

this->Article

itsjustmycode.com

Creating a Google Share and Export App

Share and export your Google Docs, Sheets and Slides to all file types from your website while keeping your file ID private and not bothering with Google Drive API

Jehan Semper, 3/15/2021

Table of Contents

1. Creating a Google Share and Export App
 - a. What Is the Google Standard Export Feature?
2. Making It Work
3. What You Need
 - a. Gathering the Google File ID
 - b. Google Docs, Sheets and Slides Editor
 - c. Google Drive
4. Creating Your Workflow
 - a. Creating the TSV Database
 - b. Parsing the Database
 - c. Constructing Valid Export URLs
5. Writing Working Code
6. Everything You'd Like to Know Concerning this->Article

Creating a Google Share and Export App

You have a website. You would like to share a bunch of Google Docs, Sheets and Slides with web users, but you don't want to build a tool with Google Drive API, you don't want to do the major nerd walk of Google's OAuth 2.0 token odyssey, you don't want to manage editing permissions and you don't want to reveal the file ID of your files. You can accomplish this goal with Google's standard export feature and **this->Article**.

What Is the Google Standard Export Feature?

The Google standard export feature allows you to export Google Docs, Sheets and Slides to many popular file types including PDF, plain text, comma and tab separated values aka CSV and TSV and more. The standard export feature is available with the Google editor for Docs, Sheets and Slides by clicking **File > Download** and choosing the file type you'd like to have.

Within **this->Article**, we will create a Google share and export app which creates PDF, plain text, XLSX, CSV and TSV versions of your chosen Google Docs, Sheets and Slides.

Making It Work

Google's standard export feature utilizes constructed URLs with pertinent query strings to export Google Docs, Sheets and Slides to your desired file type. To create a Google share and export app, we will construct valid export URLs programmatically. You can programmatically construct valid export URLs with

practically all languages, including Perl, [PHP](#), Python, Java, JavaScript, C# and, of course, GoLang. Within **this->Article**, we will complete implementation with standard [PHP](#) on [Linux](#).

What You Need

To complete this implementation, you need standard [PHP](#) skills, you need to know the difference between a single quote and a backtick, you need to know how to parse a CSV or TSV file and you need [Linux](#). And, when you have all that and before you do anything at all, you will start with:

Gathering the Google File ID

Choose the Google Docs, Sheets and Slides you would like to share within your share and export app. You can share a specific file from within the Google Docs, Sheets or Slides editor. You can share multiple files while browsing within Google Drive. When you have chosen the files you would like to share, save the file IDs by copying and pasting the link to the file within a text file. You will utilize the text file with the file links and file IDs to create a TSV database and programmatically construct valid export URLs.

Google Docs, Sheets and Slides Editor

While viewing and editing the specific file you would like to share, confirm the settings which make the file available for viewing, download, print and copy are active.

- 1. Allow Users to Download, Print and Copy**

Click *Share > Settings Glyph > Viewers and commenters can see the option to download, print and copy*

- 2. Allow Anyone with the Link to View**

Click *Share > Get link > Change to anyone with the link*

IMPORTANT: Confirm the share settings for your Google Doc, Sheet or Slide include: **Anyone on the internet with this link can view** and **Viewer** is the chosen role.

- 3. Click "Copy Link" and paste the link into a text file.**

Google Drive

Search or navigate within Google Drive to the list of specific files you would like to share. While viewing the list of specific files you would like to share, select the files, then choose and/or confirm the settings which make the files available for viewing, download, print and copy are active.

- 1. Allow Users to Download, Print and Copy**

While the files are selected and highlighted:

Right Click > Share > Settings Glyph > Viewers and commenters can see the option to download, print and copy

2. Allow Anyone with the Link to View

Click *Share* > *Get link* > *Change to anyone with the link*

IMPORTANT: Confirm the share settings for your Google Doc, Sheet or Slide include: **Anyone on the internet with this link can view** and **Viewer** is the chosen role.

3. Click “Copy Link” and paste the link into a text file.

Success! When you have completed these steps with success, you will have a text file with a list of Google Docs, Sheets and Slides file links with the file ID for each file which will look sort of like this:

```
/path/to/file.txt:  
https://docs.google.com/document/d/<FILEID>/edit?usp=sharing  
https://docs.google.com/document/d/<FILEID>/edit?usp=sharing  
https://docs.google.com/document/d/<FILEID>/edit?usp=sharing
```

Creating Your Workflow

Within **this->Article**, to complete implementation, you will create three files which will act as your database, file catalog and download mechanism.

You will create:

1. /path/to/exportdocs.tsv TSV Database
2. /path/to/exportdocs.php File Catalog
3. /path/to/download.php Download Mechanism

We will commence our workflow with creating the TSV database.

Creating the TSV Database

With your text file of file links with file IDs, you can build a database of files you would like to share. To create a database with your text file, edit the text file to include additional data about the Google Docs, Sheets and/or Slides files.

Additional data you'd probably like to include is: the title, an abstract aka descriptive data regarding the file and more. You can create a comma separated values file aka CSV database or a tab separated values file aka TSV database. Within **this->Article**, we will create a TSV database with tab separated values which includes the link to the file, a title and an abstract for each file.

