新光人壽保險股份有限公司

核心系統再造專案-前端頁面轉換

程式開發規範及標準白皮書

|  |  |
| --- | --- |
| 文件類別代號： | 01-10 |
| 版　　　　次： | V2.1 |
| 機　密 等 級： | 密 |
| 文 件 日 期： | 2018/03/26 |

|  |
| --- |
| 新光人壽保險股份有限公司 |
| Shin Kong Life Insurance Co.Ltd. |

文件制／修訂履歷

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 制／修訂  版次 | 制／修訂  日期 | 制／修訂  說明 | 作  者 | 核  准 | 備  註 |
| V0.1 | 2017/09/10 | 初稿 | 趙雪峰 | 劉昊 |  |
| V0.2 | 2017/09/15 | 內容調整 | 徐雲鵬 | 劉昊 |  |
| V1.0 | 2017/10/16 | 交付 | 徐雲鵬 | 劉昊 |  |
| V1.1 | 2017/11/10 | 依V1.0審查意見調整 | 徐雲鵬 | 劉昊 |  |
| V1.2 | 2017/11/14 | 依V1.1審查意見調整 | 徐雲鵬 | 劉昊 |  |
| V2.0 | 2017/11/15 | 審查通過 |  |  |  |
| V2.1 | 2018/03/26 | 更新圖片、更新項目結構圖,建檔圖、新增Webservice開發與測試說明 | 陳智煒/張曉飛 | 羅文琪/鄭理介 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目　　錄

[1 前言 4](#_Toc501977311)

[1.1 文檔說明 4](#_Toc501977312)

[1.2 文檔結構 4](#_Toc501977313)

[2 項目程式結構說明 6](#_Toc501977314)

[2.1 專案結構修改 6](#_Toc501977315)

[2.2 本項目程式結構 7](#_Toc501977316)

[2.3 本地SQLJ的Jar包版本問題 8](#_Toc501977317)

[3 建檔說明 10](#_Toc501977318)

[3.1 建檔路徑 10](#_Toc501977319)

[3.2 命名規則 10](#_Toc501977320)

[3.3 建檔方法 12](#_Toc501977321)

[4 存儲過程的調用 13](#_Toc501977322)

[4.1 Controller中要做的事情 14](#_Toc501977323)

[4.2 UI中要做的事情 16](#_Toc501977324)

[4.3 BLF中要做的事情 17](#_Toc501977325)

[4.4 BL中要做的事情 18](#_Toc501977326)

[5 實體工具類別 19](#_Toc501977327)

[6 存儲過程的工具類別 20](#_Toc501977328)

[SQLj相關檔案、指令產生工具 20](#_Toc501977329)

[7 Webservice 21](#_Toc501977330)

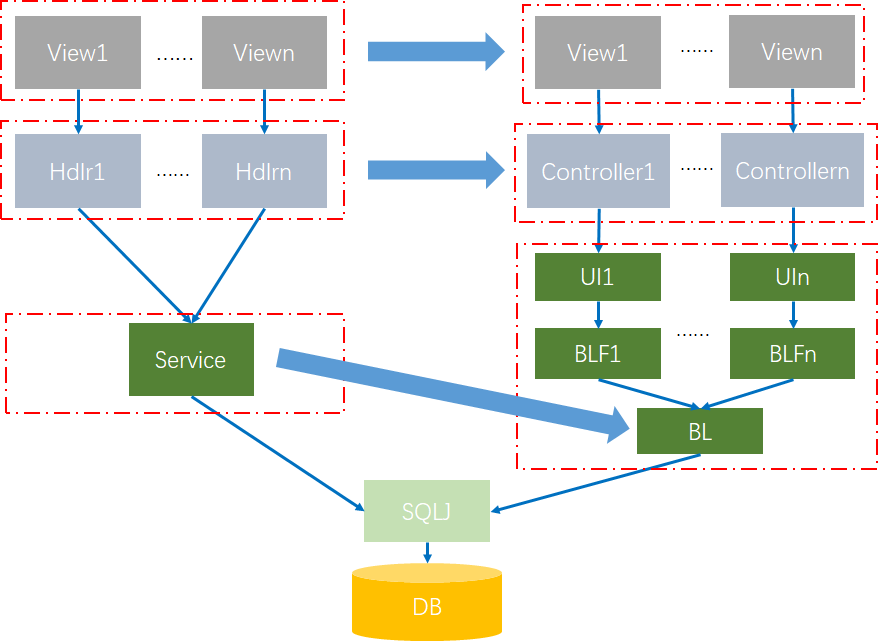
[8 備註 23](#_Toc501977331)

