JDeveloper
Read Lotus Notes Data via URL
Part 3

Introduction:

Read App Data from Lotus Notes Database into Java Server Faces Page on JDeveloper, running on Weblogic Server

Going forward, we will be looking at the data through TXT data in URL from Lotus Notes Database, feed the rest of the JDeveloper application. We could as well simply throw the records right into columns on DB2, but here we wanted to demonstrate the interaction between Apps via URL using Java, Xpages, and JSF...

HTTPDDECSVDataJDevJavaAgent | JavaAgent.java

the goal is to put it out there as a CSV or TXT file...

```
/**
 * Created from copy: 2015.07.04.12.39.PM
 * HTTPDDECSVDataJDevJavaAgent
 * CSV data for Xpages to be read by Oracle JDeveloper Apps
 */

//load imports
import lotus.domino.*;
import lotus.domino.local.Database;
import java.io.PrintWriter;
/**
 * @author Dököll Solutions, Inc.
 * @version 2015.07.04.12.39.PM
 *
 */
// begin class
public class JavaAgent extends AgentBase {
        // open method, this actually runs the whole App
        public void NotesMain() {
              // let's add a try catch here, to grab errors near the end
              try {
                    // open our session...
                    Session session = getSession();
                    // load info to console for debugging purposes
                    System.out
                                .println("HTTPDDECSVDataJDevJavaAgent System User,
We've got a session..."
                                            + session);
                    // load agentContext
                    AgentContext agentContext = session.getAgentContext();
                    // ...
                    System.out.println("got HTTPDDECSVDataJDevJavaAgent
agentContext..."
                                + agentContext);
                    // find database based on session found
                    Database database = (Database) agentContext.getCurrentDatabase();
                    // Find view in question according to current database
                    View view = database.getView("IssuesListings");
                    System.out.println("View Obtained..." + view);
                    // declare document variables
                    Document currDoc;
                    Document tempDoc;
                    // grab first doc
                    currDoc = view.getFirstDocument();
                    // PrintWriter...
                    PrintWriter pw = getAgentOutput();
                    System.out
                                .println("URL:
http://localhost/docucontent.nsf/javaagentfeedcsvexternalsites.txt");

              // Content type set at Txt
              // TODO: Figure out a better version to load a CSV file to browser
                    // currently, the csv downloads to local directory
                    pw.println("Content-type:text/txt");
                    while (currDoc != null) {
                          // write records from view into a Text file and show it in
the
                          // browser
                          pw.println(currDoc.getItemValueString("UserID") + " "
                                      + currDoc.getItemValueString("PageID") + " "
                                      + currDoc.getItemValueString("priority") + " "
                                      + currDoc.getItemValueString("issues"));

                          // Get next document
                          tempDoc = view.getNextDocument(currDoc);
                          // recycle currDoc
                          currDoc.recycle();
                          // set currDoc to tempDoc
                          currDoc = tempDoc;
                    }

              } catch (Exception e) {
                    e.printStackTrace();
              }
        }
}
```

