Döcu Content

IBM Notes Domino, DB2

Oracle JDeveloper

Grab Code Samples

++?? 2015.11.08.11.24.PM

Been out of it for a while... Taking a quick look to see what we can find on Google as well

http://www.ibm.com/developerworks/data/library/techarticle/dm-0610kokkat/

Options to consider, will keep looking though

http://stackoverflow.com/questions/2310139/how-to-read-xml-response-from-a-url-in-java

here is code actually I think we can use to grab XML via URL and split it into parts for DB2 submission

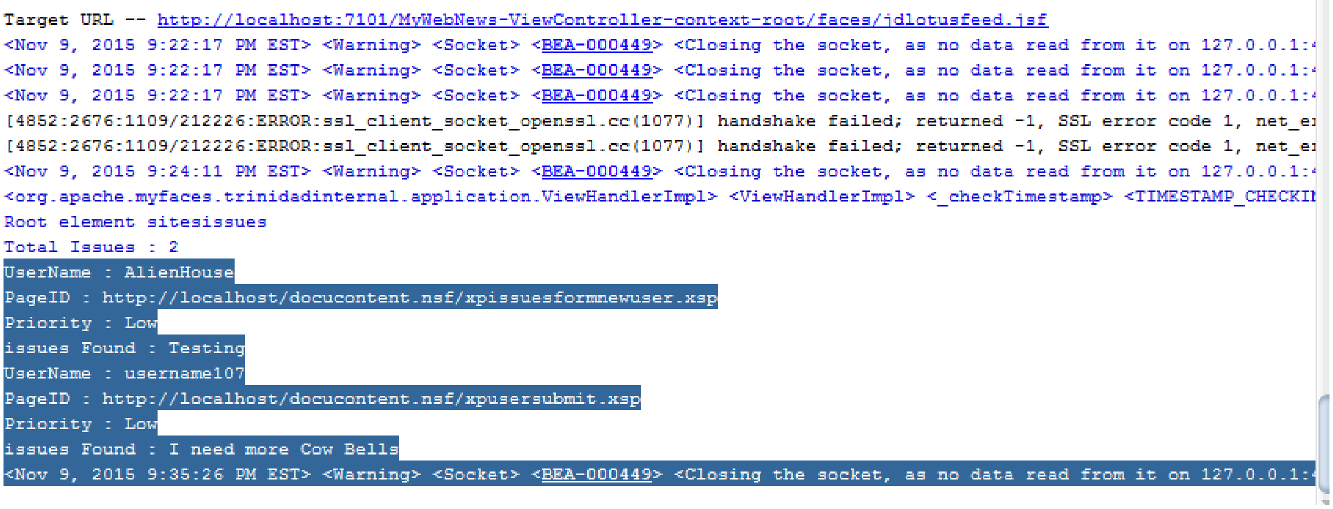
```
package com.dokoll.solutions.inc.search;

import java.net.*;
import java.io.*;


public class CreateXmlOnDir {
      public void doGetXMLFromURL() {
            try {


                  //TO DO: Send ip and other info to back-end if page firing
                  //this button code

                  //Get XML file from URL.
                  URL xmlURL = new URL("http", "localhost", 80,
                              "/docucontent.nsf/dashboardchartdata.xml");
                  // Set up Connection to URL, grab file
                  URLConnection conn3ctXML = xmlURL.openConnection();
                  // connect to URL. grab file
                  conn3ctXML.connect();

                  // Build URL file into new XML file
                  // ...
                  PrintWriter printerWriter = new PrintWriter(new FileWriter(
                              "c:\\temp\\XML_DATA\\documentbycategoryfromurl.xml"))
;
```

```
            // Read URL data into new XML file
            BufferedReader bufferRead = new BufferedReader(
                        new InputStreamReader(conn3ctXML.getInputStream()));
            // ...
            String LineRead = bufferRead.readLine();

            // ...
            while (LineRead != null) {
                  printerWriter.println(LineRead);
                  LineRead = bufferRead.readLine();

            }

            printerWriter.close();

      } catch (Exception e) {
            System.out.println("Error: " + e);

      }

}



public void doGetXMLForChart() {
      try {

            //TO DO: Send ip and other info to back-end if page firing
            //this button code

    //Get XML file from URL.
    URL xmlURL = new URL("http", "localhost", 80,
                  "/docucontent.nsf/dashboardchartd
atajasper.xml");
            // Set up Connection to URL, grab file
            URLConnection conn3ctXML =
xmlURL.openConnection();
            // connect to URL. grab file
            conn3ctXML.connect();

            // Build URL file into new XML file
            // ...
            PrintWriter printerWriter = new PrintWriter(new FileWriter(
                        "c:\\temp\\XML_DATA\\documentforcharts.xml"));
    // Read URL data into new XML file
    BufferedReader bufferRead = new BufferedReader(
                  new
InputStreamReader(conn3ctXML.getInputStream()));
            // ...
            String LineRead = bufferRead.readLine();
            // ...
            while (LineRead != null) {
                printerWriter.println(LineRead);
                LineRead = bufferRead.readLine();

            }

            printerWriter.close();

        } catch (Exception e) {
            System.out.println("Error: " + e);

        }

    }

}
```

