The `gspread` Package

The `gspread` package provides an interface to Google Sheet documents via the Google Sheets and Google Drive APIs.

Reference

Google APIs:

- https://github.com/googleapis/google-api-python-client
- https://developers.google.com/api-client-library/python/
- https://developers.google.com/sheets/api/guides/authorizing

The `gspread` Package:

- https://github.com/burnash/gspread
- https://www.twilio.com/blog/2017/02/an-easy-way-to-read-and-write-to-a-google-spreadsheet-in-python.html
- https://gspread.readthedocs.io/en/latest
- https://gspread.readthedocs.io/en/latest/oauth2.html

Installation

First install the package (and a dependent auth-related package) using Pip, if necessary:

```
pip install gspread oauth2client
```

Setup

Downloading API Credentials

Visit the Google Developer Console. Create a new project, or select an existing one. Click on your project, then from the project page, search for the "Google Sheets API" and enable it. Also search for the "Google Drive API" and enable it.

From either API page, or from the API Credentials page, follow a process to create and download credentials to use the APIs. Fill in the form to find out what kind of credentials:

- API: "Google Sheets API"
- Calling From: "Web Server"
- Accessing: "Application Data"
- Using Engines: "No"

The suggested credentials will be for a service account. Follow the prompt to create a new service account with a role of: "Project" > "Editor", and create credentials for that service account. Download the resulting .json file and store it in your project repo in a location like "auth/google_api_credentials.json".

Before committing, add the credentials filepath to your ".gitignore" file to ensure it does not get tracked in version control or uploaded to GitHub:

```
# .gitignore

auth/google_api_credentials.json
```

Configuring Spreadsheet Document

Use this example Google Sheet, or create your own. Note the document's unique identifier (e.g. `1_hisQ9kNjmc-cafIasMue6IQG-ql_6TcqFGpVNOkUSE`) from its URL, and store the identifier in an environment variable called `GOOGLE_SHEET_ID`.

If you create your own, make sure it contains a sheet called "Products" with column headers `id`, `name`, `department`, `price`, and `availability_date`. And modify the document's sharing settings to grant "edit" privileges to the "client email" address located in the credentials file.

Usage

```
from dotenv import load_dotenv
import os

import gspread
from oauth2client.service_account import ServiceAccountCredentials

load_dotenv()

DOCUMENT_ID = os.environ.get("GOOGLE_SHEET_ID", "OOPS")
SHEET_NAME = os.environ.get("SHEET_NAME", "Products")

#
# AUTHORIZATION
#

CREDENTIALS_FILEPATH = os.path.join(os.path.dirname(__file__), "..", "auth",
"spreadsheet_credentials.json")

AUTH_SCOPE = [
    "https://www.googleapis.com/auth/spreadsheets", #> Allows read/write
access to the user's sheets and their properties.
    "https://www.googleapis.com/auth/drive.file" #> Per-file access to files
created or opened by the app.
]

credentials =
ServiceAccountCredentials.from_json_keyfile_name(CREDENTIALS_FILEPATH,
AUTH_SCOPE)

#
# READ SHEET VALUES
#

client = gspread.authorize(credentials) #> <class 'gspread.client.Client'>

doc = client.open_by_key(DOCUMENT_ID) #> <class 'gspread.models.Spreadsheet'>

print("-----------------")
print("SPREADSHEET:", doc.title)
print("-----------------")

sheet = doc.worksheet(SHEET_NAME) #> <class 'gspread.models.Worksheet'>

rows = sheet.get_all_records() #> <class 'list'>

for row in rows:
    print(row) #> <class 'dict'>

#
# WRITE VALUES TO SHEET
#

next_id = len(rows) + 1 # TODO: should change this to be one greater than the
current maximum id value

next_object = {
    "id": next_id,
    "name": f"Product {next_id}",
    "department": "snacks",
    "price": 4.99,
    "availability_date": "2019-01-01"
}

next_row = list(next_object.values()) #> [13, 'Product 13', 'snacks', 4.99,
'2019-01-01']

next_row_number = len(rows) + 2 # number of records, plus a header row, plus
one

response = sheet.insert_row(next_row, next_row_number)

print("-----------------")
print("NEW RECORD:")
print(next_row)
print("-----------------")
print("RESPONSE:)
print(type(response)) #> dict
print(response) #> {'spreadsheetId': '___', 'updatedRange': '___',
'updatedRows': 1, 'updatedColumns': 5, 'updatedCells': 5}
```