**Exception規則**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=CM_Freshman_Develop_norm" \l "section-CM_Freshman_Develop_norm-Exception_E8_A6_8F_E5_89_87)

**Exception架構圖**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=CM_Freshman_Develop_norm" \l "section-CM_Freshman_Develop_norm-Exception_E6_9E_B6_E6_A7_8B_E5_9C_96)

|  |
| --- |
| http://ws93006ed:8080/JSPWiki/attach/CM_Freshman_Develop_norm/Exception%E6%B5%81%E7%A8%8B%E5%9C%96.jpg |

* 模組在一般的情況不 catch 任何的 exception，而由 TxBean 來 catch 及 handle
* 模組當有需要時才針對不同 type 的 exception 來處理不同的動作。如:當欲更新的資料不存在時，需新增一筆，就需要將 DataNotFoundException catch 起來處理
* 主程式 catch Exception 之後，使用 ReturnMessage 來將訊息封裝，傳回頁面，再藉由JavaScript來顯示，有關主程式如何將傳送訊息至JSP，請參考 [本文](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=IM_MultiLanguage#section-IM_MultiLanguage-_E8_A8_8A_E6_81_AF_E4_BD_BF_E7_94_A8_E6_96_B9_E5_BC_8F-2)http://ws93006ed:8080/JSPWiki/images/out.png。

**主程式 Exception 處理**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=CM_Freshman_Develop_norm" \l "section-CM_Freshman_Develop_norm-_E4_B8_BB_E7_A8_8B_E5_BC_8FException_E8_99_95_E7_90_86)

以下程式為一般主程式查詢時， Exception 處理的範例。

       try {  
            //----1. 參數取得 ----  
                  
                                                .....  
  
            //----2. 呼叫模組進行處理 ----  
              
                                                .....              
              
            //----3. 組織並設定需回傳至JSP頁面的資料 ----  
              
            ......  
              
            //頁面訊息  
            MessageUtil.setMsg(msg, "MILF002"); //查詢完成  
  
              
        //---- 例外處理 ----  
  
        //<1>  
        } catch (DataNotFoundException dnfe) {  
            log.error(dnfe);  
            MessageUtil.setReturnMessage(msg, ReturnCode.DATA\_NOT\_FOUND, dnfe.getMessage());  
  
        //<2>  
        } catch (ModuleException me) {  
            //<2,1>  
            if (me.getRootException() == null) {  
                log.error(me);  
                MessageUtil.setReturnMessage(msg, ReturnCode.ERROR\_MODULE, me.getMessage());  
            //<2,2>  
            } else {  
                log.error(me.getMessage(), me.getRootException());  
                if (me.getRootException() instanceof OverCountLimitException) {  
                    MessageUtil.setReturnMessage(msg, ReturnCode.ERROR\_MODULE, "XXX0\_0100\_ERRMSG\_OVERCOUNT");  
                } else {  
                    MessageUtil.setReturnMessage(msg, me, req, ReturnCode.ERROR\_MODULE, "ME00632");  
                }  
            }  
        //<3>  
        } catch (Exception e) {  
            log.error("", e);  
            MessageUtil.setReturnMessage(msg, e, req, ReturnCode.ERROR, "查詢失敗");  
        }  
  
        return resp;

**說明**

* 一般若為查詢的功能，catch Exception 的順序 依序應為 DataNotFoundException 、 ModuleException 、 Exception。若不為查詢功能， <1> 的區塊可不需補捉。
* 區塊 <2,1> 的用意在於，模組常會將有意義的自訂錯誤訊息放在一個不包含其他 Exception 的 ModuleException 中往外拋，因此這樣的寫法可以確保 ModuleException 中的自訂訊息可以被取出，放置在 ReturnMessage 中，帶回 JSP 進行顯示。
* 一些早期開發的程式會在 ModuleException 中包了一些底層系統的 Exception ，雖然目前的寫法已經不建議在模組區塊catch Exception 甚至是將exception另外再包裝，但為了避免這樣的問題，區塊 <2,2> 就是用來避免將底層 Exception 的訊息顯示給使用者，造成混淆及系統不穩定的觀感。
* <3> 的用意在於將所有其他的 Exception Catch起來，錯誤訊息使用 log 記錄起來，統一以 "查詢失敗" 傳回至JSP。

**EBAF 常見Exception**[#](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=CM_Freshman_Develop_norm#section-CM_Freshman_Develop_norm-EBAF_E5_B8_B8_E8_A6_8BException)

|  |
| --- |
| http://ws93006ed:8080/JSPWiki/attach/CM_Freshman_Develop_norm/EBAF%E5%B8%B8%E8%A6%8BException.jpg |

有關常見Exception的說明，請參考 [常見Exception](http://ws93006ed:8080/JSPWiki/Wiki.jsp?page=CM_Freshman_Develop_question#section-CM_Freshman_Develop_question-_E5_B8_B8_E8_A6_8BException)http://ws93006ed:8080/JSPWiki/images/out.png 一文。