I am actually going to use only part the code, highlighted above... of course combine with what we picked up from the CSV file read, split into DB2 column options...

Let's see what we've got!

Below is the WebLogic console results...



```
Target URL -- http://localhost:7101/MyWebNews-ViewController-context-root/faces/jdlotusfeed.jsf
<Nov 9, 2015 9:22:17 PM EST> <Warning> <Socket> <BEA-000449> <Closing the socket, as no data read from it on 127.0.0.1:
<Nov 9, 2015 9:22:17 PM EST> <Warning> <Socket> <BEA-000449> <Closing the socket, as no data read from it on 127.0.0.1:
<Nov 9, 2015 9:22:17 PM EST> <Warning> <Socket> <BEA-000449> <Closing the socket, as no data read from it on 127.0.0.1:
[4852:2676:1109/212226:ERROR:ssl_client_socket_openssl.cc(1077)] handshake failed; returned -1, SSL error code 1, net_e
[4852:2676:1109/212226:ERROR:ssl_client_socket_openssl.cc(1077)] handshake failed; returned -1, SSL error code 1, net_e
<Nov 9, 2015 9:24:11 PM EST> <Warning> <Socket> <BEA-000449> <Closing the socket, as no data read from it on 127.0.0.1:
<org.apache.myfaces.trinidadinternal.application.ViewHandlerImpl> <ViewHandlerImpl> <_checkTimestamp> <TIMESTAMP_CHECKI
Root element sitesissues
Total Issues : 2
UserName : AlienHouse
PageID : http://localhost/docucontent.nsf/xpissuesformnewuser.xsp
Priority : Low
issues Found : Testing
UserName : username107
PageID : http://localhost/docucontent.nsf/xpusersubmit.xsp
Priority : Low
issues Found : I need more Cow Bells
<Nov 9, 2015 9:35:26 PM EST> <Warning> <Socket> <BEA-000449> <Closing the socket, as no data read from it on 127.0.0.1:
```

Records are coming in from the XML file, let's try to read it into strings that can be used for submission to DB2ADMIN.NOTES

Combined code from existing project

Areas of interest are highlighted for your convenience...

```
package com.dokoll.view.backing;

import com.dokoll.solutions.inc.oracle_src.DBConnector;
```

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

import java.io.PrintWriter;

import java.net.MalformedURLException;
import java.net.URL;
import java.net.*;

import java.io.*;

import javax.faces.context.FacesContext;

import javax.servlet.http.HttpServletResponse;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;


import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;
import org.xml.sax.SAXParseException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import oracle.adf.view.rich.component.rich.RichDocument;
import oracle.adf.view.rich.component.rich.RichForm;
import oracle.adf.view.rich.component.rich.data.RichTable;
import oracle.adf.view.rich.component.rich.input.RichInputNumberSpinbox;
import oracle.adf.view.rich.component.rich.input.RichInputText;
import oracle.adf.view.rich.component.rich.input.RichTextEditor;
import oracle.adf.view.rich.component.rich.nav.RichButton;
import oracle.adf.view.rich.component.rich.output.RichOutputText;

