

```

/*****
/* PROGRAM: skyAPI-STAFF
/*
/* Developer: D. Morton (@DEMPanther)
/* Purpose: Utilize Google Web App & Blackbaud SkyAPI to present the latest staff
/*           contact information in Google Apps spreadsheet format
/* NOTE: Makes use of Oauth2 library from Google Workspace team (@workspacedevs).
/*       Google Apps Script security prevents redirect to Oauth2 authentication
/*       page without user intervention. The Oauth2 library provides a framework
/*       for managing the process in a "serverless" client environment
/*-----

/*****
/* MAIN: doGet - Initiated upon HTML GET action to Google Web App
/*-----
/* INPUT: N/A
/* OUTPUT: Html page with link to Blackbaud Oauth2 verification process
/* ACTION: Create an OAuth2 service and present a page requiring user action
/*         to initiate the Oauth2 verification process
/*****
function doGet(e) {

    var sheetURL = 'https://docs.google.com/spreadsheets/>>[YOUR SHEET ID]<<';

    // Create Oauth2 service
    var skyService = getSkyService();

    // if (!skyService.hasAccess()) {
    if (true) {
        var authorizationUrl = skyService.getAuthorizationUrl();
        var template = HtmlService.createTemplate(
            '<html>'+
            '<style>'+
            'h1 {font-family:"Roboto", sans-serif;color:#FFFFFF;background-color: #003b5c}'+
            'p{font-family:"Roboto", sans-serif;}'+
            'a.button3{'+
            'display:inline-block;'+
            'padding:0.3em 1.2em;'+
            'margin:0 0.3em 0.3em 0;'+
            'border-radius:2em;'+
            'box-sizing: border-box;'+
            'text-decoration:none;'+
            'font-family:"Roboto", sans-serif;'+
            'font-weight: bold;'+
            'color:#FFFFFF;'+
            'background-color:#003b5c;'+
            'text-align:center;'+

```

```

    'transition: all 0.2s;'+
    '}'+'
    'a.button3:hover{'+'
    'background-color:#e4002b;'+
    '}'+'
    '@media all and (max-width:30em){'+
    'a.button3{'+'
    'display:block;'+
    'margin:0.2em auto;'+
    '}'+'
    '}'+'
    '</style>'+
    '<script>'+
    'function authURL() {'+'
    '  window.open(<?= authorizationUrl ?>);'+
    '  window.open(""+sheetURL+"", "_self");'+
    '}'+'
    '</script>'+
    '// '<br><center><a href="<?= authorizationUrl ?>" target="_blank"><button>Signin
With Google</button></a></center>'
    '//   '<body><br><center><button onclick="authURL()">Signin With
Google</button></a></center></body>'
    ' '<body><center><h1>Staff Contact and Car Directory</strong></h1>'+
    ' '<p>Load a QUICK version of the list or the LATEST SYNC (Sign in with
Google) from Sheridan Central</p>'+
    ' '<p><b>NOTE: Your first time synching will require a few extra clicks /
Sign-in with Google to approve this app</b></p>'+
    'v1.01'+
    ' '<hr><br>'+
    ' '<a class="button3" href="'+sheetURL+'">QUICK</a>'+
    ' '<a class="button3" onclick="authURL()">LATEST SYNC</a>'+
    ' '</center></body>'+
    '</html>'
  );
  template.authorizationUrl = authorizationUrl;
  var page = template.evaluate();

  Logger.log('AUTH LINK:\n'+page.getContent());
  Logger.log('<[doGet]\n\n');

  return HtmlService.createHtmlOutput(page);
}

} // **[ doGet() ] **

```

```

/*****
/* authCallback - Initiated upon callback from Oauth2 provider
/*-----
/* INPUT:  Oauth2 callback parameters
/* OUTPUT: Display staff data if user is authorized otherwise display error page
/* ACTION: Opens staff contact list spreadsheet to present staff data and
/*         closes the callback page if user is authorized otherwise displays
/*         error page.
/*****
function authCallback(e){

    var params = JSON.stringify(e);
    var skyService = getSkyService();

    var isAuthorized = skyService.handleCallback(e);
    if (isAuthorized) {
        var rc = makeRequest();
        Logger.log('<[authCallback]');
        return HtmlService.createHtmlOutputFromFile('closeCallback');

    }
    else {
        Logger.log('<[authCallback]');
        return HtmlService.createHtmlOutputFromFile('dataRequestErr');
    }

}

/*****
/* makeRequest - Make HTML GET request to Oauth2 provider for data and display
/*             the data.
/*-----
/* INPUT:  N/A
/* OUTPUT: Display Staff Contact List spreadsheet
/* ACTION: Make call to SkyAPI to retrieve Blackbaud user data in JSON format
/*         and load it into a Google Apps spreadsheet that is displayed to users.
/*****
function makeRequest() {

    var skyService = getSkyService();

    // SkyAPI Users call for Non-teaching staff (Role ID 332) and teachers (Role ID 15)
    var response =
    UrlFetchApp.fetch('https://api.sky.blackbaud.com/school/v1/users/extended?base_role_id
s=332,15', {

```

```

    headers: {
        Authorization: 'Bearer ' + skyService.getAccessToken(),
        'Bb-API-Subscription-Key' : '>>[YOUR BB API SUBSCRIPTION KEY]<<'
    }
});

// Data returned from SkyAPI call
var jsonText = response.getContentText();
var jsonObj={};
var jsonItemObj={};
var jsonTot=0;
var jsonValues=[];
var jsonUserid='';
jsonObj = JSON.parse(jsonText);
jsonTot = jsonObj.count;

Logger.log('JSON ITEMS: '+jsonTot);
    jsonValues=jsonObj.value;
    // Load staff contact data page with staff information
rc = loadData(jsonValues,jsonTot);
// rc= logout();

Logger.log('<[makeRequest]');
}

/*****
/* loadData - Open Staff Contact List spreadsheet and load SkyAPI JSON data into
/*           the spreadsheet after back up of the previous spreadsheet data to
/*           another sheet.
/*-----
/* INPUT:  staff JSON data, total staff member count
/* OUTPUT: Staff Contact List Google Apps spreadsheet populated with SkyAPI staff
/*         data
/* ACTION: Open the Staff Contact List spreadsheet, backup previous contact list
/*         data to the backup sheet and load SkyAPI returned JSON staff data into
/*         the top sheet
*****/
function loadData(staff, staffTotal) {

    var thisUserID="";
    var thisUserFNAME="";
    var thisUserLNAME="";
    var thisUserOCCUPATIONS=[];
    var thisUserTITLE="";
    var thisUserEXT="";
    var thisUserPHONES=[];
    var thisUserPHONE={};

```

```

var thisUserCELL="";
var thisUserEMAIL="";
var thisUserADDRS=[];
var thisUserADDR={};
var thisUserSTREET="";
var thisUserCITY="";
var thisUserSTATE="";
var thisUserZIP="";
var thisUserAUTOS="";
var ctrRow=3;
var sheetName='Staff';
var sheetRange='';
var sheetID='>>[YOUR GOOGLE SHEET ID]<<';

// Open the staff contact spreadsheet
var ss = SpreadsheetApp.openById(sheetID);
var totSheets = ss.getNumSheets();
var sheets = ss.getSheets();

var sheetname1=sheets[0].getName();
var staffSheet = ss.getSheetByName(sheetName);
if(totSheets > 1 ) {
    var sheetname2=sheets[1].getName();
    var backupSheet = ss.getSheetByName(sheetname2);
}

if (staffTotal > 0)
{
    // copy Current Staff sheet to backup sheet
    var copyRange = staffSheet.getRange("A3:L80");
    if(totSheets > 1)
    {
        var toRange = backupSheet.getRange("A3:L80");
        Logger.log('OLD SHEET VALUE [' +copyRange.getCell(10,1).getDisplayValue()+'] NOT
Blank? '+copyRange.getCell(10,1).getDisplayValue() > " ");
        if(copyRange.getCell(10,1).getDisplayValue()>" ")
        {
            Logger.log('BACKUP SHEET');
            var tz = ss.getSpreadsheetTimeZone();
            var date = 'Prev. Vers.:'+Utilities.formatDate(new Date(), tz, 'MM-dd-yy
HH:mm');
            backupSheet.setName(date); // Rename second
            copyRange.copyTo(toRange);
        }
    }
    else
    {
        Logger.log('>>NO BACKUP');
    }
}

```

```

    }
}

// Clear the main sheet
staffSheet.clear({contentsOnly: true });
var rows = [];
rows.push(['This data is populated from Blackbaud.\nPlease DO NOT make changes in
this file.\nUpdates can be applied in your Sheridan Central profile.']);
var sheetRange='A1:A1';
var staffRange = staffSheet.getRange(sheetRange);
staffRange.setValues(rows);
var rows = [];
rows.push(['SC ID', 'First Name', 'Last
Name', 'Cell', 'Autos', 'Title', 'Ext.', 'Email', 'Address', 'City', 'State', 'Zip']);
var sheetRange='A2:L2';
var staffRange = staffSheet.getRange(sheetRange);
staffRange.setValues(rows);
}

// Loop through JSON data and populate rows in the spreadsheet
for(staffCtr=0;staffCtr < staffTotal;staffCtr++)
{
thisUserObj=staff[staffCtr];

thisUserID=thisUserObj.id;
thisUserFNAME=thisUserObj.first_name;
thisUserLNAME=thisUserObj.last_name;
thisUserOCCUPATIONS=thisUserObj.occupations;
if(thisUserOCCUPATIONS.length > 0)
{
thisUserTITLE=thisUserOCCUPATIONS[0].title;
}
else
{
thisUserTITLE="";
}
// thisUserEXT=thisUserObj.;
thisUserPHONES=thisUserObj.phones;

thisUserEXT="N/A";
thisUserCELL="TBD";

if(thisUserPHONES.length > 0)
{
for(ctrPhone=0;ctrPhone < thisUserPHONES.length;ctrPhone++ )
{
if (thisUserPHONES[ctrPhone].phone_type.substr(0,3)=="Wir") //Wireless

