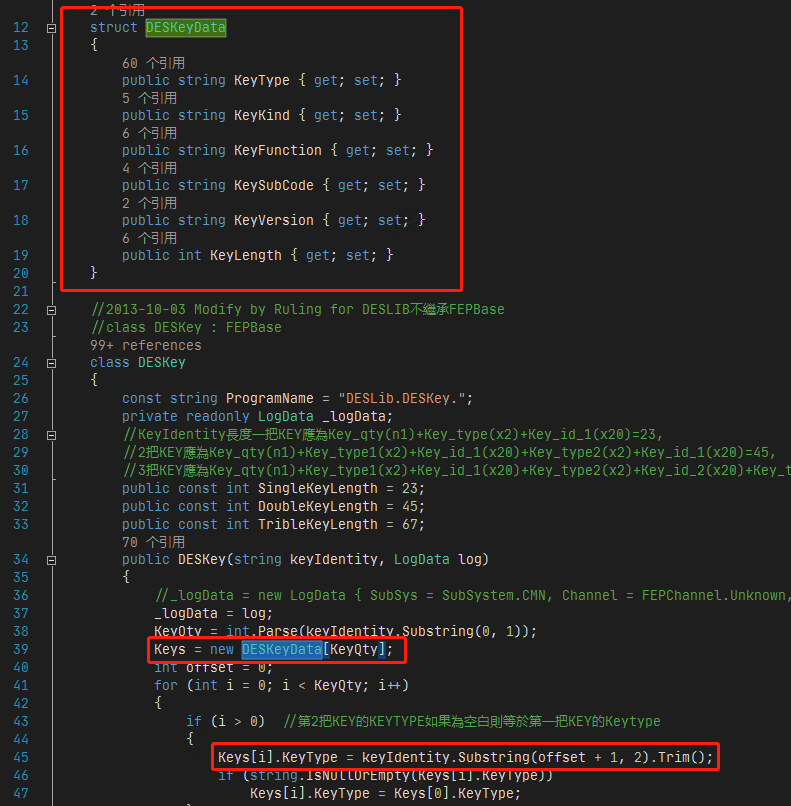


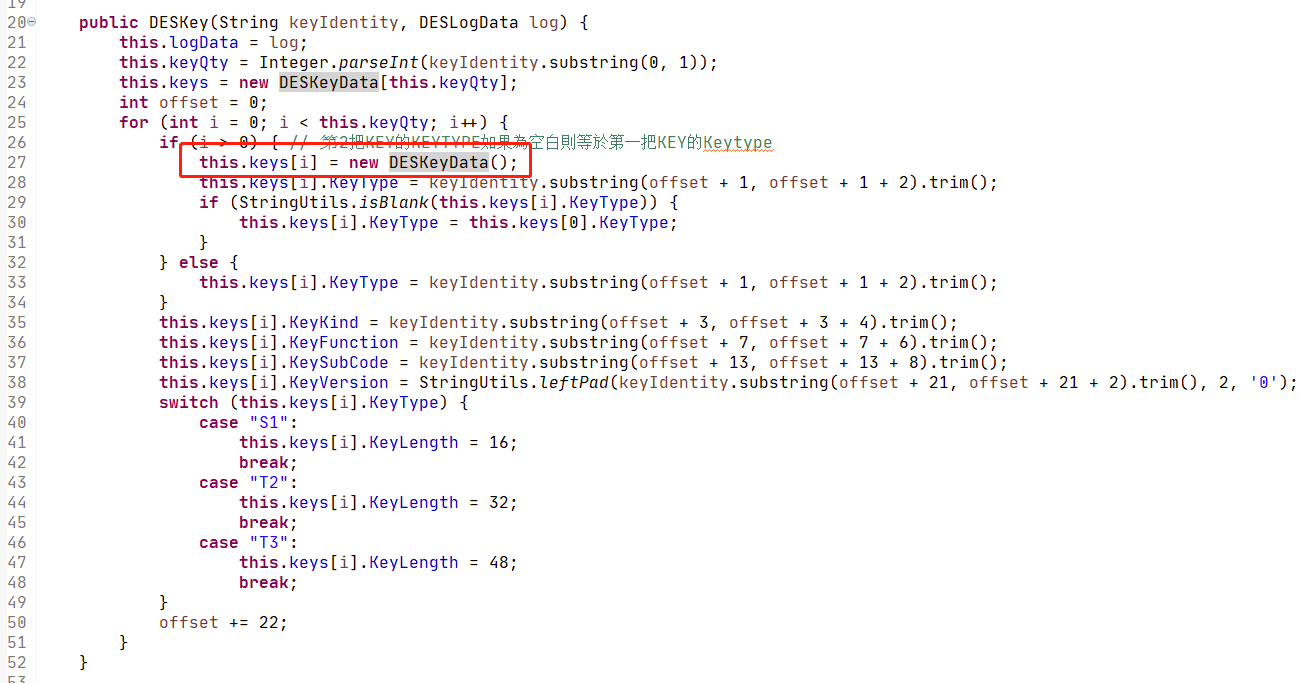
# .NET中struct物件改寫注意事項

.NET中的struct物件，在Java中一律改寫為class物件，但是在使用上一定要注意，.NET中struct物件，無論是單一實例使用，還是用於陣列中，都不需要進行new，如下圖。