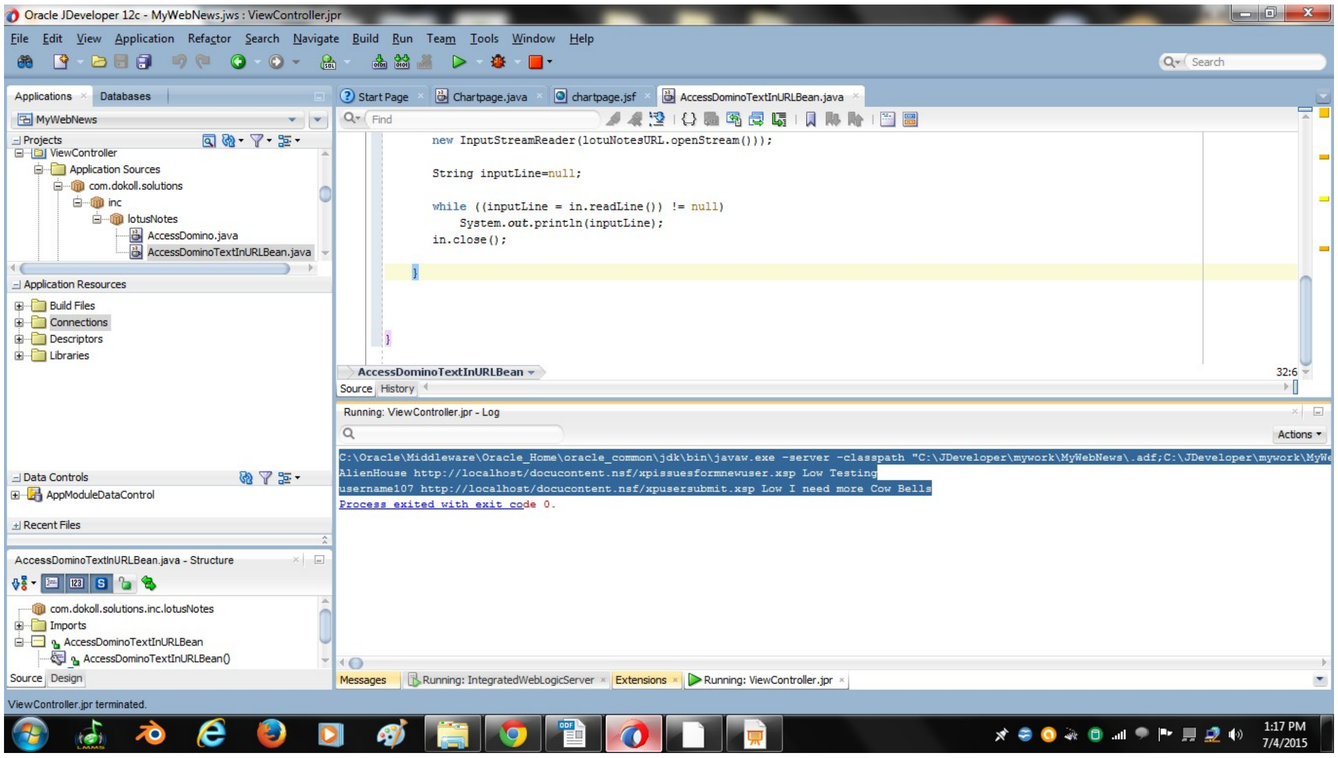
From JDeveloper Side of things

Courtesy: http://stackoverflow.com/questions/6259339/how-to-read-a-text-file-directly-from-internet-using-java

I can do this in JDeveloper to read the file over

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

 public class UrlTextfile {
public static void main(String[] args) throws Exception {

    URL oracle = new URL("http://yoursite.com/yourfile.txt");
    BufferedReader in = new BufferedReader(
    new InputStreamReader(oracle.openStream()));

