

Döcu Content

IBM Notes Domino, DB2

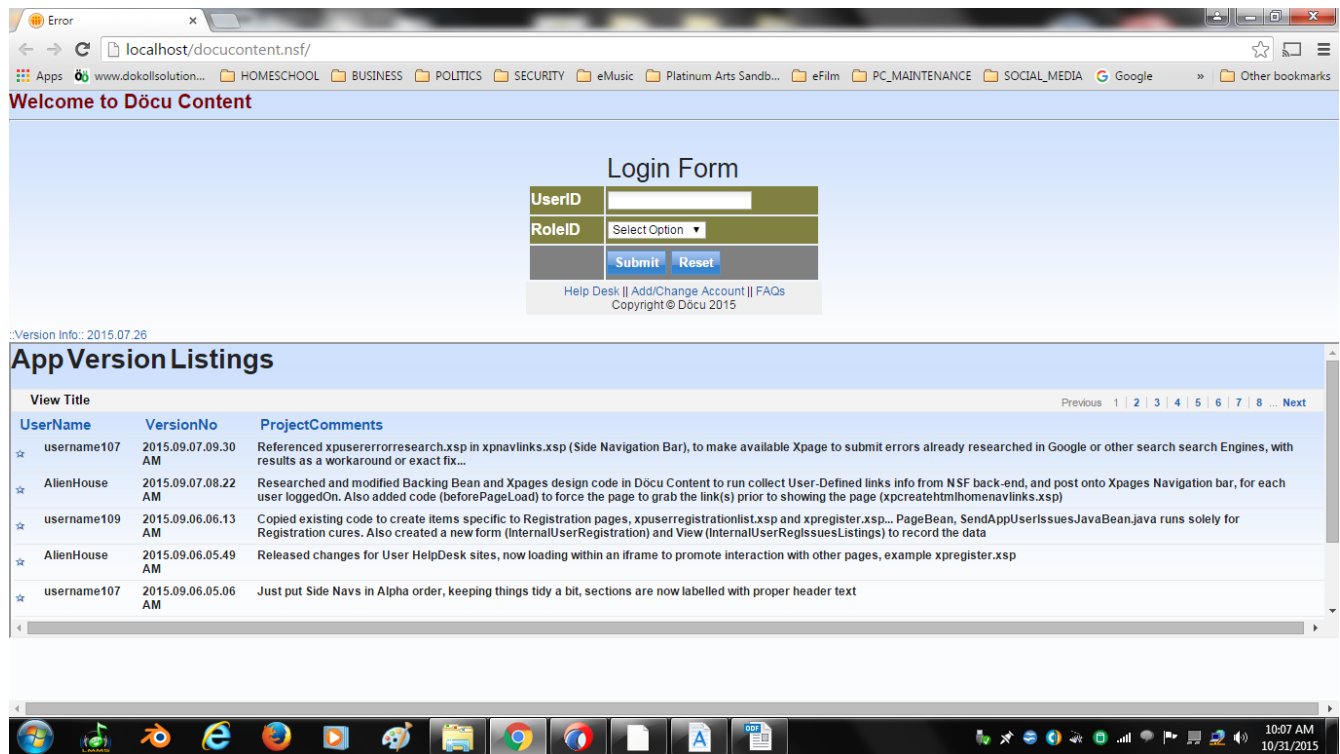
Oracle JDeveloper, WebLogic

Research/Create App Classes

+++++++>>>
2015.10.31.9.56.AM

Getting IBM Lotus Notes Domino XML, TXT, CSV data into Oracle JDeveloper JSF button control via URL, submission to DB2...

Before I go searching if there's available code elsewhere... will look through the existing Apps and see if something similar at least was worked on.



I am going to use this sample here, find a way to get data via XML or CSV URL within JDeveloper and submit as DB2 columns

```
ReadCSVFileIntoDB2Bean.java - IBM Lotus Domino Designer
File Edit Source Refactor Create Design Navigate Tools Search Project Run Window Help

Home x ReadCSVFileIntoDB2Bean.java x
//...
//Button code
public static void doGetCSVForDB2Data()
{
    //...
    //Entering try catch
    try
    {
        Connection connection = DB2Connector.getConnection();

        // create a Statement object to execute the query with
        //2012.04.21.6.02.PM
        //TO DO: Attempt using Prepared Statement here...
        Statement statement = connection.createStatement();

        // Select* from UserNewLineOutboundInformation.csv
        BufferedReader br = new BufferedReader(new FileReader("C:\\temp\\CSV_DATA\\BULK_CSV_COM_DATA_SPLIT_2012.csv"));
        String line;
        while ( (line=br.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]+"','"+values[4]+"')");
            //Use a PreparedStatement, it's easier and safer

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

Here is the full code in screenshot above

to do what I need, I will have to split the XML or CSV file obtained from IBM Lotus Notes Application and submit to DB2ADMIN.NOTES

```
/**
 * Created: 2012.04.21.5.56.PM
 * Build Results into DB2/NSF with CSV
 */
package com.dokoll.solutions.inc.cvs.trials;

/**
 * @author Dököll Solutions, Inc.
 * @version 2012.04.21.5.56.PM
 */
import java.io.BufferedReader;
import java.io.FileReader;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.faces.context.FacesContext;

import com.dokoll.solutions.inc.development.Utils.DB2Connector;

import lotus.domino.local.Database;
import lotus.domino.local.Document;

//...
//Start of Program
public class ReadCSVFileIntoDB2Bean
```

```

{
    //...
    //Button code
    public static void doGetCSVForDB2Data ()
    {
        //...
        //Entering try catch
        try
        {
            Connection connection = DB2Connector.getConnection();

            // create a Statement object to execute the query with
            //2012.04.21.6.02.PM
            //TO DO: Attempt using Prepared Statement here...
            Statement statement = connection.createStatement();

            // Select* from UserNewLineOutboundInformation.csv
            BufferedReader br = new BufferedReader(new
FileReader("C:\\temp\\CSV_DATA\\BULK_CSV_COM_DATA_SPLIT_2012.csv"));
            String line;
            while ( (line=br.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]+"', '"+values[4]
+'', '"+values[5]+"', '"+values[6]+"', '"+values[7]+"', '"+values[8]+"', '"+values[9]+"',
+'', '"+values[10]+"')");
                //Use a PeparadStatement, it's easier and safer

                System.out.println("Insert these in to DB2: " + values);
            }
            br.close();
            // clean up
            statement.close();
            connection.close();
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}

```

Other code being considered to salvage for parts...

```

//Button code
public static void doGetCSVForLogNSFData ()
{
    //...

```

```

        //Entering try catch
try
{
    // load the driver into memory
    Class.forName("org.relique.jdbc.csv.CsvDriver");

    // create a connection. The first command line parameter is assumed to
    // be the directory in which the .csv files are held
    Connection conn =
DriverManager.getConnection("jdbc:relique:csv:C:\\temp\\CSV_DATA\\");

    // create a Statement object to execute the query with
    //2012.04.21.6.02.PM
    //TO DO: Attempt using Prepared Statement here...
    Statement stmt = conn.createStatement();

    //String [] SearchTerms = {" "};
    // Select * from BULK_CSV_COM_DATA_SPLIT.csv
    ResultSet results = stmt.executeQuery("SELECT * FROM
BULK_CSV_COM_DATA_SPLIT");
    int docount = 0;
    // dump out the results
    while (results.next())
    {

        //TO DO: Write a separate JavaBean to build setters
        //that can used in Xpages to show entries newly made...
        String strFileDate = results.getString("FileDate");
        String strFileTime = results.getString("FileTime");
        String strErrorAmPm = results.getString("ErrorAmPm");
        String strErrorRequest = results.getString("ErrorRequest");
        String strErrorOrigin = results.getString("ErrorOrigin");
        String strErrorDescrip = results.getString("ErrorDescrip");
        String strErrorLang = results.getString("ErrorLang");

        //connect to NSF back-end to load data in
        Database database= (Database) FacesContext.getCurrentInstance()
        .getApplication().getVariableResolver()
        .resolveVariable(FacesContext.getCurrentInstance(), "database");
        System.out.println("Connected to Database..." + database);

        //declare and initialize doc variable
        //begin creating your document
        Document submitDocument = (Document) database.createDocument();
        //load to form in your back-end
        submitDocument.replaceItemValue("Form", "LogReport");
        submitDocument.replaceItemValue("FileDate", strFileDate);
        submitDocument.replaceItemValue("FileTime", strFileTime);
        submitDocument.replaceItemValue("ErrorAmPm", strErrorAmPm);
        submitDocument.replaceItemValue("ErrorRequest", strErrorRequest);
        submitDocument.replaceItemValue("ErrorOrigin", strErrorOrigin);
        submitDocument.replaceItemValue("ErrorDescrip", strErrorDescrip);
        submitDocument.replaceItemValue("ErrorLang", strErrorLang);

        //Load to Console for debugging purposes
        //TO DO: Comment this out, as you might expect, all app(log).nsf
documents load back
        //to console with this exercise (save your space as much as possible)

```

```

System.out.println("strFileDate " + " strErrorDescrip");
System.out.println(strFileDate + " " + strErrorDescrip);
//save this doc, otherwise it is not recorded
submitDocument.save();
//increment counter
    docount += 1;
}

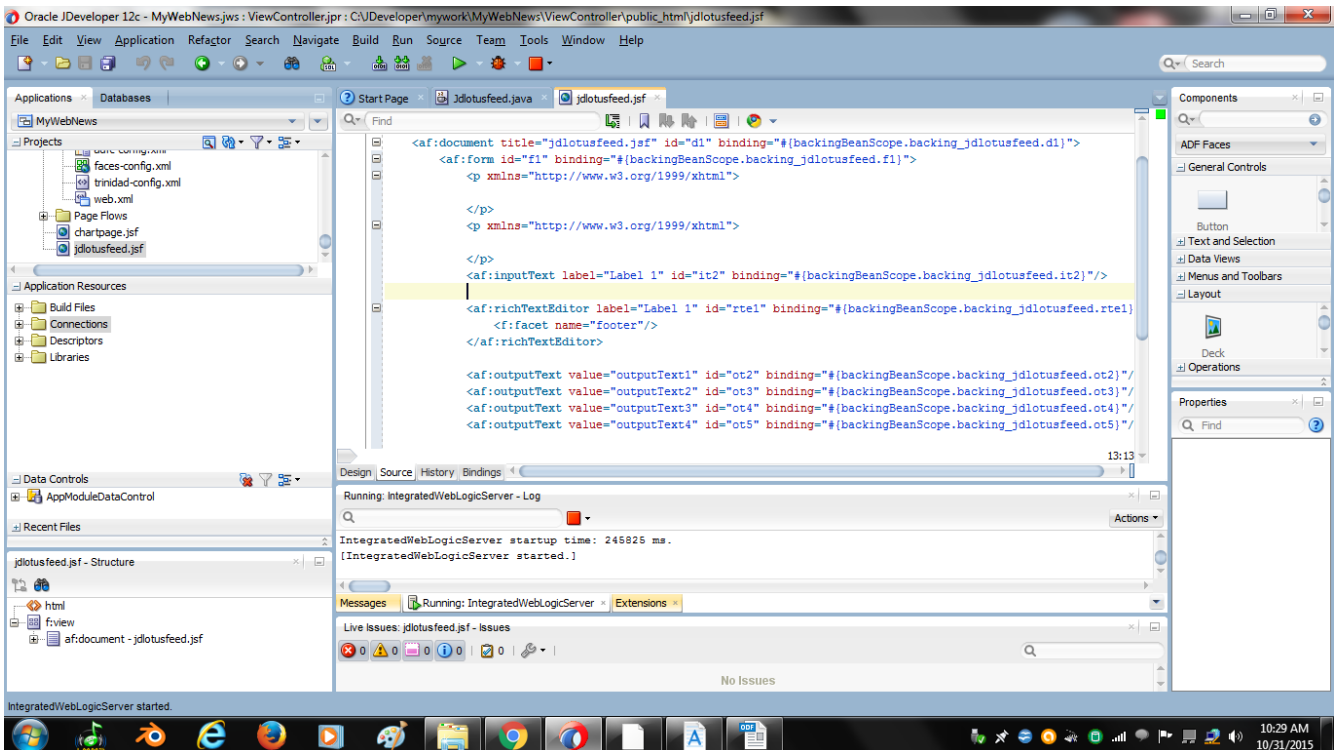
// clean up
results.close();
stmt.close();
conn.close();
}
catch (Exception e)
{
    e.printStackTrace();
}
} //end of program...
}

```

in this case, rather than going to Lotus Notes Domino database via Xpages form, it will be submitted to DB2 via JSF page in JDeveloper...

Brief Recap:

We have a running JSF code started from the following App in JDeveloper



Peek @ WebLogic Server Info, while we wait...

```
IntegratedWebLogicServer startup time: 245825 ms.
[IntegratedWebLogicServer started.]
[Running application MyWebNews on IntegratedWebLogicServer...]
[10:30:54 AM] Web Module ViewControllerWebApp.war recognized in project
ViewController.jpr
[10:30:58 AM] ---- Deployment started. ----
[10:30:58 AM] Target platform is (Weblogic 12.x).
[10:31:40 AM] Retrieving existing application information
[10:31:40 AM] Running dependency analysis...
[10:31:40 AM] Deploying 2 profiles...
[10:31:47 AM] Wrote Web Application Module to
C:\Users\Administrator\AppData\Roaming\JDeveloper\system12.1.3.0.41.140521.1008\o.j
2ee\drs\MyWebNews\ViewControllerWebApp.war
[10:31:51 AM] Wrote Enterprise Application Module to
C:\Users\Administrator\AppData\Roaming\JDeveloper\system12.1.3.0.41.140521.1008\o.j
2ee\drs\MyWebNews
[10:31:51 AM] Deploying 1 data source(s) to the server...
```

Previously submitted Information:

This is the JSF code that will be used to submit back into DB2ADMIN.NOTES database, and to do this we are going to use snippets from the code above from Lotus Notes Domino Xpages App

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

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

import java.io.PrintWriter;

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

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 bl_action() throws MalformedURLException, IOException {
    // Add event code here...

    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();

    }

}

} 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;
}

}

```

Lastly, to make it interesting, let's get the values into JSF
inputText controls prior to submitting to DB2ADMIN.NOTES

Currently this data is read in JDeveloper, let's split it and get started

