

Döcu Content

Google Worksheets Row Insert

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 attempting to connect to Google Drive Spreadsheet, insert row(s) to the second Worksheet in the Spreadsheet, with and a button on an Xpages Form and using the beforePageLoad feature in Xpages. Slight modification again is made to existing code to convert to JavaAgent, JavaBean written to run code through Browser and submit record to Google via via NSF back-end. See below for code samples to use to run the JavaAgent program, using Xpages file...

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=Rh0DJ8KHWzw&list=UUSImDTpK0oe7QrPsYOE4nww>

<https://www.youtube.com/watch?v=hemoGCiygZA&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.

WorkSheetsAddJavaAgent.java;

```

/**
 * Created from copy: 2014.05.17.1.14.AM
 * GoogleAddRowJavaAgent | WorksheetsAddJavaAgent.java
 * Inser Row(s) to 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 2014.05.17.1.14.AM
 *
 */

public class WorksheetsAddJavaAgent extends AgentBase {

    public WorksheetsAddJavaAgent() {
        // ...
    }

    // 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(
                "CompanyWorksheetsAddRow");
            try {
                service.setUserCredentials(USERNAME, PASSWORD);
            } catch (AuthenticationException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            // 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 = null;

```

```

    try {
        SPREADSHEET_FEED_URL = new URL(
"https://spreadsheets.google.com/feeds/spreadsheets/private/full");
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

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

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

        // TODO: Choose a spreadsheet more intelligently based on your
        // app's needs.
        SpreadsheetEntry spreadsheet = spreadsheets.get(0);
        System.out.println(spreadsheet.getTitle().getPlainText());

        // Get the first worksheet of the first spreadsheet.
        // TODO: Choose a worksheet more intelligently based on your
        // app's needs.
        WorksheetFeed worksheetFeed = service.getFeed(
            spreadsheet.getWorksheetFeedUrl(), WorksheetFeed.class);
        List<WorksheetEntry> worksheets = worksheetFeed.getEntries();
        WorksheetEntry worksheet = worksheets.get(1);

        // Fetch the list feed of the worksheet.
        URL listFeedUrl = worksheet.getListFeedUrl();
        ListFeed listFeed = (ListFeed) service.getFeed(listFeedUrl, ListFeed.class);

        // Create a local representation of the new row.
        ListEntry row = new ListEntry();
        row.getCustomElements().setValueLocal("firstname", "Thierry");
        row.getCustomElements().setValueLocal("lastname", "Polinis");
        row.getCustomElements().setValueLocal("age", "44");
        row.getCustomElements().setValueLocal("height", "178");

        // Send the new row to the API for insertion.
        row = (ListEntry) service.insert(listFeedUrl, row);
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (ServiceException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

Relate info

Below is an excerpt of similar the above, at the risk of adding redundant data to this tutorial; the only difference is with setLocalValue, have a look below

for the modification needed.

Note: Do change code in your Xpages form however, different method name being used.

```
/**
 * Created from copy: 2014.05.17.2.34.AM
 * GoogleAddRowForXpagesJavaAgent | WorksheetsAddJavaAgent.java
 * Inser Row(s) to Worksheets in specific Spreadsheet, housed on Google Drive
 */
//...
//Google imports

...

// Create a local representation of the new row.
ListEntry row = new ListEntry();
row.getCustomElements().setValueLocal("firstname", "0");
row.getCustomElements().setValueLocal("lastname", "Hi");
row.getCustomElements().setValueLocal("age", "56");
row.getCustomElements().setValueLocal("height", "173");

...
```

Döcu Content JavaBean/Xpages Button

We are using a copy of prior code to run Xpages and insert into Worksheets on Google Drive using a button. Copy and paste code below into DDE and run it...

UpdateGoogleWorksheetsBean.java;

```
/**
 * Created from Copy: 2014.05.18.1.29.AM
 * UpdateGoogleWorksheetsBean.java
 * Run JavaAgent, Connect to Google | Insert Row(s) to Existing Worksheet
 */
package com.dokoll.solutions.inc.google.dev;

//...
//Faces imports
import javax.faces.context.FacesContext;
//...
//Domino imports
import lotus.domino.NotesException;
import lotus.domino.local.Database;
//...
//Apache imports
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

/**
 * @author Dököll Solutions, Inc.
 * @version 2014.05.18.1.29.AM
 *
 */

//...
public class UpdateGoogleWorksheetsBean {

    //...
```

```

//get the current database being used
Database database= (Database) FacesContext.getCurrentInstance()
.getApplication().getVariableResolver()
.resolveVariable(FacesContext.getCurrentInstance(), "database");

//button code
public void doAddWorksheetData () throws NotesException{
    // set up the loggers
    Log log = LogFactory.getLog(UpdateGoogleWorksheetsBean.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("a Worksheet info message");
    debug.debug("a Worksheet debug message");

    log.info("another Worksheet info message");
    debug.debug("another Worksheet debug message");

    //...
    System.out.println("Database Obtained..." + database);
    System.out.println("Connecting to Google, inserting rows to
Worksheets...");

    //Run the agent...