    String inputLine;
    while ((inputLine = in.readLine()) != null)
        System.out.println(inputLine);
    in.close();
}
 }
```

and there you have it, data is in JDeveloper

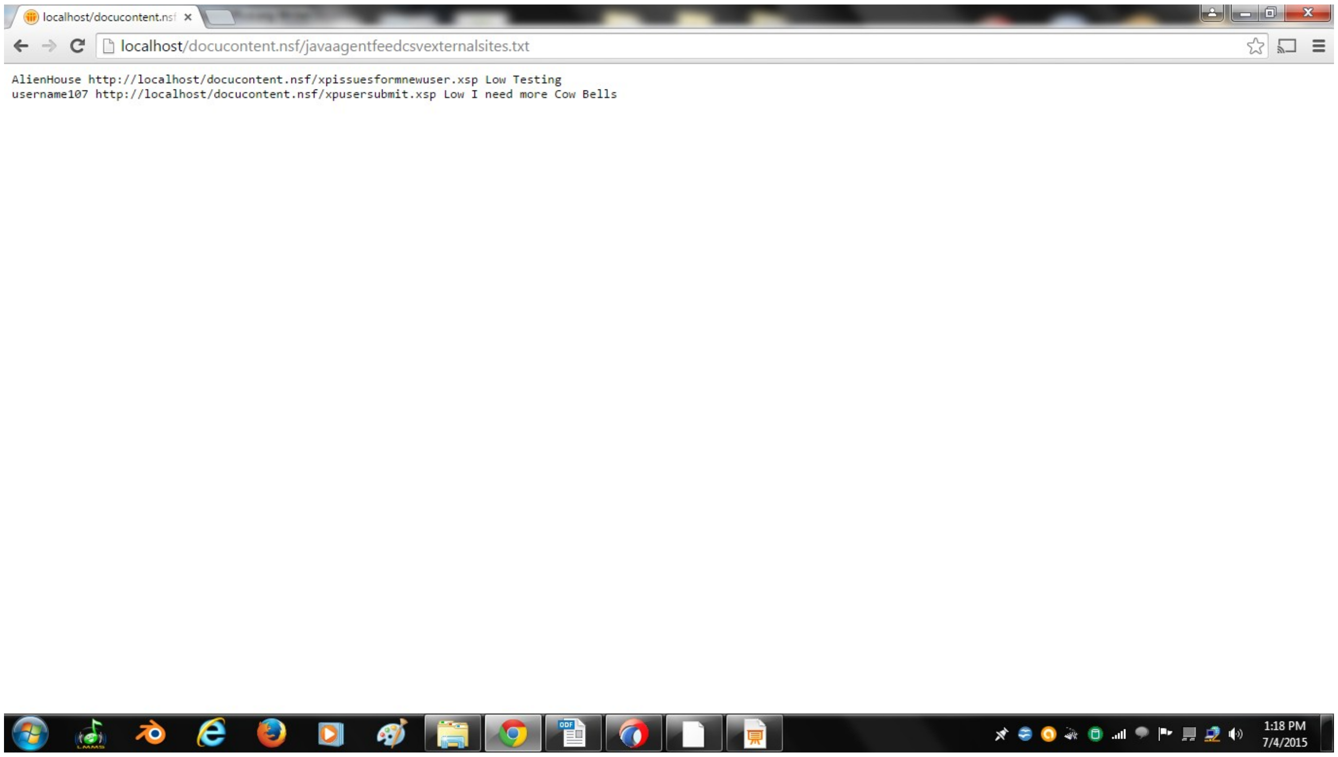


```
        new InputStreamReader(lotuNotesURL.openStream());

        String inputLine=null;

        while ((inputLine = in.readLine()) != null)
            System.out.println(inputLine);
        in.close();