第二個紅框處，new了一個陣列物件，但是在第三個紅框處，並沒有再new陣列中的每一個成員，而直接使用，就是因為這個物件是一個struct物件。

所以，在改寫到Java時，一定要給第三處紅框前，先要new陣列中的每一個成員。

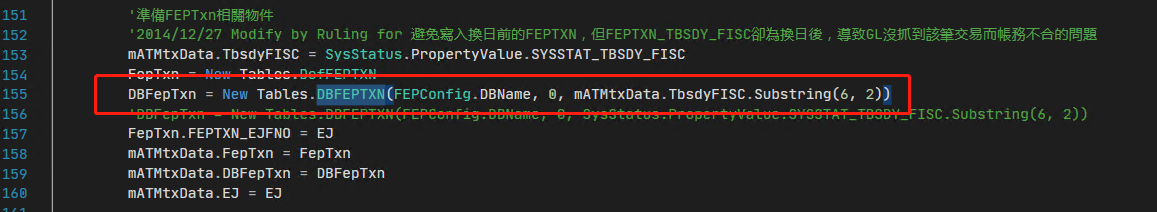




# FeptxnDao的使用

FeptxnDao通過SpringBeanFactoryUtil.getBean("feptxnDao")來獲取實例，並且每次通過SpringBeanFactoryUtil獲取的是新的實例。

在獲取到FeptxnDao之後，一定要先setTableNameSuffix方法塞入tableNameSuffix，從而確保將資料寫入正確的FEPTXNXX(XX為01~31)表中。例如下圖：



則相當於是，fepDao.setTableNameSuffix(this.mATMtxData.getTbsdyFISC().substring(6, 8))。

# VB.NET中的isNumeric方法

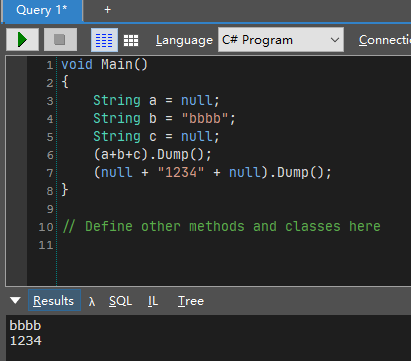
在Java這邊請使用PolyfillUtil.isNumeric方法。

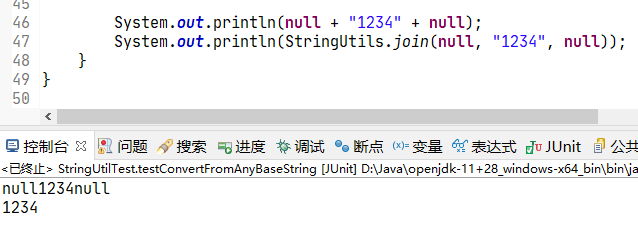
# 枚舉類的使用注意事項

|  |  |  |
| --- | --- | --- |
| DB中欄位 | 枚舉類 | 比較或賦值使用的枚舉方法 |
| FEPTXN.FEPTXN\_PCODE | FISCPCode | getValueStr() |
| FEPTXN.FEPTXN\_CHANNEL  NWDTXN.NWDTXN\_REG\_CHANNEL | FEPChannel | name()，或者toString() |
| CHANNEL.CHANNEL\_CHANNELNO  MSGFILE.MSGFILE\_CHANNEL | FEPChannel | getValue() |
| FEPTXN.FEPTXN\_TX\_CODE  FEPTXN.FEPTXN\_TX\_CODE\_ATMC  ATMC.ATMC\_TX\_CODE  MSGCTL.MSGCTL\_MSGID | ATMTXCD | name()，或者toString() |
| FEPTXN.FEPTXN \_ZONE\_CODE  ATMSTR.ATM\_ZONE | ATMZone | name()，或者toString() |
| FEPTXN.FEPTXN\_TX\_CUR\_ACT  ZONE.ZONE\_CUR | CurrencyType | name()，或者toString() |
|  |  |  |