When you have created the TSV database with success, you will have a database that looks sort of like this:

```
/path/to/file.txt:  
https://docs.google.com/document/d/<FILEID>/edit<tab>Document<tab>Google Doc  
with data that will save the world
```

Parsing the Database

Within **this->Article**, we will complete implementation by parsing the TSV database with standard [PHP](#). To complete this task, your [PHP](#) must complete each of these tasks within this sequence and with success.

1. `$file = file_get_contents(/path/to/database.tsv);`
2. `$db = preg_split('/\n\r/', $file);`
3. `foreach ($db as $var => $item) { $itemData = preg_split('/\t/', $item); }`

Success! When you have completed parsing the TSV Database with success, you will have an array called `$itemData` with the link to the file, the title and the abstract.

Constructing Valid Export URLs

Google Docs, Sheets and Slides have export URLs that are specific to each document and file type. You will utilize the valid export URL structure to create a widget which you can customize programmatically to export the files you would like to share with the desired file type.

Within **this->Article**, to construct a valid export URL, you will need two arrays.

1. \$fileTypes

Nested array with the document type and desired export file types.

```
$data =  
    array(  
        'document' => array(  
            'pdf' => 'PDF',  
            'txt' => 'Plain Text'),  
        'presentation' => array(  
            'pptx' => 'Powerpoint',  
            'txt' => 'Plain Text', 'pdf' =>  
            'PDF'),  
        'spreadsheets' => array(  
            'pdf' => 'PDF',  
            'xlsx' => 'Microsoft Excel',  
            'csv' => 'Comma Separated Values',  
            'tsv' => 'Tab Separated Values'),  
    );
```

2. \$exporturl

Array with document type and valid export URL widget.

Google Docs

```
https://docs.google.com/feeds/download/documents/export/Export?id=__FILEID__&exportFormat=__FILETYPE__
```

Google Sheets

```
https://docs.google.com/spreadsheets/export?id=__FILEID__&exportFormat=__FILETYPE__',
```

Google Slides

```
https://docs.google.com/feeds/download/presentations/Export?id=__FILEID__&exportFormat=__FILETYPE__
```

Success! When you have created \$fileTypes and \$exporturl, you will have the data to construct a valid export URL.

Writing Working Code

Now that you have \$fileTypes and \$exporturl, you can combine them to create a valid export URL or *you can proceed to **Actual Working Code** and copy and paste the code we have created to accomplish this task to your Linux server.*

Within **this->Article**, we will combine the document title, the abstract, \$fileTypes and \$exporturl to create a widget/share item which provides web users with access to the documents you have chosen to share while keeping the file ID of each file private.

To complete this task with success, you need to utilize your standard [PHP](#) skills to write code that says something like this:

```
foreach ( $database as $fileID ) {
    $count = 1;
    foreach ( $fileTypes[$fileType] as $var => $value ) {
        $exporturlwidget = $exporturl[$fileType];
        $exporturlwidget = preg_replace('/__FILEID__/', $fileID, $exporturlwidget);
        $exporturlwidget = preg_replace('/__FILETYPE__/', $var, $exporturlwidget);
        $urlwidget[$count] = '<A LINK THAT DOES NOT REVEAL THE FILE ID>';
        /* write code to:
            + associate $urlwidget with the real link to the file
            + add the title, abstract and $urlwidget to the widget/share item
            + tell your code how to request the real file ID with the alias link
            + commence download of the exported file type
        */
        $count++;
    }
}
/* write code to:
    + actually list the files available for export and download
*/
```

Success! When you have completed this task with success, you will have a widget/share item which provides web users with access to the documents you have chosen to share while keeping the file ID of each file private.

Writing this code yourself would require 3 to 6 hours of development time *or you can utilize and adapt the **Actual Working Code** we have created to accomplish this task.*

Either way, you will have a Google export app that looks sort of like this:

Share and Export your Google Docs, Sheets and Slides

while keeping your file ID private and not bothering with Google Drive API

1. File: **Document**

Download: [pdf](#) [txt](#)

Abstract: Google Document you can export to PDF and TXT

2. File: **SpreadSheet**

Download: [pdf](#) [xlsx](#) [csv](#) [tsv](#)

Abstract: Google Spreadsheet you can export to PDF, XLSX, CSV and TSV

3. File: **Presentation**

Download: [pptx](#) [txt](#) [pdf](#)

Abstract: Google Slide you can export to PDF

JEHAN SEMPER CONSULTING

2021

Continue reading to
copy and paste
majorly small text of
Actual Working Code

Enjoy

Everything You'd Like to Know Concerning this->Article

Title: Creating a Google Share and Export App: Share and export your Google Docs, Sheets and Slides to all file types from your website while keeping your file ID private and not bothering with Google Drive API

Url: <https://drive.google.com/file/d/1yRKcvSz7osBEwamAQh5XLLoDwUPlz4vn/view>

DateCreated: 03/15/2021

ArticleSection: Google Tools

Author: Jehan Semper

AuthorUrl: <http://itsjustmycode.com>

Organization: itsjustmycode.com

Description: Within this->Article, we will create a Google share and export app which creates PDF, plain text, XLSX, CSV and TSV versions of your chosen Google Docs, Sheets and Slides from your website while keeping your file ID private and not bothering with Google Drive API.

CodeSampleType: Scripts

ProgrammingLanguage: PHP

ProficiencyLevel: Intermediate

InLanguage: En

Dependencies: PHP, Linux, Google files

ArticleType: TechArticle