

Döcu Content

Google Spreadsheets API

Script Libray

JavaAgent/Xpages

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 using a Java console program to list the name of a certain Spreadsheet document housed on Google Drive, using the Spreadsheets API. We will be reading publically available document(s), previously created... We will be programming via Domino Desginer Environement (DDE).

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 (Script Library)

Go to URLs provided, download JAR files, then create a Java Script Library in DDE with the jar files contained in the zip (**gdata-src.java-1.47.1.zip**). Reference the Script Library in your JavaAgent program, run code to list Spreadsheet info to console window....

Take a look at the JAR file section below, view screenshots; you may need to remove deprecated JARs from your download. See 'Hint' in this document.

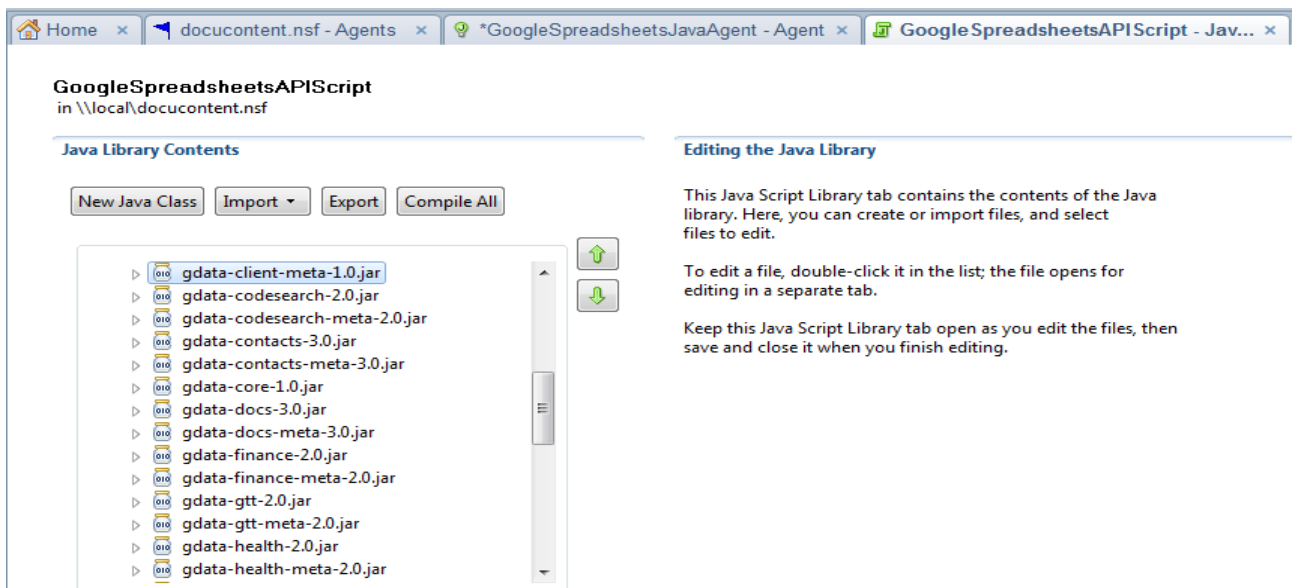
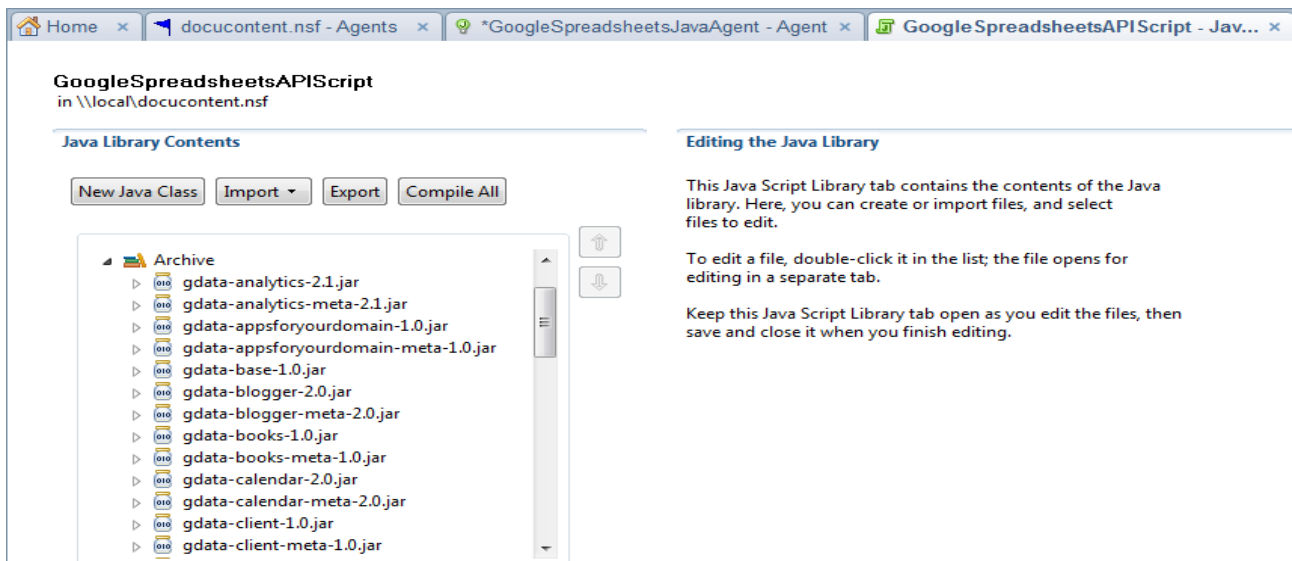
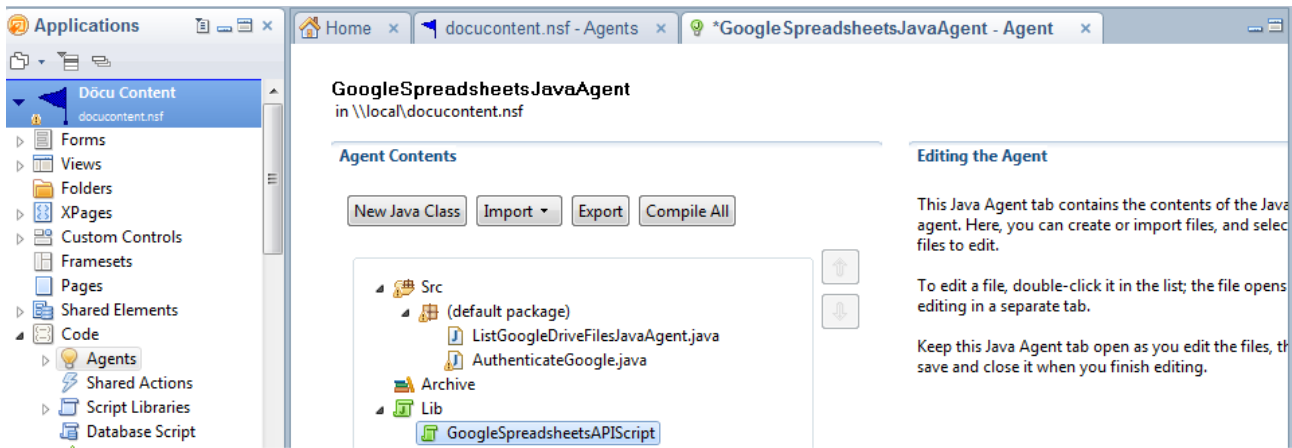
Related Info:

<https://www.youtube.com/watch?v=2TtnqJMnXg&list=UUSImDTpK0oe7QrPsYOE4nww>

<https://www.youtube.com/watch?v=7U1HP5yEhI8&list=UUSImDTpK0oe7QrPsYOE4nww>

Döcu Content Design JARs

User configuration and design: the following library files are needed as Script files... Add Script as an Archive item. *Besire to navigate to lib folder to include all JAR files from download...*

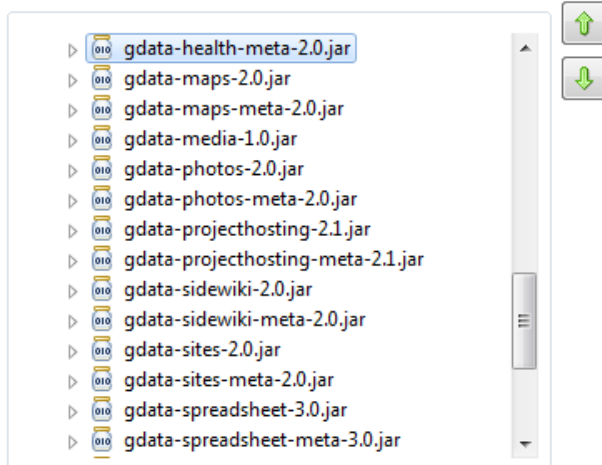


GoogleSpreadsheetsAPIScript

in \\local\docucontent.nsf

Java Library Contents

New Java Class Import Export Compile All



Editing the Java Library

This Java Script Library tab contains the contents of the Java library. Here, you can create or import files, and select files to edit.

To edit a file, double-click it in the list; the file opens for editing in a separate tab.