# 前言

* 1. 文檔說明

本文檔編寫的目的是爲了使開發人員，通過本文檔可以順利的進行開發工作。本文檔著重介紹後臺開發，介紹開發人員在使用LIS7架構開發的過程中，可能需要用到的方法。

如下圖所示，本文檔主要說明後臺，即UI、BLF、BL的開發規範。其中，view和controller的部分爲前臺。



## 文檔結構

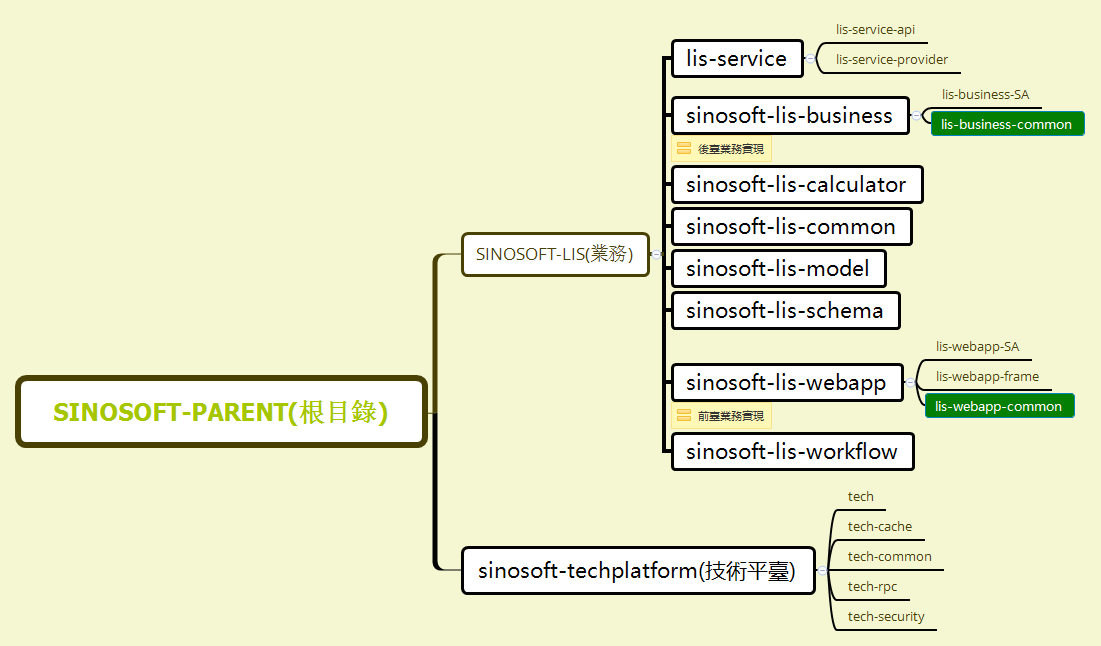
* 項目程式結構說明
* 建檔說明
* UI BLF BL三層結構使用
* 存儲過程的調用
* 實體類別工具
* 存儲過程生成工具
* Webservice配置說明

1. 項目程式結構說明
   1. 專案結構修改

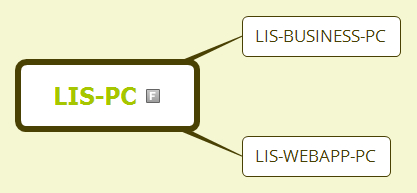
針對本專案，LIS程式結構修改為13線後，項目結構為，每一線有對應的前後臺項目，每一線的前臺都依賴lis-webapp-common且都依賴在lis-webapp-ALL中；每一線的後臺都依賴lis-business-common且都依賴在lis-business-ALL中。lis-webapp-ALL將lis-business-ALL依賴進去，達到前後台一起的目的。SA模塊為系統管理是必備模塊。Webapp-frame為前臺公共的頁面。

下圖中，sinosoft-lis-business是後臺UI、BLF、BL檔放置的地方，sinosoft-lis-webapp是前臺jsp，js，json，controller檔放置的地方。

公共部分:



線別部分:



* 1. 本項目程式結構

本項目程式結構沿用LIS7原有程式結構不變。

1. 建檔說明

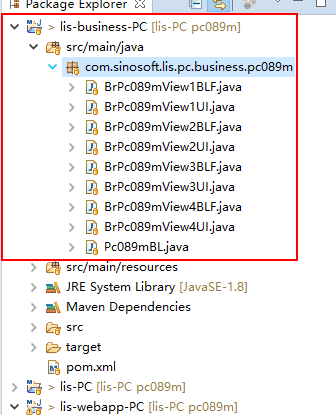
每個頁面的都對應有後臺的邏輯文檔。LIS7中後臺的邏輯文檔有三種，分別是UI、BLF、BL。UI進行數據的轉接，不需要做過多處理，BLF提供訪問BL的邏輯。而真正的邏輯功能都是在BL中進行實現的，在LIS7現行系統中，每種程式功能對應同一個BL，並通過各個view不同的BLF進行指定訪問。