    }


}
```

Running: ViewController.jpr - Log

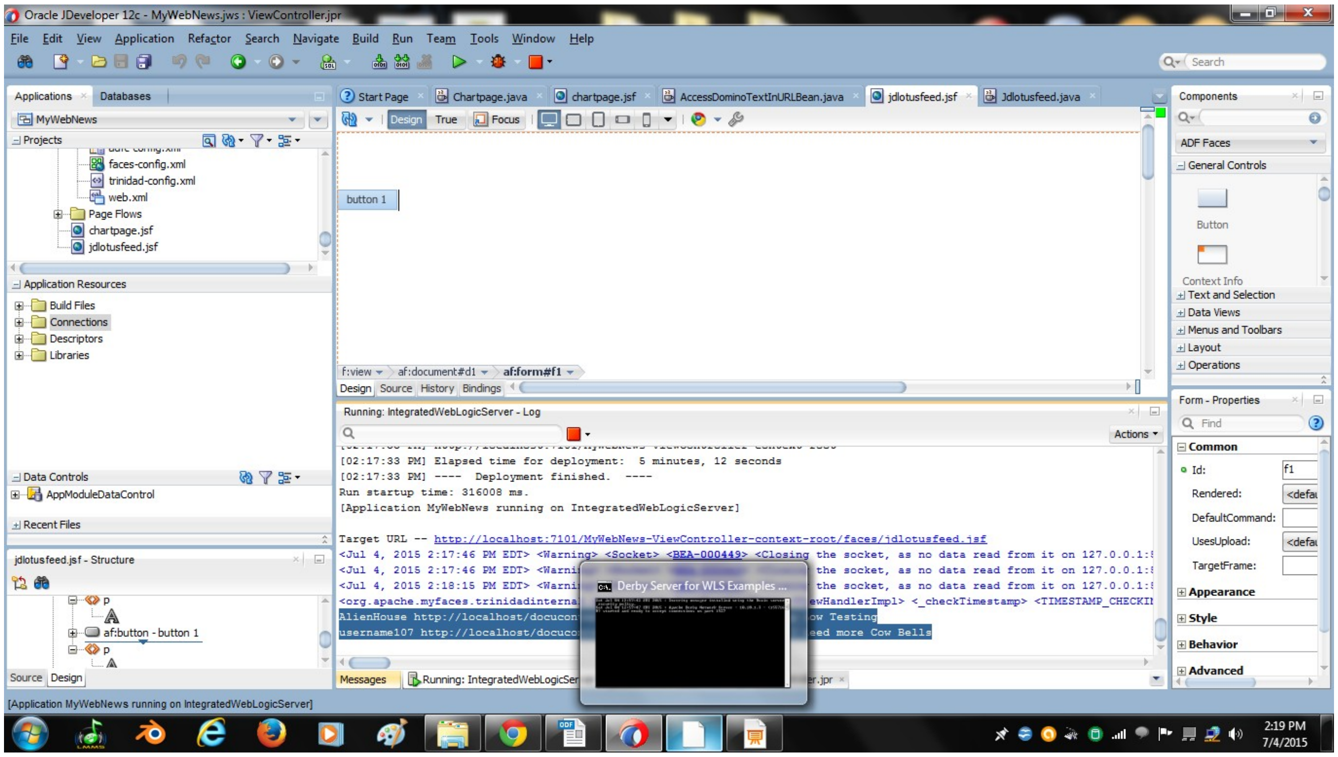
```
C:\Oracle\Middleware\Oracle_Home\oracle_common\jdk\bin\javaw.exe -server -classpath "C:\JDeveloper\mywork\MyWebNews\.adf;C:\JDeveloper\mywork\MyWe
AlienHouse http://localhost/docucontent.nsf/xpissuesformnewuser.xsp Low Testing
username107 http://localhost/docucontent.nsf/xpusersubmit.xsp Low I need more Cow Bells
Process exited with exit code 0.
```

Notes Database URL, ran from Domino Designer (DDE)



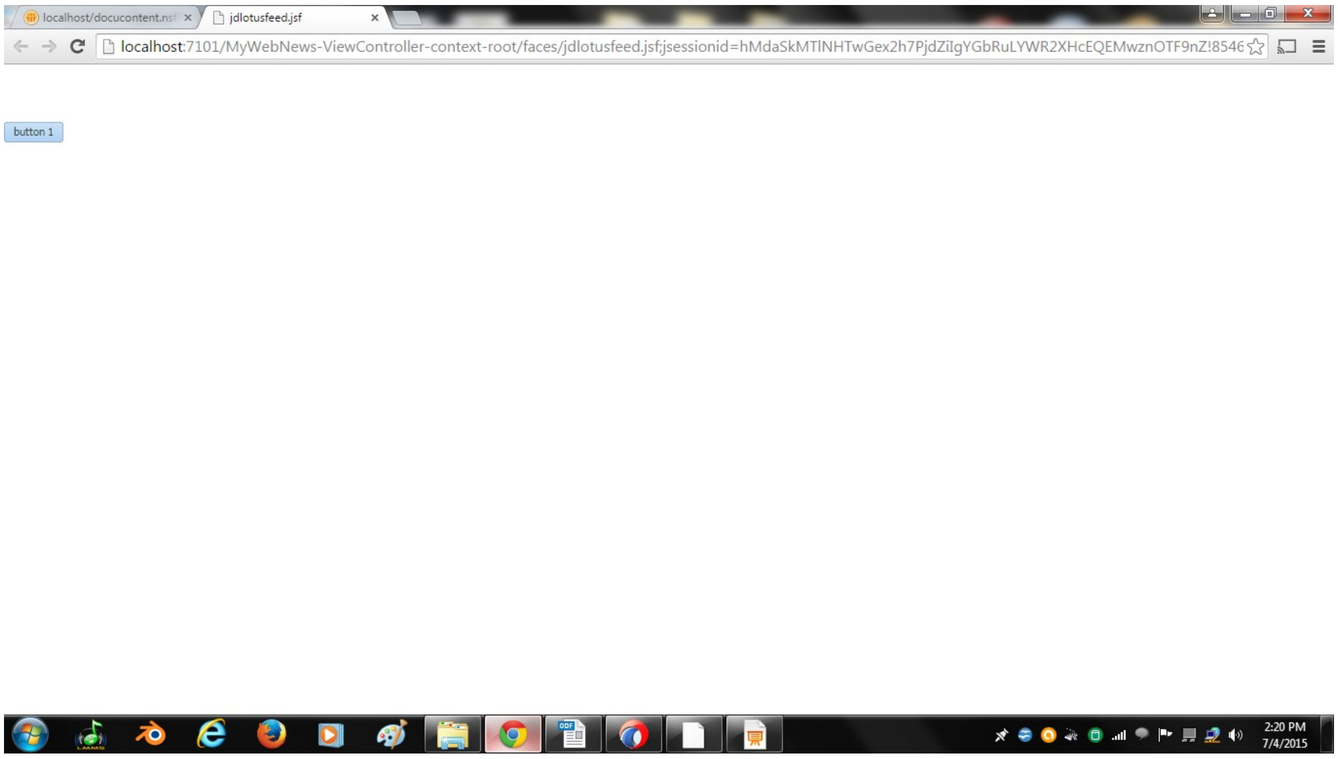
```
AlienHouse http://localhost/docucontent.nsf/xpissuesformnewuser.xsp Low Testing
username107 http://localhost/docucontent.nsf/xpusersubmit.xsp Low I need more Cow Bells
```

created button on JSF page launched via JDeveloper