# 字串拼接

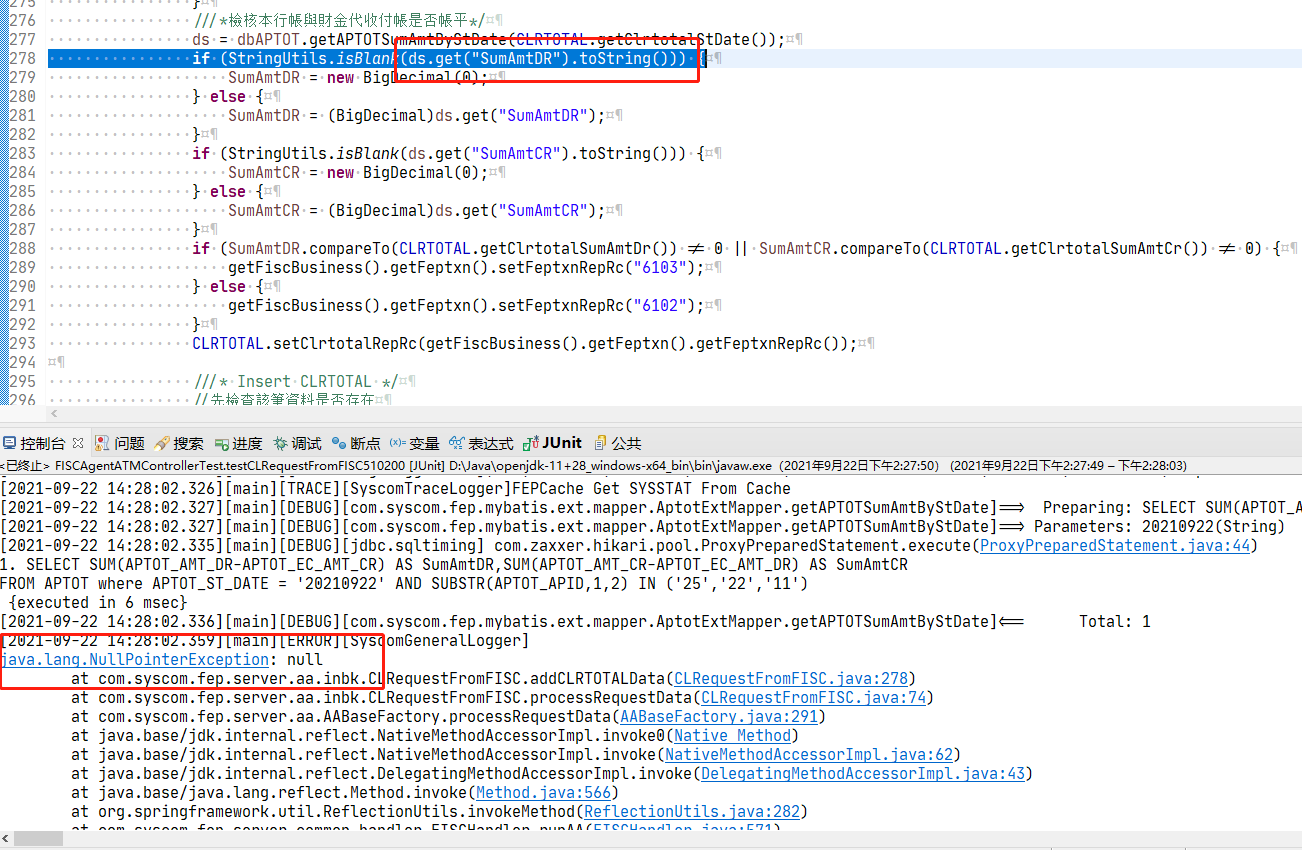
字串拼接儘量使用StringUtils.join，並用逗號隔開。尤其是在組建電文字串中，一定要用StringUtils.join，不要使用+拼接字串。



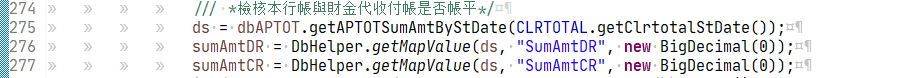


# Mybatis使用Map作為查詢結果

當使用Map作為查詢結果集合時，在取查詢結果欄位值並使用前，一定要判斷是否存在，否則可能會引發NullPointException。

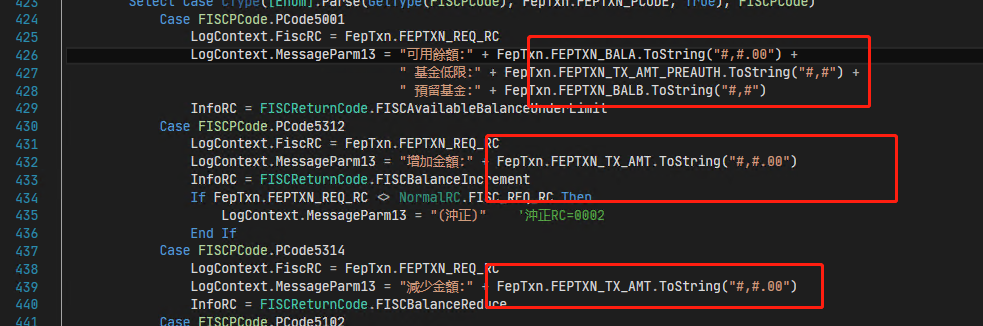


建議使用DbHelper.getMapValue方法取值，修改後如下：



# C#中的decimal變數ToString格式化

C#代碼如下圖



請對應使用下圖的MathUtil.toString方法，注意在某些場合下，removeDot參數的使用。

