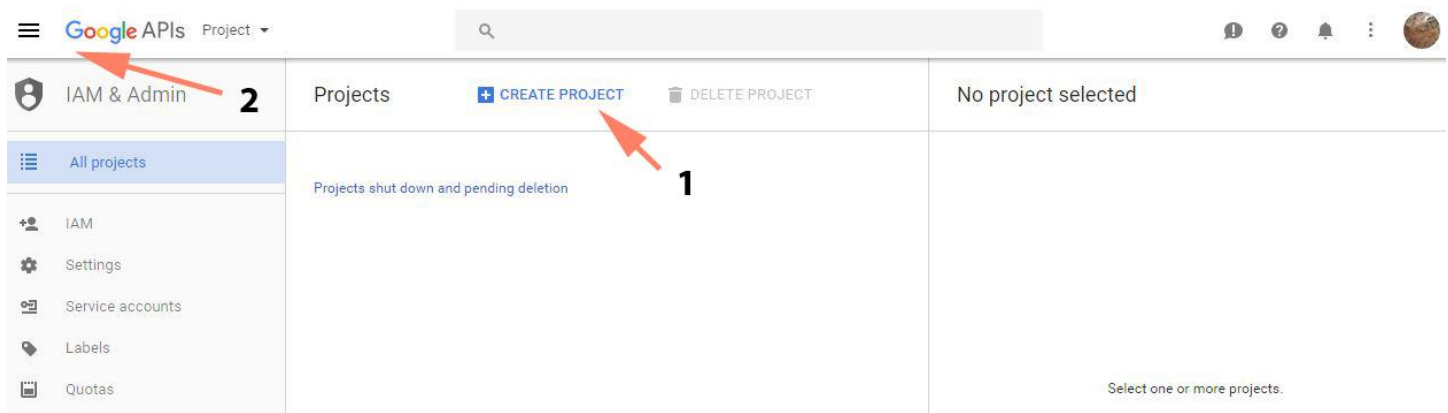


## README FOR MAT2SHEETS

```
function status = mat2sheets(spreadsheetID, sheetID, sheetpos, d)
% This function takes values from an array or cell array and places them in
% a Google spreadsheet. It requires the one-time use of RunOnce (see below)
% SYNTAX: status = mat2sheets(spreadsheetID, sheetID, pos, d)
%
% ARGUMENTS:
%     spreadsheetID: (string), identifier from URL of your Google Sheet
%     sheetID:       (string), another identifier from URL
%     pos:           1x2 array with indices for [sheetrow, sheetcolumn]
%                   to start pasting data
%     d:             array or cell array of data to paste into Sheet
%
% RETURNS: status (0=failed, 1=success)
%
% EXAMPLES:
%     For sheet with the following URL:
%     https://docs.google.com/spreadsheets/d/1GPd-vBsX5VUejz5hrxE/edit#gid=552
%
%     A call may look like:
%     mat2sheets('1GPd-vBsX5VUejz5hrxE', '552', [2 3], [1 2 3 4 5])
%
%     Would put values 1,2,3,4,5 into cells C2,D2,E2,F2,G2, respectively
%
% USING RunOnce().
% Before using this code, you've got to enable the Drive/Sheets APIs via:
%     https://console.developers.google.com/
% Here, you will "create credentials" via an OAuth 2.0 client ID that comes
% with a Client ID and Client Secret. These codes are the two arguments of
% RunOnce(client_id, client_secret). Run RunOnce with both of these
% codes passed as strings, and follow the instructions.
%
% The following code is inspired by, and makes use of, code
% originally published in the file exchange by Claudiu Giurumescu.
% (https://www.mathworks.com/matlabcentral/fileexchange/31221-matlab-to-google-spreadsheets)
%
% I wrote this to accommodate for latest changes in Google API, added some
% comments, and simplified it all so that it can be implemented by the
% average user (hopefully!)
%
% Additionally, I use loadjson by Qianqian Fang to read the input streams
% from Google for learning meta data about the sheets
% (https://www.mathworks.com/matlabcentral/fileexchange/33381-jsonlab--a-toolbox-to-encode-decode-
% json-files)
%
% andrew robert bogaard 26 sept 2016
% abogaard@uw.edu
```