具體邏輯在第4章介紹。

* 1. 建檔路徑

UI、BLF、BL都屬於java檔，需要寫在LIS7的後臺，即lis-business中。如lis-business-PC等package下。其路徑如下圖所示：

其中，com\sinosoft\lis是固定的，pc是某個綫別的模組名稱，business是說明文件在後臺，pc089是當前檔所屬的程式代號。



* 1. 命名規則

對於本專案的明明規則，下面做一下介紹。

在前臺jsp，js，json部分的命名規則爲：

檔案名稱：頁面作業類型+程式代號(第一碼大寫)+View+n(n從1開始)

檔案路徑： lis-webapp-線別\src\main\webapp\WEB-INF\views\模組名稱\程式代號

頁面作業類型又可以分成以下五種

* Br：程式功能的入口頁面
* Ad：新增頁面
* Up：修改頁面
* Qu：查詢細項頁面
* Lov：手電筒查詢頁面

如有一個頁面的jsp爲：

BrPc089mView1.jsp

其中Br是前綴，如還有Ad，Up等。Pc089m是當前頁面所屬的程式代號，View1是頁面的編

號，還有View2,View3等。

相對應的，該js和json檔的名稱分別爲：

BrPc089mView1.js

BrPc089mView1.json

Controller檔的名稱爲：

BrPc089mView1Controller.java

在後臺中，檔案路徑應該為為：

lis-business-線別\src\main\java\com\sinosoft\lis\模組名稱\business\程式代號

其中，lis-business是固定的，表示後臺，APPSUP是線別。

APPSUP之後其他的部分，除了漢字部分需要動態改動，都是固定的。

在後臺對應的UI和BLF爲：

BrPc089mView1UI.java, BrPc089mView1BLF.java。

多個頁面根據程式代號共用同一個BL，即爲：

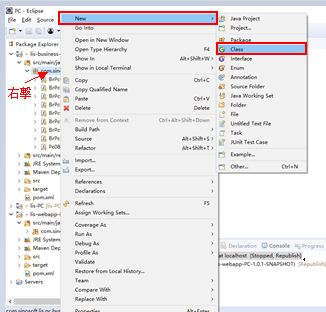
Pc089mBL.java

* 1. 建檔方法

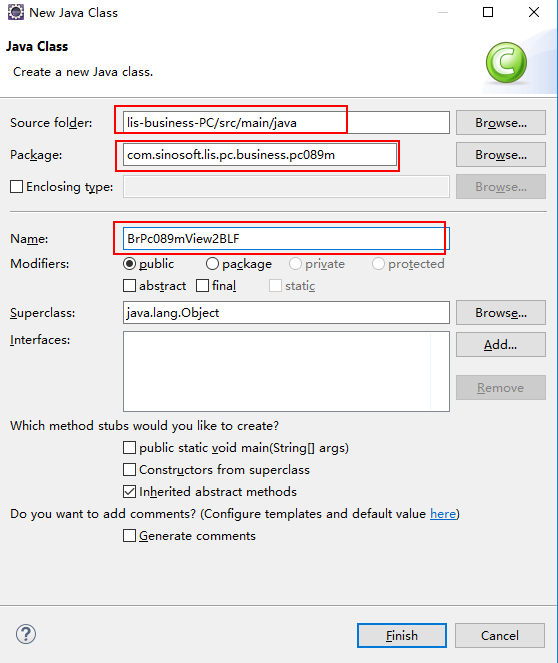
找到需要建立java檔的具體路徑，如，要在lis-business-PC包下建立新的java檔，需要進行如下操作：

