

Döcu Content

Google Worksheets

JavaAgent/JavaBean

Xpages Application

System Requirements:

Download Domino Designer **8.5.3** Environment (DDE)

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

Download /Make available Google Spreadsheets API

<https://developers.google.com/google-apps/spreadsheets/>

<https://developers.google.com/google-apps/documents-list/>

Introduction:

In this tutorial, we are connecting to Google to read a Spreadsheet and listing its Worksheets using a button. Slight modification is made to existing code to convert to JavaAgent, JavaBean written to run code through Browser. See below for code samples to use to run the program, Xpages XSP file included..

Disclaimer:

Information contained in the following is presented as is. This tutorial assumes you have basic Lotus Notes Configuration, Programming knowledge, and are familiar with Google APIs.

Döcu Content JavaAgent/JavaBean/Xpages

At this point, we assume you have gone through the prior tutorial, and maintained JAR configuration to be able to use the Google Spreadsheet API. Copy and paste code below into DDE and submit a new Worksheet, areas of interests are highlighted for your convenience.

Related Info:

<https://www.youtube.com/watch?v=GCPsQwv7xVo&list=UUSImDTpK0oe7QrPsYOE4nww>

Copy and Paste Page Design code

More than likely, you have also configured faces config XML file in this application. Load the Xpage to your Browser and click that button.

ReadGoogleDriveWorkSheetJavaAgent.java;

```
/**
 * Created from copy: 2014.05.15.9.14.PM
 * GoogleSpreadsheetsGoodJavaAgentCopy | ReadGoogleDriveWorkSheetJavaAgent.java
 * List Worksheets in specific Spreadsheet, housed on Google Drive
 */
```

```

//...
//Google imports
import com.google.gdata.client.spreadsheet.SpreadsheetService;
import com.google.gdata.data.PlainTextConstruct;
import com.google.gdata.data.spreadsheet.SpreadsheetEntry;
import com.google.gdata.data.spreadsheet.SpreadsheetFeed;
import com.google.gdata.data.spreadsheet.WorksheetEntry;
import com.google.gdata.util.AuthenticationException;
import com.google.gdata.util.ServiceException;

//...
//Java imports
import java.io.IOException;
import java.net.URL;
import java.util.List;

//...
//Lotus Domino imports
import lotus.domino.AgentBase;

/**
 * @author Dököll Solutions, Inc.
 * @version 2014.05.15.9.14.PM
 *
 */

public class ReadGoogleDriveWorkSheetJavaAgent extends AgentBase {

    public ReadGoogleDriveWorkSheetJavaAgent() {
        // ...
    }

    // button code...
    public void NotesMain() {

        try {

            //TODO: see about connecting automatically...
            //existing Google Drive login creds...
            String USERNAME = "yourgoogleaccount@gmail.com";
            String PASSWORD = "youraccountpassword";

            SpreadsheetService service = new SpreadsheetService(
                "MySpreadsheetIntegration");
            service.setUserCredentials(USERNAME, PASSWORD);

            // TODO: Authorize the service object for a specific user (see other
            // sections)

            // Define the URL to request. This should never change.
            URL SPREADSHEET_FEED_URL = new URL(
                "https://spreadsheets.google.com/feeds/spreadsheets/private/full");

            // Make a request to the API and get all spreadsheets.
            SpreadsheetFeed feed = service.getFeed(SPREADSHEET_FEED_URL,
                SpreadsheetFeed.class);
            List<SpreadsheetEntry> spreadsheets = feed.getEntries();

            if (spreadsheets.size() == 0) {
                // TODO: There were no spreadsheets, act accordingly.
            }
        }
    }
}

```



```

* @version 2014.05.15.9.16.AM
*
*/

//...
public class RunGoogleSpreadsheetsBean {

    //button code
    public void doRunSpreadsheetData() throws NotesException{
        // set up the loggers
        Log log = LogFactory.getLog(RunGoogleSpreadsheetsBean.class);
        Log debug = LogFactory.getLog("DEBUG");

        // tell what they are
        System.out.println("log is a " + log.getClass().getName());
        System.out.println("debug is a " + debug.getClass().getName());

        log.info("an info message");
        debug.debug("a debug message");

        log.info("another info message");
        debug.debug("another debug message");
        //...
        //get the current database being used
        Database database= (Database) FacesContext.getCurrentInstance()
            .getApplication().getVariableResolver()
            .resolveVariable(FacesContext.getCurrentInstance(), "database");
        System.out.println("Database Obtained..." + database);
        System.out.println("Connecting to Google...");
        //...
        //Run the agent...

        database.getAgent("GoogleSpreadsheetsGoodJavaAgentCopy").runOnServer();
    }
}

```

xprungooglespreadsheets.xsp;

```

<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core">
    <xp:this.resources>
        <xp:script src="/docucontentGACodeTest.js" clientSide="true" />
    </xp:this.resources>
    <html>
        <head>
            <title>Google Spreadsheets API and Google Drive</title>
        </head>
        <xp:br></xp:br>
        <xp:div style="text-align:center">
            <xp:span style="font-weight:bold;font-size:28pt;background-
color:rgb(255,255,128)">
                </xp:span>
            <body style="font-weight:bold;font-size:10pt;text-
align:center;background-color:rgb(255,255,128)">
                <xp:form id="formId" style="font-weight:bold;font-