//...
//Start of App
public class Jdlotusfeed {
    private RichForm f1;
    private RichDocument d1;
    private RichButton b1;
    private String ShowData;
    private String UserID;
    private String PageID;
    private String Priority;
    private String Issues;
    private RichInputText it1;
    private RichInputNumberSpinbox ins1;
    private RichInputNumberSpinbox ins2;
    private RichInputText it2;
    private RichInputText it3;
    private RichInputText it4;
    private RichTable t1;
    private RichOutputText ot2;
    private RichOutputText ot3;
    private RichOutputText ot4;
    private RichOutputText ot5;
    private RichTextEditor rte1;


    public String getShowData() {
        return ShowData;
    }

    public void setShowData(String showData) {
        ;
        this.ShowData = showData;

    }


    public String getUserID() {
        return UserID;
    }

    public void setUserID(String userID) {
        ;
        this.UserID = userID;

    }


    public String getPageID() {
        return PageID;
    }

    public void setPageID(String pageID) {
        ;
        this.PageID = pageID;

    }


    public String getPriority() {
        return Priority;
    }

    public void setPriority(String priority) {
        ;
        this.Priority = priority;

    }
```

```
    public String getIssues() {
        return Issues;
    }

    public void setIssues(String issues) {
        ;
        this.Issues = issues;

    }


    public void setF1(RichForm f1) {
        this.f1 = f1;
    }

    public RichForm getF1() {
        return f1;
    }

    public void setD1(RichDocument d1) {
        this.d1 = d1;
    }

    public RichDocument getD1() {
        return d1;
    }

    public void setB1(RichButton b1) {
        this.b1 = b1;
    }

    public RichButton getB1() {
        return b1;
    }


    //Button code
    @SuppressWarnings("oracle.jdeveloper.java.nested-assignment")
    public String b1_action() throws MalformedURLException, IOException {
        // Add event code here...

            //...

            //TESTING PURPOSES IN CONSOLE

            //...

        try {

            DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docBuilderFactory.newDocumentBuilder();

            Document doc =
docBuilder.parse("http://localhost/docucontent.nsf
/javaagentdataforwebnewshttp.xml");
```

```
            // ...
            doc.getDocumentElement().normalize();
            System.out.println("Root element " +
doc.getDocumentElement().getNodeName());

            NodeList listOfIssues = doc.getElementsByTagName("person");
            int totalSheets = listOfIssues.getLength();
            System.out.println("Total Issues : " + totalSheets);

            for (int s = 0; s < listOfIssues.getLength(); s++) {

                Node currPersonNode = listOfIssues.item(s);
                if (currPersonNode.getNodeType() == Node.ELEMENT_NODE) {

                    Element currPersonElement = (Element) currPersonNode;

                    // ...
                    // firstname value...
                    NodeList userNameList =
currPersonElement.getElementsByTagName("username");
                    Element userNameElement = (Element) userNameList.item(0);

                    NodeList textUserNameList = userNameElement.getChildNodes();
                    System.out.println("UserName : " + ((Node)
textUserNameList.item(0)).getNodeValue().trim());

                    // ...
                    // lastname value...
                    NodeList pageIDList =
currPersonElement.getElementsByTagName("pageid");
                    Element pageIDNameElement = (Element) pageIDList.item(0);

                    NodeList textPageIDList = pageIDNameElement.getChildNodes();
                    System.out.println("PageID : " + ((Node)
textPageIDList.item(0)).getNodeValue().trim());

                    // ...
                    // age value...
                    NodeList priorityList =
currPersonElement.getElementsByTagName("priority");
                    Element ageElement = (Element) priorityList.item(0);

                    NodeList priorityInfoList = ageElement.getChildNodes();
                    System.out.println("Priority : " + ((Node)
priorityInfoList.item(0)).getNodeValue().trim());

                    // ...
                    // height value...
                    NodeList issuesFoundList =
currPersonElement.getElementsByTagName("usernotes");
                    Element heightElement = (Element) issuesFoundList.item(0);

                    NodeList textHeightList = heightElement.getChildNodes();
                    System.out.println("issues Found : " + ((Node)
textHeightList.item(0)).getNodeValue().trim());

                    //ShowData= ((Node)
textUserNameList.item(0)).getNodeValue().trim();
```

```
//...