1. 要找到需要建立新檔的合適的路徑,如：

lis-business-PC\src\main\java\com\sinosoft\lis\pc\business\pc089m\

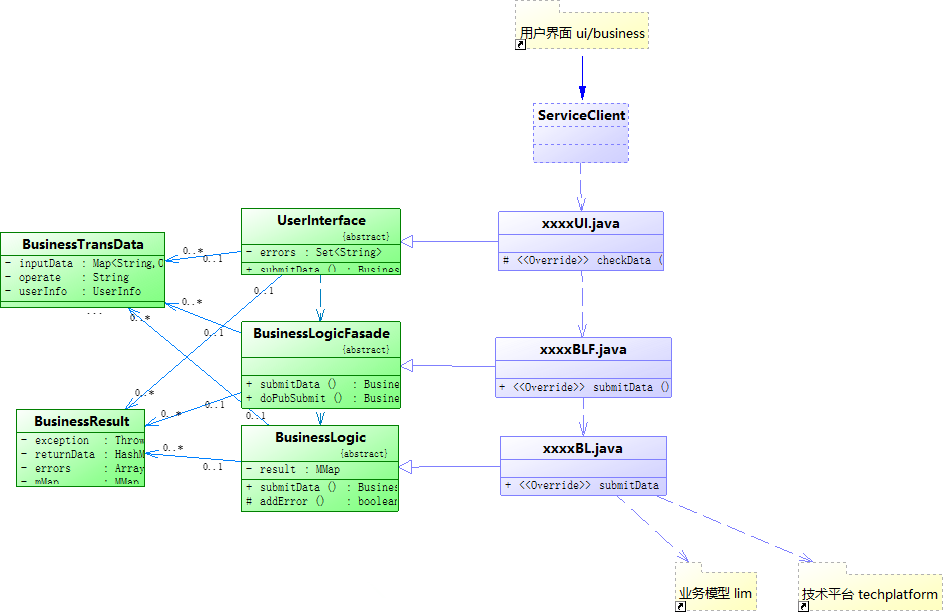


在上邊的路徑上建立BrPc089mView2BLF.java的操作爲：



1. 建立的檔是空白檔，需要自行添加需要繼承的父類別，並添加實現業務邏輯相關的代碼。

# 存儲過程的調用

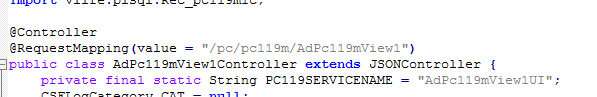
調用存儲過程要使用的UI、BLF、BL的整體結構如下圖所示：

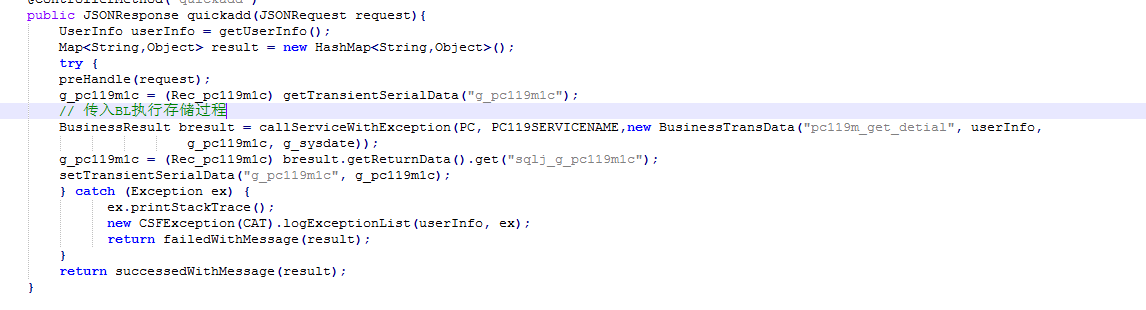
* 1. Controller中要做的事情

在準備完頁面的數據之後，要將頁面上的數據傳遞到controller中，（controller在頁面設計規範及標準白皮書中有詳細介紹），並在controller進行相應的封裝，具體來説，就是有的值需要放入到type實體類別中。

備準好數據之後，執行callServiceWithException的方法，如下：

首先在類別中，需要import當前所屬模塊，聲明當前controller要進行調用的service的名字，如：





然後，對某個方法進行編寫：

callServiceWithException ()方法有3個參數，第一個參數，是所屬模塊，即PC；第二個參數，即後臺UI即service的名稱，如上圖：AdPc006mView\_SERVICENAME；第三個參數是BusinessTransData類別，內部封裝了兩個固定參數，和多個不固定參數：

兩個固定參數分別是要執行的BL中的方法，如上圖中的pc006m\_get\_user和UserInfo資訊。

多個不固定參數是要執行存儲過程的參數，如上圖中的p\_user\_code，這個可以有多個。

上圖中，BusinessResult類別是後臺邏輯執行後，傳回來的返回值。

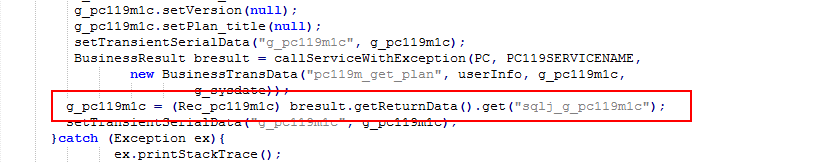
返回值是一個map，

即Map<String,Object> map =result.getReturnData();

如果調用的存儲過程，有返回值，則使用以下方法接取返回值。

String returnData = result.getSingleReturnData();