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

//button code
public void doAddRowWorksheetData () throws
NotesException{
    // set up the loggers
    Log log = LogFactory.getLog(UpdateGoogleWorksheetsBean.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("another Worksheet info message");
    debug.debug("a Worksheet debug message");

    log.info("totally awesome Worksheet info");
    debug.debug("Worksheet debug message to the max");
    //...

    System.out.println("Connecting to Notes Agent, inserting rows to
Worksheets...");
    //...
    //Run the agent...

    database.getAgent("GoogleAddRowForXpagesJavaAgentAlias").ru

```

```
nOnServer();
    }
```

```
}
```

xpgoogleinsertrow.xsp;

```
<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core">
    <xp:this.beforePageLoad><![
[CDATA[#{javascript:UpdateGoogleWorksheetsBean.doAddRowWorksheetData()}]]
]></xp:this.beforePageLoad>
</xp:view>
```

xpgoogleinsertsheetdata.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 -
Worksheets</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-
```

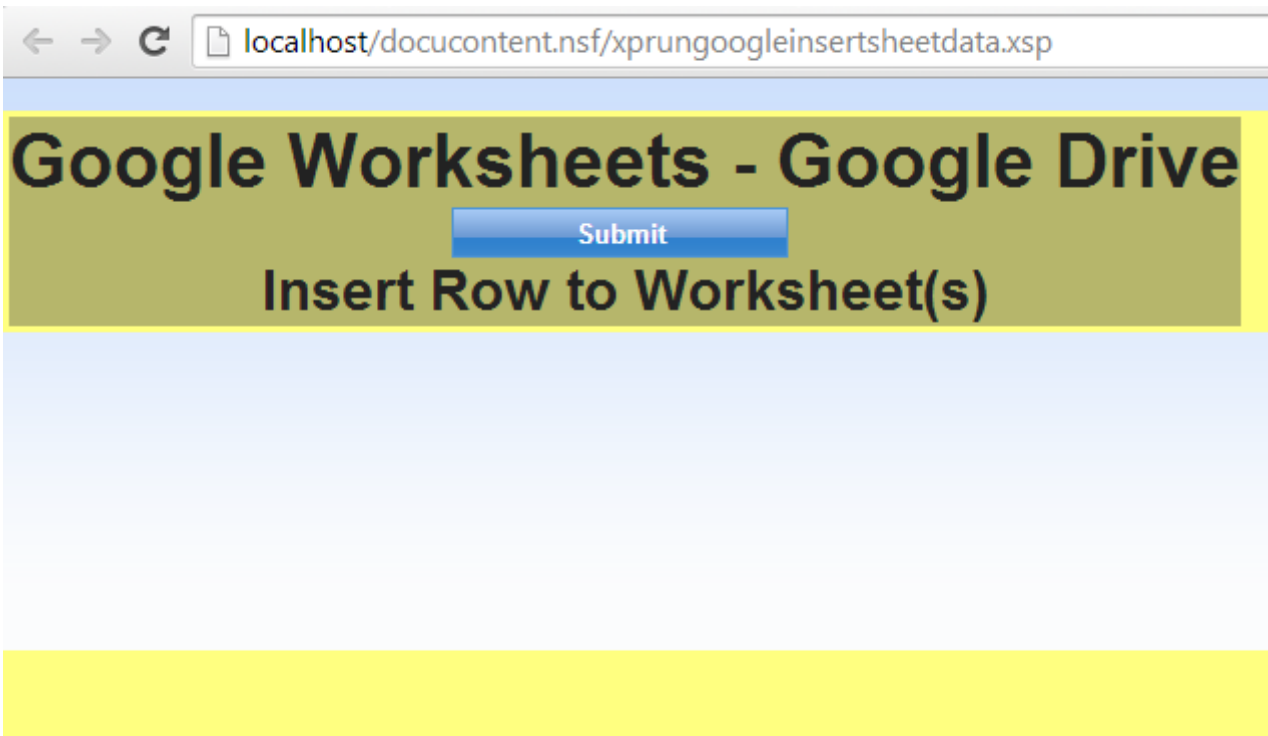
```

align:center;background-color:rgb(182,182,107)">
weight:bold;font-size:28pt;background-color:rgb(182,182,107)">
- Google Drive
size:19pt;background-color:rgb(255,255,128)">
align:center;background-color:rgb(182,182,107)">
style="width:168.0px;font-weight:bold;font-size:10pt;background-
color:rgb(255,255,128)">
event="onclick"
refreshMode="complete" immediate="false"
id="eventHandler1">
<xp:this.action><!
[CDATA[#{javascript:UpdateGoogleWorksheetsBean.doAddWorksheetData()}
]]></xp:this.action>
Insert Row to&#160;Worksheet(s)
</tr>
</table>
</xp:form>
</body></xp:div></html>
</xp:view>

```

Döcu Content Worksheets View (Xpages/Google Drive)

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



current cell records

UserFilesSpreadsheet ☆

File Edit View Insert Format Data Tools Help All changes saved in Drive

fx | firstname

	A	B	C	D	E
1	firstname	lastname	age	height	
2	Metal	Mania	28	144	
3	O	Hi	56	173	
4	Thierry	Polinis	44	178	
5					

TIP: Be sure to login to Google Drive account prior to launching code...

Conclusion:

You can now run code from a button to insert row(s) to existing Worksheets, 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:2014.05.18.1.56.AM