

Sheetgo Back-end Test

This test aims to challenge you to create an API and a client for it. The API must provide endpoints for the extraction of basic data from an Excel file (.xlsx) and, also, a simple image conversion route. Regarding the client, it will have to allow the evaluators to test the API in a simple way.

General information

The Project

- MUST be hosted within a private Github repository and shared with **@mau21mau** and **@klassmann**. Github private repositories are now free, so it should not be an issue.
- MUST contain a **README.md** with the details of the project installation and execution.
- MUST be developed using **Python >= 3.6** and **Flask**

Endpoints

- MUST accept only HTTP request using POST.
- MUST validate the token according to the parameters described at [Authentication using JWT at the API](#).

Details

Authentication using JWT at the API

- The JWT encryption key must be the following:
 - **z6Ct_d2Wy0ZcZZVUYD3beI5ZCSsFrR6-f3ZDyn_MW00**
- The token must be sent with the header:
 - **Authorization: Bearer <token>**.
- The data sent with the token must follow the following structure:
 - **{"email": "email@company.com"}**
- The token email must be one of the following:
 - ["lucas@sheetgo.com", "mauricio@sheetgo.com", "rafael@sheetgo.com"]

Endpoints

Excel: `/excel/info/`

- Create an endpoint to extract information from an Excel file (.xlsx).
 - **Request:** must send a binary xlsx file via HTTP POST.
 - **Response:** must return a JSON with the list of the tabs from the excel file, ordered alphabetically.

Image: `/image/convert/`

- Create an endpoint that will convert the format of an image.
 - **Request:** must receive an HTTP POST containing the file and the parameter “**format**” containing one of the following values: “png” or “jpeg”. These values will represent the output format of the image.
 - **Response:** The converted image file.

Bonus (optional)

In case you want to go beyond the scope provided above and the test was too easy :D

Endpoint for Dropbox: `/image/convert/fromdropbox/`

- Try to implement an additional endpoint for converting a given image, but instead of receiving the image file directly on the request, it would receive the image PATH of an image stored at Dropbox along with the desired format to be converted. Then, the response would be the converted image. The dropbox *access_token* should be sent with the JWT key to the API so the Dropbox API can be accessed.
- Add also unit tests for the application

Testing the application

Must implement a way for us to test the API. A few suggestions:

- Test requesting the API through CURL or WGET
- Bash script(s) that receives the parameters via terminal and bypass them to the API.
- Scripts in Python for each endpoint. Ex: `convert_img.py`, `excel_info.py`, etc.

In case you implement a static script for the test, like in the example **`excel_info.py`**, the parameters for the request may be provided statically into the script with the exception of the JWT token. The JWT token MUST be generated from the given JSON according to the section [Authentication using JWT at the API](#).

Here is an example of a request using curl for testing the API for retrieving the excel file tabs. In this example, the JWT token is generated on the fly:

```
curl http://127.0.0.1:5000/excel/info -H 'X-Authentication-Token: $(python generate_jwt.py)' -F file=@Sample.xlsx
```

Tips

Here there is some additional information that might help you with the project.

- Libraries that might be used for the tasks: **`pyjwt`**, **`dropbox`**, **`openpyxl`**, **`Pillow`**

Questions

Any questions you have regarding the project you can reply us via e-mail.