如果要獲取回傳值,則方法如下：



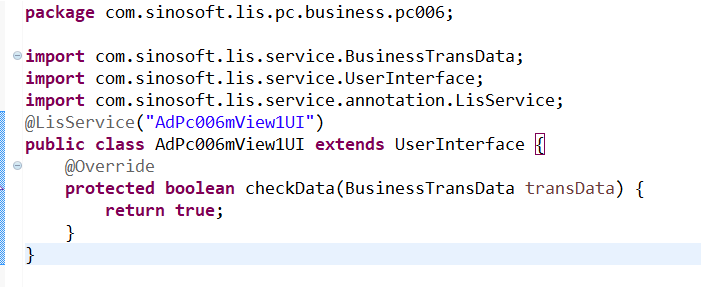
* 1. UI中要做的事情
     1. 功能概述

UI層為前臺模組提供服務（LisService），前臺模組使用CallService訪問後臺將數據傳給UI，UI不進行任何操作，將數據傳遞給BLF，然後接收BLF的資料並返回給前臺模組。

開發人員針對UI不需要做過多的事情，在UI的父類別中已經自動將數據傳遞到BLF，並接收返回的數據，返回到了前臺。

* + 1. 實作

所有的UI都應該繼承UserInterface（com.sinosoft.lis.service），並需要在UI上添加@ LisService注解，注解的名字，要跟當前類別的名字相同。作爲前臺controller進行callService時的唯一識別標記。然後只需要實現父類別介面的checkData方法介面，直接return true即可。一個完整的UI如下。



* 1. BLF中要做的事情
     1. 功能概述

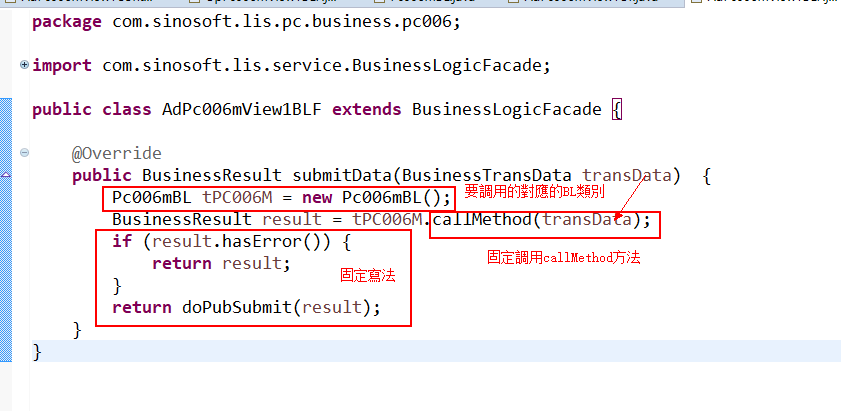
BLF層調用BL完成後續操作，在BL中執行完存儲過程，會得到執行存儲過程之後的結果。然後把執行結果返回到給BLF並在BLF的父類別中返回給UI，然後UI會自動將處理結果返回給前臺controller。

* + 1. 實作

所有BLF都應該繼承BusinessLogicFacade。

在現行系統中，BLF負責進行各個view對相應BL的調用，並執行BL中對存儲過程的呼叫。

如下圖：



上圖為完整的BLF檔案，只需要實現submitData方法即可，除了新聲明一個BL類別之外，其他的代碼都是固定寫法。

* 1. BL中要做的事情
     1. 功能概述

BL主要是對存儲過程進行調用，LIS7採用sqlj的方式對存儲過程進行調用，並會將結果返回到前臺，供後續邏輯使用。

BL的父類別會根據controller中要執行方法的名字標識，自動去BL中找到相應的方法，執行方法中調用的存儲過程，並將存儲過程的返回值和回傳值傳給BLF，進而返回給臺前controller。

* + 1. 實作

所有BL都應該繼承BusinessLogic。路徑在：

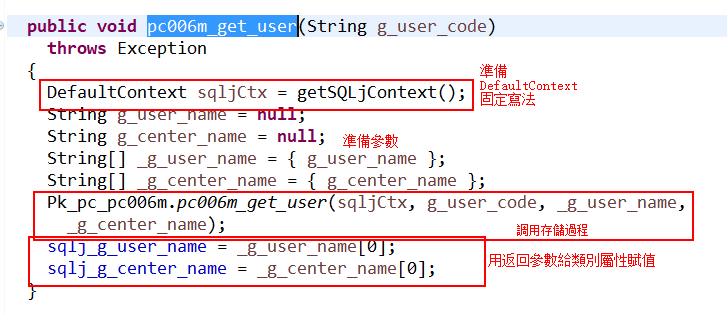
lis-service-provider\src\main\java\com\sinosoft\lis\service\BusinessLogic.java

在BL中首先要把一個DefaultContext進行實例化，即使用：

DefaultContext sqljCtx = getSQLjContext();

