

Reports upload to Google.Drive v0.1.

- Does:
- Results:
- Prerequisites:
- Commits:
- Location:
- Uses:
- Options:
- CRONJOB & example of usage:
- TODO & Remarks:
- Security notice:
 - Google API Roles:
 - Restrictions and Overrides:

Does:

*Uploads Reports as CSV outputs generated from SQL query to Google.Drive's folder accessible by given users, whose ID indicated as their emails.
The script can be used for any such upload.*

Results:

See [this Google.Drive folder](#).

Prerequisites:

```
pip install --upgrade google-api-python-client oauth2client
pip --trusted-host pypi.org --trusted-host files.pythonhosted.org install mysql
pip --trusted-host pypi.org --trusted-host files.pythonhosted.org install mysqlclient
pip install --upgrade --force-reinstall --trusted-host pypi.org --trusted-host files.pythonhosted.org mysql-python
DBsql.py
    stored in the same folder
config.py
    stored in the same folder, contains configuration to connect to the DB and necessary paths
```

Commits:

On [bitbucket](#), see there DBsql.py, config.py and the GoogleDriveUpload.py

Location:

/axpm/data-mine/GoogleDriveUpload.py
JSON files (see below [json_file](#)) with IDs and user emails are stored in folder (/axpm/data-mine/jsons)

Uses:

/axpm/data-mine/service.json

credentials to interact with Google.Drive API: they belong to [Google service account](#) created under axpm.com domain [fileshare@reports-upload-to-gdrive.iam.gserviceaccount.com](#) *) within a project «Reports Upload to GDrive»

/axpm/data-mine/jsons/unrealized_invoices.json

in order to manage them for different type of Reports (SQLs), all uploads are stored in the following format:

```
{
  "default_users":
  [list of user emails who can
  access reports],
  "files": [
    {

      "google_source_id":
      "1kSk07TwdbvAlQ6DYGtX-
      1Axt1ln5delH",

      "google_source_parent_id":
      "0A0m9IAvcMICxUk9PVA", //
      this is Root folder ID, see
      output of the script with no
      options

      "name":
      "Auto_Generated_Reports",
      "source_type":
      "folder",
    },
    {

      "google_source_id":
      "1DCrE6Kr46OygcC3WRmV6TsX80SC
      mDtVx",

      "google_source_parent_id":
      "1kSk07TwdbvAlQ6DYGtX-
      1Axt1ln5delH", // this is a
      folder ID where the files
      are put

      "name":
      "Unrealized_invoices_8-14-
      2018.csv",
      "source_type":
      "file",
    }
  ]
}
```

*) However, when emails notification about an upload is sent to the users, they see the notice «fileshare@reports-upload-to-gdrive.iam.gserviceaccount.com is outside your organization» —

presumably it's because this service account, although created within axpm.com domain, needs to be accepted by the domain administrator. Virtually, any Google user can create and send files by means of the Google Drive API to any email.

Options:

arguments:	if necessary:	does:
------------	---------------	-------

--json [json file]	yes for uploads & folder creations	json_file with default_users and list of uploaded files with their google_source_ids and names; see /axpm/data-mine/jsons/uploads_and_permissions.json
--del [files separated by commas]	yes for deletions	to delete uploaded files permanently, by their google_source_id (see json_file)
--folder [folder name]	yes for folder creation	to create a folder with a given name to upload files which are due to be accessible from this folder
--upload [sql script location]	yes for uploads	to upload a CSV generated by a given SQL code to the folder indicated in --parent option
--ppl [emails separated by commas]	yes for updates of default_users only	to update default_users (see json_file), allowed to read the uploaded reports with the google_source_id indicated in --report option
--report [google_source_id]	yes for updates of default_users only	any google_source_id of the uploaded files to update default_users specified in --ppl option: after this the users can see this file and all others uploaded later on
--parent [google_source_id]	yes for uploads	google_source_id of a parent folder where the file is to be uploaded, if not specified, files are stored attached to the Root folder ID and this means they are not accessible from the folder (see below in TODO & Remarks) but scattered by/over different IDs
--help	no	outputs the below information, see the next row and the usage options
no options		outputs the info about its service account data, like: Current user name: fileshare@reports-upload-to-gdrive.iam.gserviceaccount.com Root folder ID: 0AOm9IAvcMICxUk9PVA Total quota (bytes): 16106127360 Used quota (bytes): 11393849

CRONJOB & example of usage:

to run it every Monday:

```
00 01 * * 1 python /axpm/data-mine/GoogleDriveUpload.py --parent
1tX0vZ9RbdU26fd2_xcoCb-4yYi-0fTDm --upload /axpm/data-mine/sql
/unrealized_invoices.sql --json /axpm/data-mine/jsons/unrealized_invoices.
json >> /var/log/unrealized_invoices_upload.log
10 01 * * 1 python /axpm/data-mine/GoogleDriveUpload.py --parent
15VSdBevt5FCbf7pZJgoq2i4GHLpdudTs --upload /axpm/data-mine/sql
/active_payment_plans.sql --json /axpm/data-mine/jsons
/active_payment_plans.json >> /var/log/active_payment_plans.log
```

TODO & Remarks:

Permissions for file views are not stored but can be removed, if it's necessary, by their permission_id which can be obtained by the corresponding file_id (i.e., google_source_id — see above).

NB: *permissions in Google.Drive API are associated with resources*, but not with the people who got access to a given resource. Therefore, once a permission to a given file is removed — all group of people, who was granted the access to the file, won't be able to see it anymore.

Currently files can be deleted without a prompt by their google_source_id stored in /axpm/data-mine/uploads_and_permissions.json — see above in «Uses» its format.

Folder links are formatted this way: https://drive.google.com/drive/u/0/folders/google_source_id.

File links: https://drive.google.com/file/d/google_source_id

NB: Google notifier by email currently *sends the above links like HTTP-links disrespecting SSL*.

Security notice:

Google API Roles:

Restrictions and Overrides: