

Backing Bean/CSV

Query CSV file from Java

Xpages Button

System Requirements:

Download Domino Designer 8.5.2 Environment (DDE)

<http://www.ibm.com/developerworks/downloads/ls/dominodesigner/>

Introduction:

Load XML file from C Drive using Java, create a CSV file.

Disclaimer:

Information contained in the following is presented as is. This tutorial assumes you have basic programming knowledge. All tutorials are based on an Eclipse/Eclipse-based software. Should you need to familiarize yourself with a certain Eclipse environment, prior to continuing this tutorial, please stop now and see our Tutorials page...

Write Java class to create CSV file

At this point we assume Domino Designer 8.5.2 is downloaded/installed and two Java clas has been created. You are going to use an XML file with a Java to create a CSV file with NSF data. Copy and paste below code to your environment, areas of interest have been highlighted for your convenience.

CreateCSVBean.java

[CODE]

```
/**
 *
 * @AppName: Docu.nsf
 * Program: CReadCSVFileVBean.java
 * Created from Copy: 2012.04.21.5.56.PM
 * Build Results into Console with CSV
 */

package com.dokoll.solutions.inc.cvs.trials;

/**
 * @author Dököll Solutions, Inc.
```

```

* @version 2012.04.21.5.56.PM
*
*/
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

//...
//Start of Program
public class ReadCSVFileBean
{

    //...
    //Button code
    public static void doGetCSVConsoleData ()
    {
        //...
        //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();

            // Select* from UserNewLineOutboundInformation.csv
            ResultSet results = stmt.executeQuery("SELECT * FROM
UserNewLineOutboundInformation");

            // dump out the results
            while (results.next())
            {

                //TO DO: Write a separate JavaBean to build setters for variables
                //that can used in Xpages...
                String SiteName = results.getString("SiteName");
                String SiteNumber = results.getString("SiteNumber");
                //Load to Console
                System.out.println("SiteName " + " SiteNumber");
                System.out.println(SiteName + " " + SiteNumber);
            }

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

```

```
}  
} //end of program...
```

```
}
```

Build Xpage file

Full code added below, jump ahead if necessary, areas of interest have been highlighted for your convenience.

xpload2csv.xsp

```
<?xml version="1.0" encoding="UTF-8"?>  
<xp:view xmlns:xp="http://www.ibm.com/xsp/core">  
  
        <xp:link escape="true" id="link4"  
            style="text-align:center;background-position:center  
center;width:88.0px">  
            <xp:eventHandler event="onclick"  
                submit="true"  
refreshMode="complete" immediate="false"  
                save="true"  
id="eventHandler2">  
                <xp:image  
style="width:270.0px;height:170.0px;hspace:10.0px;align:center" id="image3"  
url="http://localhost/FileBin.nsf/DepartmentInventory.jpg"/>  
                <xp:this.action><!  
[CDATA[#{javascript:CreateCSVBean.getKeywords()}]></xp:thi  
s.action>  
                </xp:eventHandler></xp:link>  
</xp:view>
```

Conclusion:

You can now build CSV files from XML, data that can be used by your application in the making of charts and other types of reports. This could be a good way to use SQL to perform queries against this file and attempt other tasks

TIP: You will need to reference [ReadCSVFile.java](#) in your faces-config.xml file. Also, this item can be used with a current XML creation from URL sample loaded prior.

References:

<http://csvjdbc.sourceforge.net/>

Questions, comments, please post a brief message on our [Contact](#) form on the main site. Thank you for coming...