然後，調用執行sqlj中的存儲過程，並把執行結果後的返回參數放到全域的public變量中。

即如下圖中的：



1. 實體工具類別

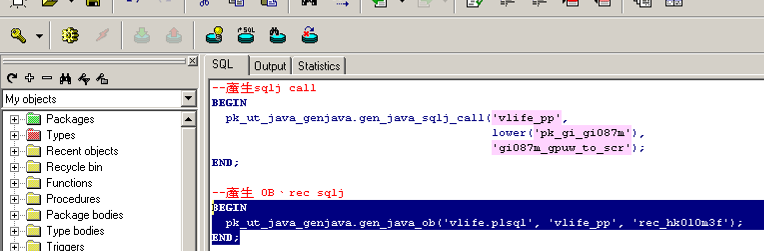
調用資料庫存儲過程時會用到自訂的資料庫類別，LIS7使用利用oracle工具產生的的rec類型來調用存儲過程，實體類別的生成方法見ADF教學文檔第一章，LIS7只用到了其中set值和get值的方法，獲得VO的方法沒有使用。是以jar檔的形式引用到系統中。

1. 存儲過程的工具類別

調用存儲過程使用存儲過程的工具類別，也是已jar檔的形式引入到LIS7的系統中，具體生成sqlj的方法，見ADF教學文檔。引用如下：

SQLj相關檔案、指令產生工具

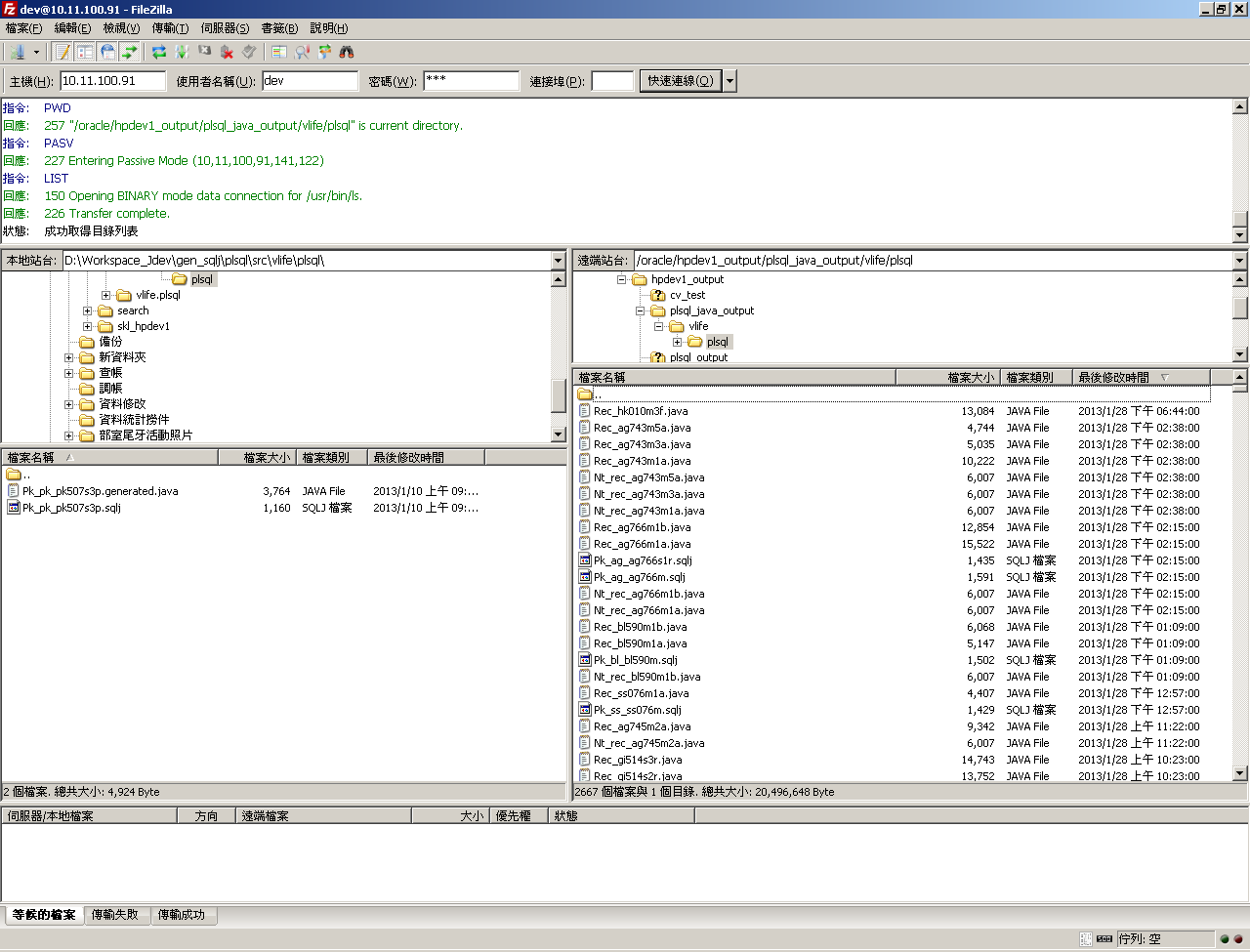
* 工具名稱：gen\_sqlj.sql
* 另外需要安裝的工具：FileZilla FTP
* 使用方法：於開發環境的PL/SQL Developer打開檔案，找出符合需要的指令區段，將名稱改好後選取執行，檔案結果會放到DB後端的plsql\_java\_output資料夾底下。例如：欲產生vlife\_pp底下的rec\_hk010m3f.java檔



