

Google Drive Downloader relies on this google drive API

<https://github.com/Elringus/UnityGoogleDrive.git#package>

First, get yourself set up with this package, following Elringus' instructions

**This package does not contain a way to save data from your unity scene!
You must save files to the correct asset path separately from this package**

This package contains two relevant files for its functionality, Uploader.cs and Downloader.cs

Uploader.cs is a unity monobehaviour script that takes files from the local asset path, and uploads them to specific google drive folder ids

Downloader.cs is an editor window, that takes a csv of google drive ids, and downloads those items into local folders

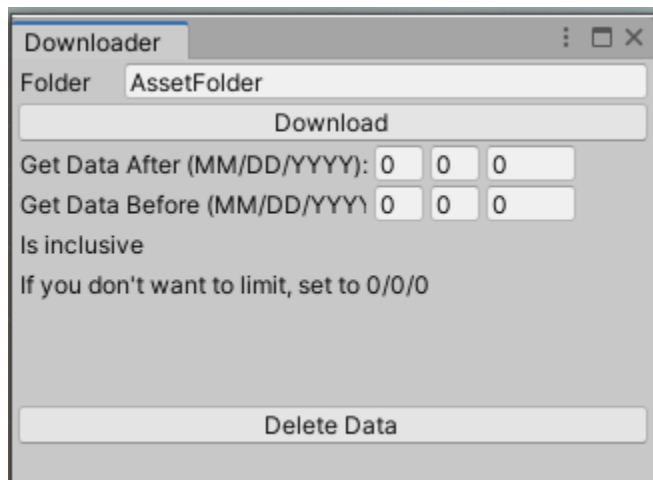
You should never have to open this file and program in it, however if you are comfortable and want to add more features go ahead, you cannot, however sell it

(can not download google drive types, only items uploaded to google drive)

Editor:

Downloader is an editor window, and all functionality can be accessed through just the window. You can open this through the top bar Tools>Downloader. You can move around and dock this window like all others

*This is what it should look like when first added



Folder:

This is the local folder where the PARENT.csv is held, and where all the files will be downloaded to.

Find your data path after the Application.dataPath, for example for
<GameName>\Assets\<DownloadFolder>, Local Path would equal
“<DownloadFolder>”

Download:

Use this button to download all relevant data. Uses the Ids found in PARENT.csv and downloads all items in the date range

Date Range:

These are inclusive ranges for dates of data you want to download. If you want to download all, leave them at 0/0/0000.

If you would like to download after, use the top options, before, the bottom options

Delete Data:

Deletes all folders in local folder

Use before pushing if using git to not clog a repo.

Parent.CSV

The downloaders uses a file called PARENT.csv to correctly find the files to download.

Make your own:

Open a new google sheet.

Go to Extensions > AppsScripts and replace the default function with:

```
function myFunction123() {  
  var foldername = ''; //Google drive folder name  
  var folderlisting = 'listing of folder ' + foldername;  
  
  var folders = DriveApp.getFoldersByName(foldername)  
  var folder = folders.next();  
  var levelFolders = folder.getFolders()
```

```

//var contents = folder.GetFiles();

//var ss = SpreadsheetApp.create('listing of folder ' + foldername + );
//var sheet = ss.getActiveSheet();
var ss = SpreadsheetApp.openById("ID of this sheet")
var sheet = ss.getActiveSheet();
sheet.clear();
sheet.appendRow(['level', 'date', 'name', 'id']);

while(levelFolders.hasNext())
{
    var newFolder = levelFolders.next();
    var contents = newFolder.GetFiles();
    var file;
    var level = newFolder.getName();
    var id;
    var date;
    var name;
    while(contents.hasNext()){
        file = contents.next();
        date = Utilities.formatDate(file.getDateCreated(), "EST", "MM/dd/yyyy");
        id = file.getId();
        name = file.getName();
        sheet.appendRow([level, date, name, id]);
    }
}

};

```

Replace “foldername” with the name of the parent folder. Currently there needs to be one level of folder between this one and the data to download.

Suggested format is

MyUnityFolder

 NewSheet (PARENT.csv)

 FolderWithData1

 FolderWithData2

And foldername would equal “MyUnityFolder”

Replace ID of this sheet in `var ss = SpreadsheetApp.openById("ID of this sheet")` (line 12) with the ID of the sheet that you want to use as your parent.

For this sheet:

<https://docs.google.com/spreadsheets/d/11r9ajRsWFQ63-F1Lng9UVmxeKNOc03kQGDQO9vtMX70>

The highlighted section is the id

Run the script

When running you will see “Notice Execution started” in the execution log at the bottom of the screen. This may take a while, but when completed a new notice will show at the bottom of the screen.

Once this is done running, download the sheet as a csv, name it PARENT and place it in the folder you will be using as “FOLDER” inside downloader