**ETL & RESTful Security 應用範例說明**

1. **讓 ETL Tool (Kettle - Spoon) 可呼叫 SSO RESTful 基本步驟：**

* Check-out <http://tcci-svn/svn/src/ETL/trunk/src/ETL/setup/SSOClient>
* copy httpclient-4.2.3.jar, httpcore-4.2.2.jar

to <data-integration-4.4.0>\libext\commons

* copy SSOClient.jar

to <data-integration-4.4.0>\libext

* 修改 kettle.properties, 增加下列資訊

(注意需escape字元, 可用Spoon [Edit] > [Edit the kettle.properties file]修改)

# SSO\_RESTFUL CONFIG

SSO\_TICKET\_URL=http\://192.168.203.81/cas-server/v1/tickets

SSO\_RESTFUL\_USER=devadmin

SSO\_RESTFUL\_PWD=abcd12\#4

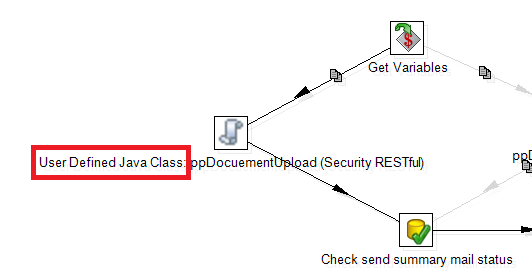
# 自己要呼叫的 RESTFUL SERVICES

SSO\_RESTFUL\_SERVICE\_PP\_DOCUMENT\_BUILD=http\://soa-dev-test.taiwancement.com/procurement/resources/documentService/buildPlantMaterial

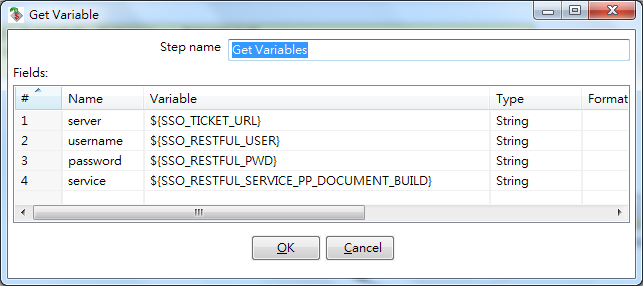
1. **ETL實作範例說明: (參考 \ETL\jobs\procurement\pp\_dump\_data )**
2. 如下圖加入

[User Defined Java Class] 元件：執行SSOClient.jar 呼叫 SSO RESTfull

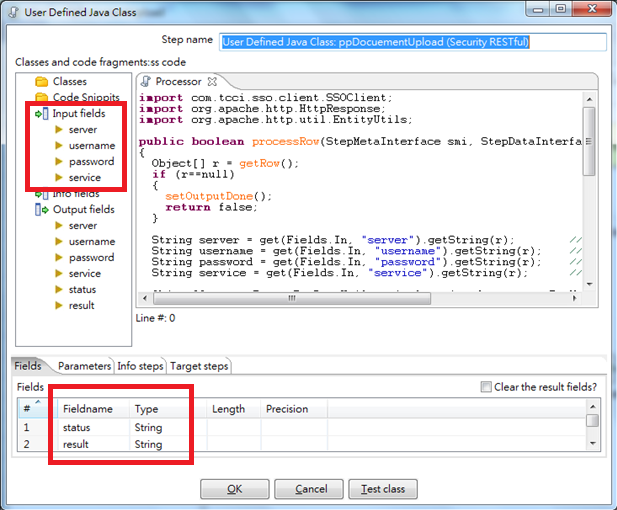
[Get Variables] 元件：傳遞相關參數 至 [User Defined Java Class] 元件用



B: [Get Variables] 內容如下:



C: [User Defined Java Class] 內容如下:



其中 Processor 詳細程式如下:

|  |
| --- |
| **import** com.tcci.sso.client.SSOClient;  **import** org.apache.http.HttpResponse;  **import** org.apache.http.util.EntityUtils;  **public boolean** processRow(StepMetaInterface smi, StepDataInterface sdi) **throws** KettleException  {  Object[] r = getRow();  **if** (r==null)  {  setOutputDone();  **return** false;  }  String server = get(Fields.In, "server").getString(r); //getVariable("server");  String username = get(Fields.In, "username").getString(r); //getVariable("username");  String password = get(Fields.In, "password").getString(r); //getVariable("password");  String service = get(Fields.In, "service").getString(r); //getVariable("service");  Object[] outputRow = RowDataUtil.resizeArray(r, data.outputRowMeta.size());  **int** outputIndex = getInputRowMeta().size();    HttpResponse response = SSOClient.callService(server, username, password, service);  **if** (response != null) {  **try** {  String result = EntityUtils.toString(response.getEntity());  **int** status = response.getStatusLine().getStatusCode();  **if** (status != 200) {  **throw new** KettleException("SSOClient.callService return wrong status: " + status);  }  // LOG.info("response code:" + statusCode);  // LOG.info("Responget(Fields.Out, "status")se (1k): "  // + responseString.substring(0, Math.min(1024, responseString.length())));  **get(Fields.Out, "status").setValue(outputRow, String.valueOf(status));**  **get(Fields.Out, "result").setValue(outputRow, result);**  putRow(data.outputRowMeta, outputRow);  } **catch** (Exception ex) {  **throw new** KettleException(ex.getMessage());  }  } **else** {  **throw new** KettleException("SSOClient.callService return null.");  }  **return** true;  } |

1. **RESTful 實作範說明：(procurement : DocumentREST.java)**
2. web.xml 設定 (基本SSO設定請用參閱 \Guideline\SSO)

*<!-- for REST Security -->*

*<security-constraint>*

*<web-resource-collection>*

*<web-resource-name>documentService</web-resource-name>*

*<url-pattern>/resources/documentService/\*</url-pattern>*

*</web-resource-collection>*

*<auth-constraint>*

*<role-name>Administrators</role-name>*

*<role-name>AP-Role</role-name>*

*</auth-constraint>*

*</security-constraint>*

1. 若需 Method 細部設定:

@Singleton

@Path("documentService")

public class DocumentREST {

….

@GET

@Path("buildPlantMaterial")

@Produces("text/plain")

@RolesAllowed({"Administrators"}) // for REST Security

@TransactionAttribute(TransactionAttributeType.NOT\_SUPPORTED)

public String buildPlantMaterial(@QueryParam("factory\_code") String factory\_code){

….

}

….

}