**2**

**1**

打開FileZilla連線到開發環境的DB後端(主機：10.11.100.91，帳號、密碼：dev)，並選擇到plsql\_java\_output/vlife/plsql此路徑底下



**3**

**2**

**1**

* 後續請看**SKL- Std-DEV Advance Skill.doc裡的How to 自行產生plsql.jar裡面沒有的class**

1. Webservice
   1. webservice的設定檔的相關設定：



在lis-WS線別下的src\main\java\vlife\v3\ws\中添加源碼即可。

* 1. webservice的開發

Webservice引用uix中的實現方式，所以在開發webservice的時候只需要將uix中的代碼拷貝到指定的線別中就好。然後啟動項目來查看webservice是否發佈成功。

* 1. 判斷是否發佈成功

1. Webservice引用uix中的實現方式，所以在開發webservice的時候只需要將uix中的代碼拷貝到指定的線別中就好。然後啟動項目來查看webservice是否發佈成功。
2. 發佈成功會有下圖所示:發佈路徑/service



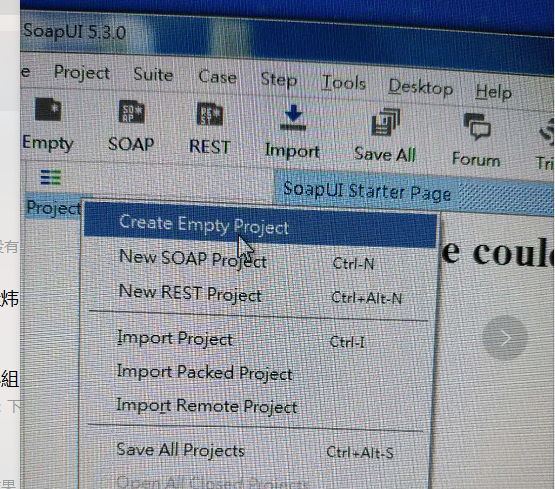
* 1. 測試webservice

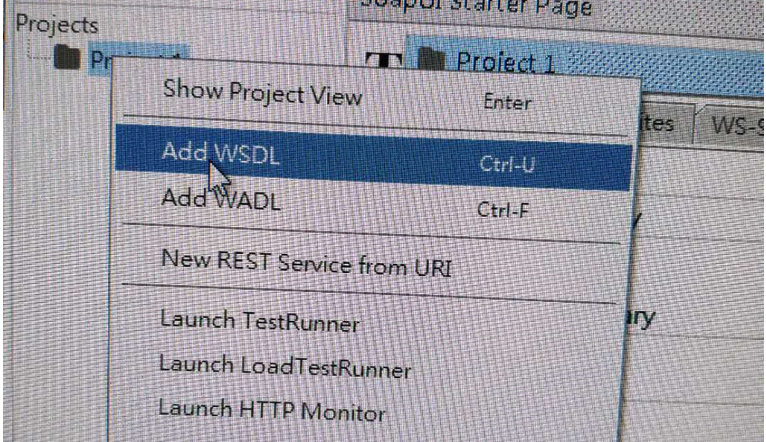
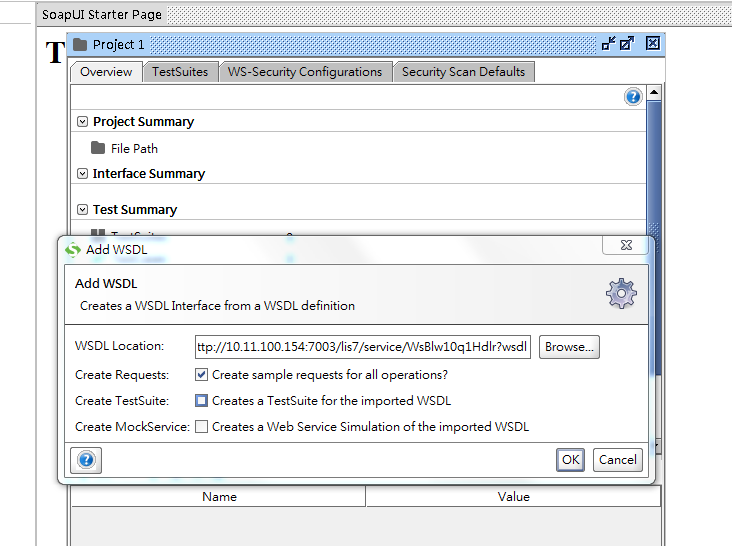
1. 找到自己發佈的Hdrl點擊wsdl在瀏覽器地址欄拷貝wsdl路徑