## HOW TO SET UP GOOGLE DRIVE API CREDENTIALS

Open <https://console.developers.google.com/>



API API Manager

Dashboard

Library

Credentials

Library

Google APIs

Search all 100+ APIs

Popular APIs

Google Cloud APIs

Compute Engine API

BigQuery API

Cloud Storage Service

Cloud Datastore API

Cloud Deployment Manager API

Cloud DNS API

More

Mobile APIs

Google Cloud Messaging

Google Play Game Services

Google Play Developer API

Google Places API for Android

Advertising APIs

AdSense Management API

DCM/DFA Reporting And Trafficking API

Ad Exchange Seller API

Ad Exchange Buyer API

DoubleClick Search API

DoubleClick Bid Manager API

Google Maps APIs

Google Maps Android API

Google Maps SDK for iOS

Google Maps JavaScript API

Google Places API for Android

Google Places API for iOS

Google Maps Roads API

More

Social APIs

Google+ API

Blogger API

Google+ Pages API

Google+ Domains API

Other popular APIs

Analytics API

Translate API

Custom Search API

URL Shortener API

PageSpeed Insights API

Fusion Tables API

Web Fonts Developer API

Google Apps APIs

Drive API

Calendar API

Gmail API

Sheets API

Google Apps Marketplace SDK

Admin SDK

More

YouTube APIs

YouTube Data API

YouTube Analytics API

YouTube Reporting API

Google APIs

abogaardmatlab

Search

Info

Help

Notifications

More

Profile

API API Manager

Dashboard

Library

Credentials

Google Drive API

ENABLE

About this API

Documentation

Try this API in APIs Explorer

The Google Drive API allows clients to access resources from Google Drive.

Using credentials with this API

Accessing user data with OAuth 2.0

You can access user data with this API. On the Credentials page, create an OAuth 2.0 client ID. A client ID requests user consent so that your app can access user data. Include that client ID when making your API call to Google. [Learn more](#)