            //...

            //TESTING PURPOSES IN DB2ADMIN.NOTES DOMINODB

            //...
//TO DO: Split programs into two so that button runs for the specific  process,
results in console or split into Strings for DB2


                    //Get XML file from URL.
                    URL xmlURL = new URL("http", "localhost", 80,
"docucontent.nsf/javaagentdataforwebnewshttp.xml");
                            // Set up Connection to URL, grab file
                            URLConnection conn3ctXML =
xmlURL.openConnection();
                            // connect to URL. grab file
                            conn3ctXML.connect();


                    // Read URL data into new XML file
                    BufferedReader bufferRead = new BufferedReader(new
InputStreamReader(conn3ctXML.getInputStream()));
                    //...
                    //establish a Connection
                    Connection connection = DBConnector.getConnection();
                    //TO DO: Attempt using Prepared Statement here...
                    Statement statement = connection.createStatement();
                    String line;
                    while ((line = bufferRead.readLine()) != null) {
            String[] values = line.split(","); //your seperator

            //Convert String to right type. Integer, double, date etc.
            statement.executeUpdate("INSERT INTO
DB2ADMIN.NOTES_DOCS_RPT VALUES('" + values[0] + "','" +
                                                          values[1] + "','" +
values[2] + "','" + values[3] + "')");
                        //statement.executeUpdate("INSERT INTO
DB2ADMIN.NOTES_DOCS_RPT
VALUES('"+values[0]+"','"+values[1]+"','"+values[2]+"','"+values[3]+"','"+values[4]
+"','"+values[5]+"','"+values[6]+"','"+values[7]+"','"+values[8]+"','"+values[9]+"'
,'"+values[10]+"')");
                        //Use a PeparedStatemant, it´s easier and safer

                        System.out.println("Insert these in to DB2: " + values);
                    }
                    bufferRead.close();
                    // clean up
                    statement.close();
                    connection.close();
```

```
                }

            }

        } catch (SAXParseException err) {
            System.out.println("Error" + ", errLine " + err.getLineNumber() + ",
uri " + err.getSystemId());
            System.out.println(" " + err.getMessage());

        } catch (SAXException e) {
            Exception x = e.getException();
            ((x == null) ? e : x).printStackTrace();

        } catch (Throwable t) {
            t.printStackTrace();
        }


        return null;


    }


    public void setIt1(RichInputText it1) {
        this.it1 = it1;
    }

    public RichInputText getIt1() {
        return it1;
    }

    public void setIns1(RichInputNumberSpinbox ins1) {
        this.ins1 = ins1;
    }

    public RichInputNumberSpinbox getIns1() {
        return ins1;
    }

    public void setIns2(RichInputNumberSpinbox ins2) {
        this.ins2 = ins2;
    }

    public RichInputNumberSpinbox getIns2() {
        return ins2;
    }

    public void setIt2(RichInputText it2) {
        this.it2 = it2;
    }

    public RichInputText getIt2() {
        return it2;
    }

    public void setIt3(RichInputText it3) {
        this.it3 = it3;
    }

    public RichInputText getIt3() {
        return it3;
    }

    public void setIt4(RichInputText it4) {
        this.it4 = it4;
    }

    public RichInputText getIt4() {
        return it4;
    }

    public void setT1(RichTable t1) {
        this.t1 = t1;
    }

    public RichTable getT1() {
        return t1;
    }

    public void setOt2(RichOutputText ot2) {
        this.ot2 = ot2;
    }

    public RichOutputText getOt2() {
        return ot2;
    }

    public void setOt3(RichOutputText ot3) {
        this.ot3 = ot3;
    }

    public RichOutputText getOt3() {
        return ot3;
    }

    public void setOt4(RichOutputText ot4) {
        this.ot4 = ot4;
    }

    public RichOutputText getOt4() {
        return ot4;
    }

    public void setOt5(RichOutputText ot5) {
        this.ot5 = ot5;
    }

    public RichOutputText getOt5() {
        return ot5;
    }

    public void setRte1(RichTextEditor rte1) {
        this.rte1 = rte1;
    }
    public RichTextEditor getRte1() {
        return rte1;
    }


}
```