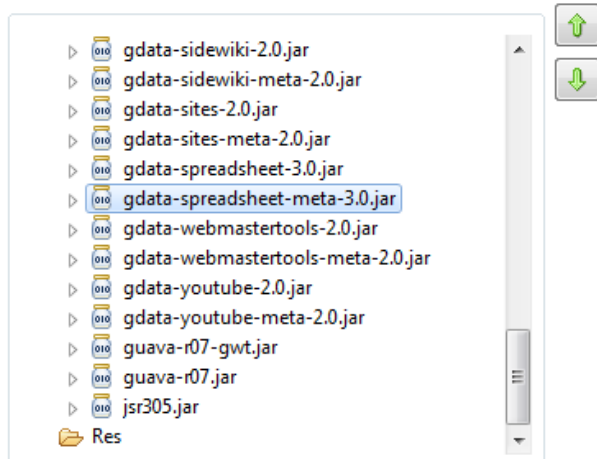
Keep this Java Script Library tab open as you edit the files, then save and close it when you finish editing.

GoogleSpreadsheetsAPIScript

in \\local\docucontent.nsf

Java Library Contents

New Java Class Import Export Compile All



Editing the Java Library

This Java Script Library tab contains the contents of the Java library. Here, you can create or import files, and select files to edit.

To edit a file, double-click it in the list; the file opens for editing in a separate tab.

Keep this Java Script Library tab open as you edit the files, then save and close it when you finish editing.

HINT:

You may run into an issue where your code cannot connect to Google, it could be because your Script Library contains one of the deprecated JARs, '*google-collect-1.0-rc1.jar*'... Delete it, save Script Library, save the program, and try again.

Copy and Paste Page Design code

More than likely, with the proper JAR files in place, the program compiles, run as Java program and view results in Console

ListGoogleDriveFilesJavaAgent.java;

```
/**
 * Created from copy: 2014.05.11.9.05.PM
 * GoogleSpreadsheetsJavaAgent | ListGoogleDriveFilesJavaAgent.java
 * List Spreadsheets in specific Google Drive Account
 */

/**
 * @author Dököll Solutions, Inc.
 * @version 2014.05.11.9.05.PM
 *
 */
//Google imports...
//...
import com.google.gdata.client.spreadsheet.*;
import com.google.gdata.data.spreadsheet.*;
import com.google.gdata.util.*;

//java imports...
//...
import java.io.IOException;
import java.net.*;
import java.util.*;

public class ListGoogleDriveFilesJavaAgent {
    public static void main(String[] args)
        throws AuthenticationException, MalformedURLException, IOException,
        ServiceException {

        //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: for code to put here...

        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();

        // Iterate through all of the spreadsheets returned
        for (SpreadsheetEntry spreadsheet : spreadsheets) {
            // Print the title of this spreadsheet to the screen
            System.out.println(spreadsheet.getTitle().getPlainText());

            // Make a request to the API to fetch information about all
            // worksheets in the spreadsheet.
            List<WorksheetEntry> worksheets = spreadsheet.getWorksheets();

            // Iterate through each worksheet in the spreadsheet.
            for (WorksheetEntry worksheet : worksheets) {
```

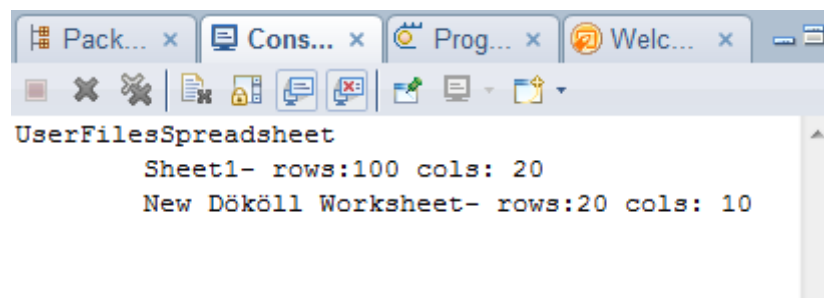
```
// Get the worksheet's title, row count, and column count.
String title = worksheet.getTitle().getPlainText();
int rowCount = worksheet.getRowCount();
int colCount = worksheet.getColCount();

// Print the fetched information to the screen for this worksheet.
System.out.println("\t" + title + "- rows:" + rowCount + " cols: " +
colCount);
} //for worksheet
} //for spreadsheet

}
}
```

Döcu Content Console View (Console)

Full page design: Here is what your results might look like when program runs...



Conclusion:

Access publically available document(s) housed on Google's Infrastructure with the push of a button using the Google APIs.

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

Thank you for coming...

Version:2014.05.11.8.54.PM