2

CAUTION

Your app

User consent

User data

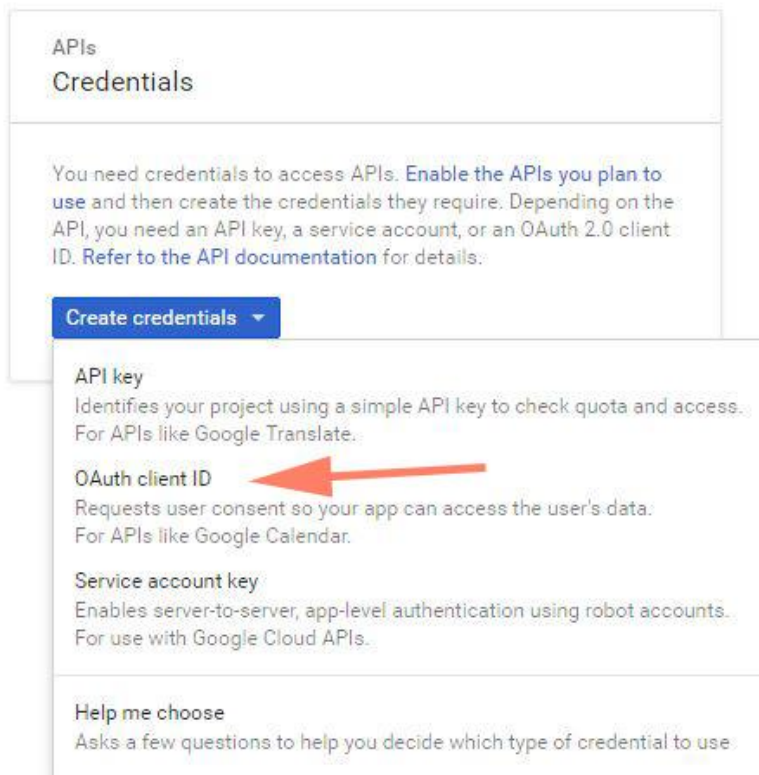
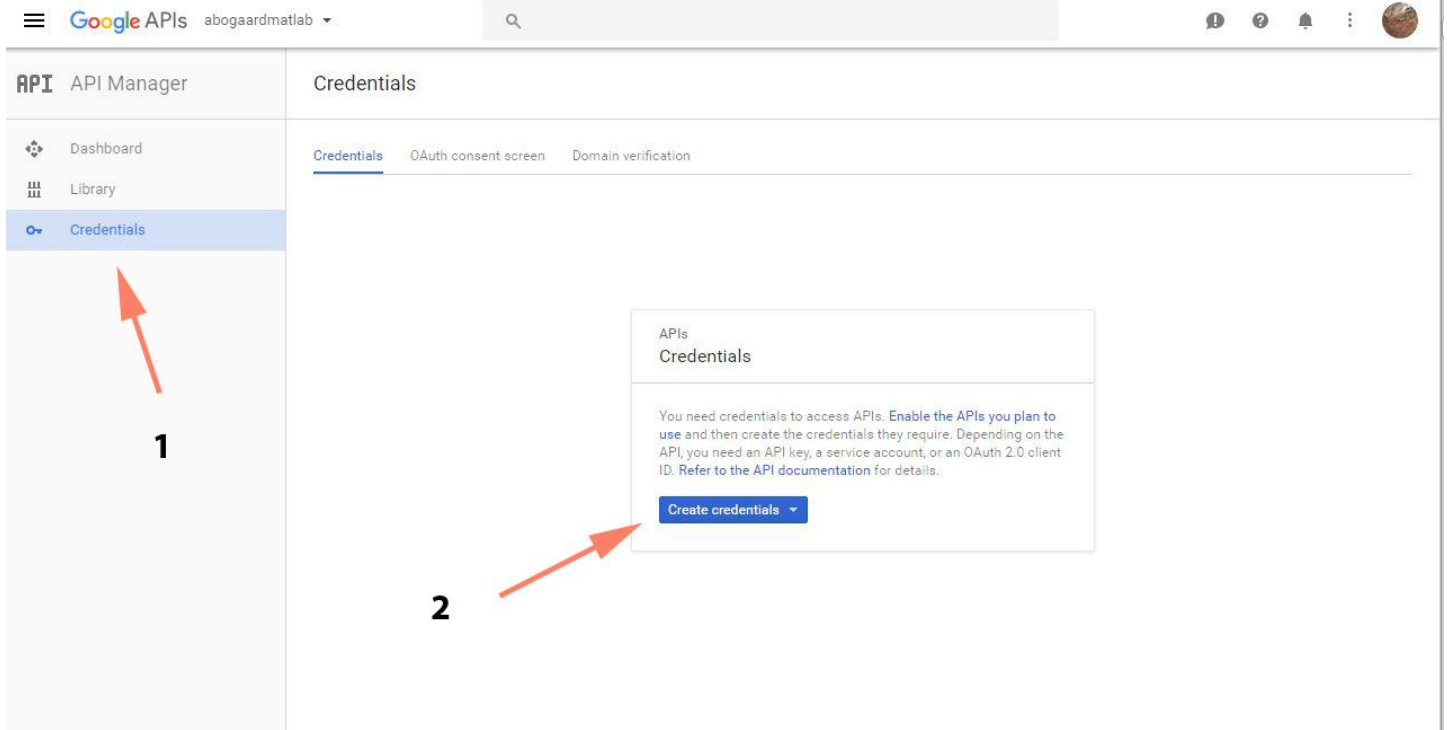
Server-to-server interaction

You can use this API to perform server-to-server interaction, for example between a web application and a Google service. You'll need a service account, which enables app-level authentication. You'll also need a service account key, which is used to authorize your API call to Google. [Learn more](#)

Your service

Authorization

Google service



API Manager

Dashboard

Library

Credentials

←

Create client ID

Application type

☐ Web application

☐ Android [Learn more](#)

☐ Chrome App [Learn more](#)

☐ iOS [Learn more](#)

☐ PlayStation 4

☒ Other

Name

matlab

Restrictions

Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (http://\*.example.com) or a path (http://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

http://www.example.com

Authorized redirect URIs

For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

http://example.com/oauth2callback

Create

Cancel

API Manager

Dashboard

Library

Credentials

Credentials

Create credentials

OAuth client

Here is your client ID

' = client\_id

Here is your client secret

' = client\_secret

OK

OAuth 2.0 client IDs

Name	Creation date	Type	Client ID
matlab	Sep 29, 2016	Web application	