```
[02:17:33 PM] Elapsed time for deployment:  5 minutes, 12 seconds
[02:17:33 PM] ----  Deployment finished.  ----
Run startup time: 316008 ms.
[Application MyWebNews running on IntegratedWebLogicServer]

Target URL -- http://localhost:7101/MyWebNews-ViewController-context-root/faces/jdlotusfeed.jsf
<Jul 4, 2015 2:17:46 PM EDT> <Warning> <Socket> <BEA-000449> <Closing the socket, as no data read from it on 127.0.0.1:
<Jul 4, 2015 2:17:46 PM EDT> <Warni                                    the socket, as no data read from it on 127.0.0.1:
<Jul 4, 2015 2:18:15 PM EDT> <Warni                                    the socket, as no data read from it on 127.0.0.1:
<org.apache.myfaces.trinidadinterna                                 ewHandlerImpl> <_checkTimestamp> <TIMESTAMP_CHECKI
AlienHouse http://localhost/docucon                                 ow Testing
username107 http://localhost/docuco                                 eed more Cow Bells
```

Java Server Pages button


localhost/docucontent.nsf
jdlotusfeed.jsf
localhost:7101/MyWebNews-ViewController-context-root/faces/jdlotusfeed.jsf;jsessionid=hMdaSkMTlNHTwGex2h7PjdZiIgYGbRuLYWR2XHcEQEMwznOTF9nZ!8546
button 1
2:20 PM
7/4/2015

here is the JSF code

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE html>
<f:view xmlns:f="http://java.sun.com/jsf/core"
xmlns:af="http://xmlns.oracle.com/adf/faces/rich">
    <af:document title="jdlotusfeed.jsf" id="d1"
binding="#{backingBeanScope.backing_jdlotusfeed.d1}">
        <af:form id="f1" binding="#{backingBeanScope.backing_jdlotusfeed.f1}">
            <p xmlns="http://www.w3.org/1999/xhtml">

            </p>
            <p xmlns="http://www.w3.org/1999/xhtml">

            </p>
            <af:button text="button 1" id="b1"
binding="#{backingBeanScope.backing_jdlotusfeed.b1}"
          action="#{backingBeanScope.backing_jdlotusfeed.b1_action}"/>
            <p xmlns="http://www.w3.org/1999/xhtml">

            </p>
        </af:form>
    </af:document>
    <!--oracle-jdev-comment:auto-binding-backing-bean-name:backing_jdlotusfeed-->
</f:view>
```

Backing Bean, for above code

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

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

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

import oracle.adf.view.rich.component.rich.RichDocument;
import oracle.adf.view.rich.component.rich.RichForm;
import oracle.adf.view.rich.component.rich.nav.RichButton;

public class Jdlotusfeed {
    private RichForm f1;
    private RichDocument d1;
    private RichButton b1;

    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;
    }

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

        URL lotuNotesURL = new
URL("http://localhost/docucontent.nsf/javaagentfeedcsvexternalsites.txt");

           BufferedReader in = new BufferedReader(
           new InputStreamReader(lotuNotesURL.openStream()));

           String inputLine=null;

           while ((inputLine = in.readLine()) != null)
               System.out.println(inputLine);
           in.close();


        return null;
    }
}
```

version: 2015.07.05.12.08.AM