**投資系統-程式碼結構**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=IM_CodeStructure#section-IM_CodeStructure-_E6_8A_95_E8_B3_87_E7_B3_BB_E7_B5_B1_E7_A8_8B_E5_BC_8F_E7_A2_BC_E7_B5_90_E6_A7_8B)

* EBAF架構overview - [觀看完整大小圖片](http://ws93006ed:8080/JSPWiki/attach/IM_CodeStructure/EBAF%20MVC%E6%9E%B6%E6%A7%8B.jpg)[(info)](http://ws93006ed:8080/JSPWiki/PageInfo.jsp?page=IM_CodeStructure/EBAF%20MVC%E6%9E%B6%E6%A7%8B.jpg)

|  |
| --- |
| http://ws93006ed:8080/JSPWiki/attach/IM_CodeStructure/EBAF%20MVC%E6%9E%B6%E6%A7%8B.jpg |

**moduleTx BeanXMLJSP交易規則VO**

**程式碼結構-module**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=IM_CodeStructure#section-IM_CodeStructure-_E7_A8_8B_E5_BC_8F_E7_A2_BC_E7_B5_90_E6_A7_8BModule)

模組，又稱做 module ，MVC 中對應角色為 model ，主要是滿足主程式所需要的功能

，依職責又可分做一般模組、檢核模組及現在比較少用到的計算模組。

**一般模組檢核模組**

*一般模組，主要處理與資料庫處理有關的功能，舉凡增刪查改等等事項都屬於一般模組的範疇，可說是最廣泛使用的一種模組。*   
  
**一般模組寫作約略可分成以下區塊**

* 1. 套件及匯入區塊
* 2. SPEC 註解區塊
* 3. CLASS 主體
  + 3.1 CLASS 宣告
  + 3.2 屬性宣告
  + 3.3 方法實作

**以下試對各區塊寫作應注意事項進行說明:** 

**1. module-套件及匯入區塊**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=IM_CodeStructure" \l "section-IM_CodeStructure-1.Module_E5_A5_97_E4_BB_B6_E5_8F_8A_E5_8C_AF_E5_85_A5_E5_8D_80_E5_A1_8A)

package com.cathay.xx.s0.module;  
  
import java.util.List;  
import java.util.Map;  
  
import org.apache.commons.lang.StringUtils;  
import org.apache.log4j.Logger;  
  
import com.cathay.common.exception.ModuleException;  
import com.cathay.common.im.util.MessageUtil;  
import com.cathay.common.im.util.VOTool;  
import com.cathay.common.util.db.DBUtil;  
import com.cathay.util.Transaction;  
import com.cathay.xx.vo.DTXXS001;  
import com.igsapp.db.DataSet;

**注意事項** 

* package 放置位置請參照程式碼命名及 package 規則一文-[連結](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=CM_Freshman_Develop_norm#section-CM_Freshman_Develop_norm-_E7_A8_8B_E5_BC_8F_E7_A2_BC_E5_91_BD_E5_90_8D_E5_8F_8Apackage_E8_A6_8F_E5_89_87)http://ws93006ed:8080/JSPWiki/images/out.png
* import 區塊，避免直接使用" **.\*** "進行宣告，必須將匯入的完整 classpath 打出
* 避免留下 import ，但下方程式碼卻未引用的情況，可從 RAD 的黃色驚嘆號符號確認

**2. module-SPEC註解區塊**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=IM_CodeStructure" \l "section-IM_CodeStructure-2.ModuleSPEC_E8_A8_BB_E8_A7_A3_E5_8D_80_E5_A1_8A)

/\*\*  
\* <pre>  
\* 一、  程式功能概要說明：  
\* 程式功能    黃金單身榜維護模組  
\* 程式名稱    XX\_S00100.java  
\* 作業方式    MODULE  
\* 概要說明    黃金單身榜單檔維護    
\* </pre>  
\*  
\* @author 湯姆  
\* @since  2010/01/01  
\*/

**注意事項** 

* 請將 SPEC 首頁的的規格清單貼上，如下圖所示。

|  |
| --- |
| http://ws93006ed:8080/JSPWiki/attach/IM_CodeStructure/module001.png |

* 加上 @author 的註解，並將開發者的姓名打上。(請直接輸入名字，避免SA還要從id去查開發人員是誰)
* 說明內容前後記得標記 <pre></pre>， 維持 javaDoc 的格式
* 詳細註解說明請參考[程式註解](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=CM_Freshman_Develop_norm#section-CM_Freshman_Develop_norm-_E7_A8_8B_E5_BC_8F_E8_A8_BB_E8_A7_A3)http://ws93006ed:8080/JSPWiki/images/out.png

**3.1 module-CLASS主體-CLASS宣告**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=IM_CodeStructure" \l "section-IM_CodeStructure-3.1ModuleCLASS_E4_B8_BB_E9_AB_94CLASS_E5_AE_A3_E5_91_8A)

@SuppressWarnings("unchecked")  
public class XX\_S00100 {  
          
        .  
        .  
        .