```

```

        {
            thisUserCELL=thisUserPHONES[ctrPhone].number;
        }
        if (thisUserPHONES[ctrPhone].phone_type.substr(0,3)=="Bus") // Business
        {
            thisUserEXT=thisUserPHONES[ctrPhone].number;
        }
    }

}

thisUserEMAIL=thisUserObj.email;

thisUserADDRS=thisUserObj.addresses;
if(thisUserADDRS.length > 0)
{
    thisUserSTREET=thisUserADDRS[0].line_one;
    thisUserCITY=thisUserADDRS[0].city;
    thisUserSTATE=thisUserADDRS[0].state;
    thisUserZIP=thisUserADDRS[0].postal_code;
}
else
{
    thisUserSTREET="";
    thisUserCITY="";
    thisUserSTATE="";
    thisUserZIP="";
    thisUserAUTOS="";
}
thisUserAUTOS=thisUserObj.custom_field_five;

// Exclude test users (last name starting with "_") from contact list
if(thisUserLNAME.substr(0,1)!=="_")
{
    var rows = [];

    rows.push([thisUserID, thisUserFNAME, thisUserLNAME, thisUserCELL,
thisUserAUTOS, thisUserTITLE, thisUserEXT ,    thisUserEMAIL, thisUserSTREET,
thisUserCITY, thisUserSTATE, thisUserZIP]);

    var sheetRange='A'+(1*ctrRow)+' :L'+(1*ctrRow);
    var staffRange = staffSheet.getRange(sheetRange);
    staffRange.setValues(rows);
    ctrRow++;

}
else

```

```

    {
    Logger.log('TEST USER EXCLUDED');
    }

    Logger.log('User '+staffCtr+ ' : '+thisUserLNAME+ ' :'+thisUserPHONES);

    }
    // Sort the sheet
    var srRange='A3:L'+(1*(ctrRow-1));
    sortRange = staffSheet.getRange(srRange);
    // Sort sheet by last name, first name
    sortRange.sort([{column: 3, ascending: true},{column: 2, ascending: true}])
}

/*****
/* MAIN: getSkyService
/*-----
/* INPUT: N/A
/* OUTPUT: New service named "sky"
/* ACTION: Create new service with static parameters
*****/
function getSkyService() {
    // Create a new service with the given name. The name will be used when
    // persisting the authorized token, so ensure it is unique within the
    // scope of the property store.

    return OAuth2.createService('sky')

        // Set the OAuth2 endpoint URLs
        .setAuthorizationBaseUrl('https://oauth2.sky.blackbaud.com/authorization')
        .setTokenUrl('https://oauth2.sky.blackbaud.com/token')

        // Set the client ID and secret, from Blackbaud
        .setClientId('>[BB CLIENT ID]<<')
        .setClientSecret('>[BB CLIENT SECRET]<<')

        // Set the name of the callback function in the script referenced
        // above that should be invoked to complete the OAuth flow.
        .setCallbackFunction('authCallback')

        // Set the property store where authorized tokens should be persisted.
        //.setPropertyStore(PropertiesService.getUserProperties())
        .setPropertyStore(PropertiesService.getScriptProperties())

        // Set the scopes to request (space-separated for Google services).

```



```
.setScope('https://www.googleapis.com/auth/script.external_request')
.setScope('https://www.googleapis.com/auth/spreadsheets')
//.setScope('https://www.googleapis.com/auth/script.container.ui')
//.setScope('https://www.googleapis.com/auth/userinfo.email')
//.setScope('https://www.googleapis.com/auth/datastore')
.setScope('https://www.googleapis.com/auth/cloud-platform')
.setScope('https://www.googleapis.com/auth/logging.write')
.setScope('https://www.googleapis.com/auth/monitoring')
.setScope('https://www.googleapis.com/auth/devstorage.full_control')
.setScope('https://www.googleapis.com/auth/cloud_debugger')
.setScope('https://www.googleapis.com/auth/userinfo.email')
//.setScope('https://www.googleapis.com/auth/userinfo.profile')
.setScope('https://www.googleapis.com/auth/drive.file')
//.setScope('https://www.googleapis.com/auth/drive')

// Below are Google-specific OAuth2 parameters.

// Sets the login hint, which will prevent the account chooser screen
// from being shown to users logged in with multiple accounts.
.setParam('login_hint', Session.getEffectiveUser().getEmail())
//.setParam('login_hint', Session.getActiveUser().getEmail())

// Requests offline access.
.setParam('access_type', 'offline')

// Consent prompt is required to ensure a refresh token is always
// returned when requesting offline access.
.setParam('prompt', 'consent');
}
```