commonFunctions class:

Mostly has backend connectivity and functions to execute queries.

import following packages:

**import** com.sforce.soap.enterprise.sobject.CRD\_\_c;

**import** com.sforce.soap.partner.Connector;

**import** com.sforce.soap.partner.PartnerConnection;

**import** com.sforce.soap.partner.QueryResult;

**import** com.sforce.soap.partner.sobject.SObject;

**import** com.sforce.ws.ConnectionException;

**import** com.sforce.ws.ConnectorConfig;

Connectivity code:

**public** **static** PartnerConnection connectDeveloperConsole() {

**try** {

ConnectorConfig config = **new** ConnectorConfig();

config.setUsername(*txt\_UserName*);

config.setPassword(*txt\_Password*);

*connection* = Connector.*newConnection*(config);

System.***out***.println("Success");

} **catch** (ConnectionException e) {

System.out.println(“Fail”);

}

**return** *connection*;

}

Sample Query:

**public** **static** **void** getExternalCase\_Id() **throws** ConnectionException {

**try** {

*strExtrnalCase*="";

String query = "SELECT Id FROM Attachment where Parentid = '"+*case\_\_c*+"' AND Name like '%xml%'";

QueryResult result\_CRDRecordId = *connection*.query(query);

SObject record\_CRD[]=result\_CRDRecordId.getRecords();

*strExtrnalCase*=(String) record\_CRD[0].getField("Id");

}**catch**(Exception e) {

System.***out***.println("Record Failed");

}

}

ExcelUtils

Mostly has functions to deal with fetching values from excel and writing to excel.

xmlFunctions

Has functions to fetch values from xml.

//Getting the XML file

**public** **static** **void** setXmlFile(String Path){

**try** {

File fXmlFile = **new** File(Path);

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.*newInstance*();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

*doc* = dBuilder.parse(fXmlFile);

*doc*.getDocumentElement().normalize();

} **catch** (Exception e) {

e.printStackTrace();

}

}

//Getting the Node List

**public** **static** NodeList gettingNodeList(String tagName)

{

NodeList TList = *doc*.getElementsByTagName(tagName);

**return** TList;

}

//Getting the XML data

**public** **static** String [][] getXMLData(NodeList nList, String[] tags){

String [][] Elements = **new** String [nList.getLength()][tags.length];

**try** {

//test =doc.getDocumentElement().getNodeName();

**for** (**int** temp = 0; temp < nList.getLength(); temp++) {

Node nNode = nList.item(temp);

**int** temp1 = temp +1;

//System.out.println("\nCurrent Element :" + nNode.getNodeName() + " " + temp1);

**if** (nNode.getNodeType() == Node.***ELEMENT\_NODE***) {

Element eElement = (Element) nNode;

**for**(**int** i=0;i<tags.length;i++)

{

Elements[temp][i]= eElement.getElementsByTagName(tags[i]).item(0).getTextContent();

// System.out.println(tags[i]+ " :" + eElement.getElementsByTagName(tags[i]).item(0).getTextContent());

}

}

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return**(Elements);

}

}