```

```

size:11pt;text-align:center;background-color:rgb(255,255,128)">

        <table style="font-weight:bold;font-
size:10pt;text-align:center;background-color:rgb(255,255,128)">

                <tr style="font-weight:bold;font-
size:11pt;text-align:center;background-color:rgb(255,255,128)">

                        <td style="font-weight:bold;font-
size:12pt;text-align:center;background-color:rgb(255,255,128)">

                                <xp:div style="text-
align:center;background-color:rgb(255,255,128)"><xp:span style="font-
weight:bold;font-size:28pt;background-color:rgb(255,255,128)">Google
Spreadsheets - Google Drive</xp:span><xp:span style="font-size:19pt;background-
color:rgb(255,255,128)"></xp:span></xp:div><xp:div style="text-
align:center;background-color:rgb(255,255,128)">
                                <xp:button value="Submit"
id="button1" style="width:168.0px;font-weight:bold;font-size:10pt;background-
color:rgb(255,255,128)">
                                        <xp:EventHandler
event="onclick" submit="true" refreshMode="complete" immediate="false"
save="true" id="eventHandler1">
                                                <xp:this.action><!
[CDATA[#{javascript:RunGoogleSpreadsheetsBean.doRunSpreadsheetData()}]]></xp:thi
s.action>
                                                        </xp:EventHandler>
                                                                </xp:button></xp:div><xp:div
style="text-align:center;background-color:rgb(255,255,128);font-size:21pt">
                                                                Create&#160;Worksheet(s)</xp:div></td>
                                </tr>
                        </table>

                </xp:form>

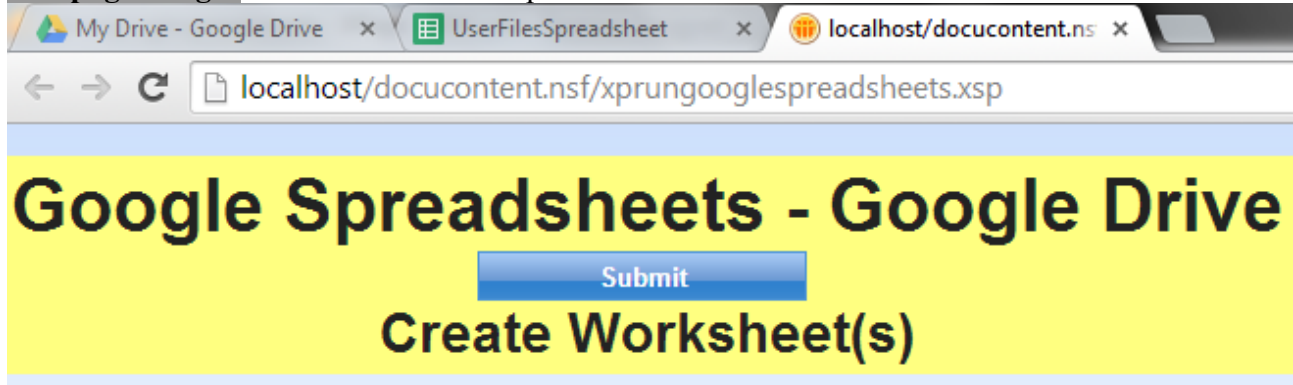
</body></xp:div></html>

</xp:view>

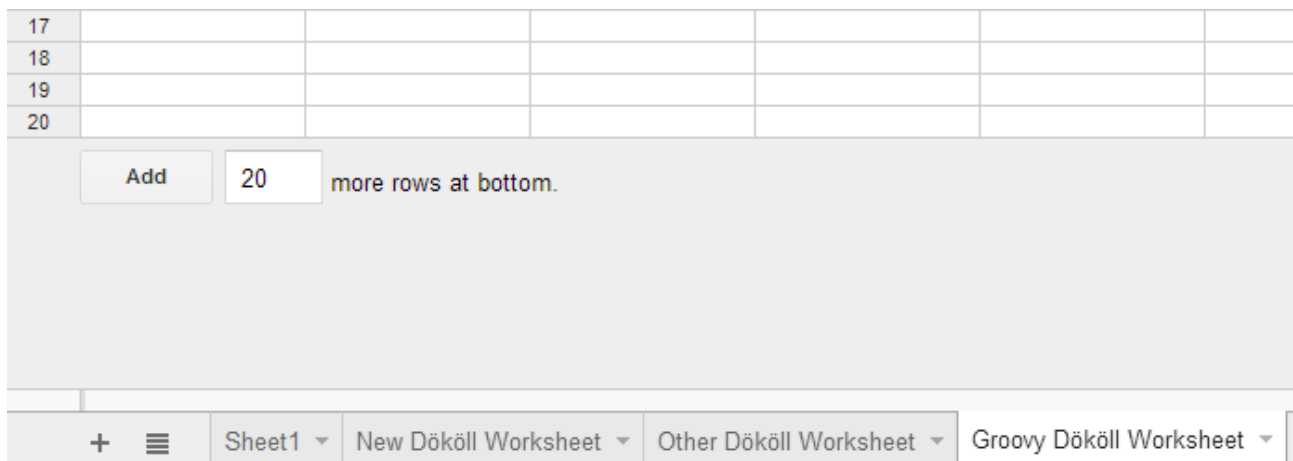
```

Döcu Content Console View (Console/Drive)

Full page design: Here is the button in question...



bottom part of the document



TIP: You may want to Extract part of the Spreadsheet listing code to loop through and list worksheet(s) by name via yet another Xpages form, or to existing submit page...

Conclusion:

You can now run code from a button to add Worksheets to existing Spreadsheet, publically available on Google Drive.

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

Thank you for coming...

Version:2014.05.17.1.41.AM