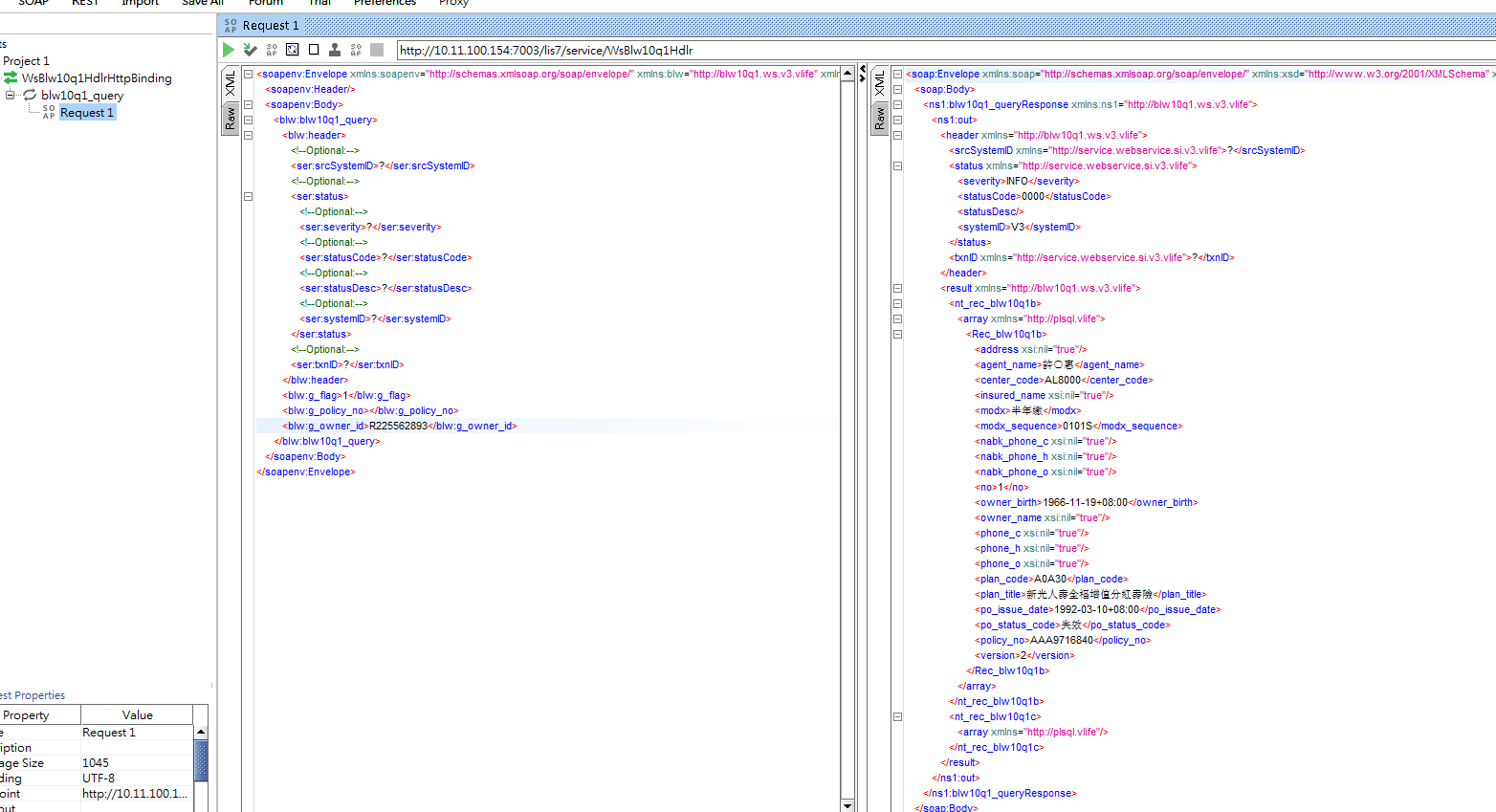


1. 打開測試工具SoapUI

2.1右鍵project創建一個Empty Project



* 1. 右鍵project1 Add Wsdl
  2. 將在瀏覽器地址欄粘貼的路徑拷貝到輸入框中點擊ok。
  3. 找到Hdrl的方法名會發現有個Request1，再點擊會出現xml格式的畫面，在此處輸入對應的參數進行測試。



1. 備註

待ADF標準整理出來後，會根據修改的內容，繼續對本文檔提供更新支援。