**注意事項** 

* 若要抑制編譯器警訊，可在 class 宣告前加上 @SuppressWarnings("unchecked") ，有關此語法可參考此文章-[抑制編譯器警訊 - SuppressWarnings](http://caterpillar.onlyfun.net/Gossip/JavaGossip-V2/AnnotationSuppressWarnings.htm)http://ws93006ed:8080/JSPWiki/images/out.png
* 不需 extends DB\_module ，除非要建立獨立連線的 DataSet 時，才需繼承以取得 DB\_module 的 getDataSet 方法。

**3.2 module-CLASS主體-屬性宣告**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=IM_CodeStructure" \l "section-IM_CodeStructure-3.2ModuleCLASS_E4_B8_BB_E9_AB_94_E5_B1_AC_E6_80_A7_E5_AE_A3_E5_91_8A)

   /\*\* log \*/  
    private static Logger log = Logger.getLogger(XX\_S00100.class);  
          
    /\*\*Log4j是否為Debug模式\*/  
    private boolean isDebug = log.isDebugEnabled();  
  
    private final static String SQL\_001 = "com.cathay.xx.s0.module.XX\_S00100.SQL001";   
          
    private final static String SQL\_002 = "com.cathay.xx.s0.module.XX\_S00100.SQL002";   
          
        .  
        .  
        .

**注意事項** 

* Log 及 isDebug 的用法為固定方式，有關 Log 及 isDebug 用法，請參考 [Log4j用法](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=CM_Freshman_Develop_norm#section-CM_Freshman_Develop_norm-Log4j_E7_94_A8_E6_B3_95)http://ws93006ed:8080/JSPWiki/images/out.png。
* 模組的 SQL 除非特殊情況，否則一律使用 SQL KEY 的方式， SQL KEY 的命名規則為完整 class path 加上 SQL 流水號
  + 以此例完整 class path 為 "com.cathay.xx.s0.module.XX\_S00100" + SQL 流水號 ".SQL001"
    - ex : com.cathay.xx.s0.module.XX\_S00100.SQL001
* SQL 變數名稱同 SQL 流水號，程式宣告一律為 private final static ，以避免被重覆更動
  + ex : private final static String SQL001
* SQL 變數上方需以區塊註解的方式將原 SQL CODE 貼上，方便維護及了解。可將區塊註解前的 \* 拿掉，方便剪下貼上。

**3.1 module-CLASS主體-方法實作**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=IM_CodeStructure" \l "section-IM_CodeStructure-3.1ModuleCLASS_E4_B8_BB_E9_AB_94_E6_96_B9_E6_B3_95_E5_AF_A6_E4_BD_9C)

   /\*\*  
     \* 黃金單身資料檔查詢  
     \* @param id        身分證字號 (非必輸)  
     \* @param sex       性別      (非必輸)  
     \* @return List<Map>  
     \* @throws ModuleException  
     \*/  
    public List<Map> query(String id, String sex) throws ModuleException{  
                  
        DataSet ds = Transaction.getDataSet();  
                  
        if(StringUtils.isNotEmpty(id)){  
                ds.setField("ID", id);  
        }  
        if(StringUtils.isNotEmpty(sex)){  
                ds.setField("SEX", sex);  
        }  
          
        //兩段式寫法，適用於取回之dataSet需要再作處理時  
        //DBUtil.searchAndRetrieve(ds, SQL\_001);  
        //while(ds.next()){  
        //      Map tempMap = VOTool.dataSetToMap(ds);  
                //tempMap處理  
        //}  
          
        //一段式寫法  
        return VOTool.findToMaps(ds, SQL\_001);  
    }  
          
    /\*\*  
     \* 新增  
     \* @param DTXXS001 vo  
     \* @throws ModuleException  
     \*/  
    public void doInsert(DTXXS001 vo) throws ModuleException{  
  
        if(isDebug){  
            log.debug("input vo:");  
            log.debug(vo);  
        }  
                  
        //傳入參數檢核  
        if(StringUtils.isBlank(vo.getID())){  
            MessageUtil.ErrorInput("XX\_S00100\_ERRMSG\_01");      //身分證字號不得為空  
        }  
       
        DataSet ds = Transaction.getDataSet();  
        ds.clear();  
  
        VOTool.insert(vo, ds);  
    }  
           
         .  
         .

**注意事項** 

* 需加上清楚的 javaDoc 註解，請將 SPEC 上的方法說明，及傳入參數的名稱貼上，必要時，請加註此參數是否為必要輸入參數
  + 可用快捷鍵 ctrl+shift+j ，產生區塊註解
* 若傳入參數為必要輸入參數，可利用 MessageUtil 幫我們傳回 ErrorInputException (繼承自 ModuleException ，特別用來表示為輸入參數錯誤的 Exception )
  + 有關 MessageUtil 及輸入參數檢核，請參考 [輸入參數檢核技巧](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=PG_ImInputValidate) 一文。
* 若 SPEC 有要求回傳中文訊息，請務必使用代碼方式取代直接鍵入中文，有關語系代碼的使用，請參考 [多國語系與訊息](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=IM_MultiLanguage) 一文。
* 非必要，請勿在模組內 catch Exception ，再進行包裝的動作
* 非必要，請勿在模組內做 Transaction 的 begin, commit, rollback 的動作
* 如只是初次取得 DataSet，不需執行 clear()；同一 DataSet 需重複使用時，每次使用前都須執行 clear() 清除內容