

Requires:

Python 3.5 (Make sure environment variable for python3.5 is set, especially if there is another older python on system as well)

Django	pip install django
Djangorestframework	pip install djangorestframework
ImgurPython	pip install imgurpython

Optional: Use virtual environment or condaenv (requires Anaconda distribution of python)

Register an Imgur application:

1. Register an Imgur account (refer to Imgur instructions)
2. Under your account settings (hover over top right of page with your account name, select “settings”), select applications (second from bottom option on left navigation menu)
3. Click [create your own!](#)
4. Under the left menu [general information](#), select [register an application](#). Then select [register their application](#).
5. Fill in the form (use any url for **Authorization callback URL**) and submit it.
6. In the new screen, there should be a **Client ID** and **Client secret**. Keep these values somewhere safe; they will be used in next step.
7. Under `imagesite` folder (the main project directory), create a new text file called `keys-imgur.txt`. Fill the text file with your **Client ID** and **Client secret** in as follows:
(Do not add extra spaces)

```
client_id:client-ID
client_secret:client-secret
```

You do not need to do anything further with the Imgur API; the program handles the rest of it.

Installation:

- To test on local server, navigate to `imagesite` folder then run `python3 manage.py runserver` from command line.
- To install on other server, first make sure the above requirements are installed on the system. Then copy the entire `imagesite` folder to server destination and refer to server host's instructions.

How to add an album:

1. Start an album in Imgur (if this is your first time using Imgur, you will need to register an account) and upload all the photos there. Copy the album id (last section of url, ex from <http://imgur.com/a/AbCdE> select 'AbCdE').
2. In project directory, open `set_album_values.py`, and set the 'album_id' to your album id. Optional: Give a name to the album and set the album icon ('init_message') to a web link to an image. Do not modify any other variables.
3. From project directory, run `python3 album.py add`; this will add the current album from `set_album_values.py` to the database.

How to modify an album (Note: can only add images, or modify 'name' or 'init_message' fields):

1. If adding/removing images, add/remove images from imgur album first.
2. Open `set_album_values.py`; set 'album_id' to the album id. **Set any variables you do not want to modify to `None`; if you do not wish to modify variables, do not leave these as any strings or else they will be modified to those strings.** To remove a variable, set it to an empty string. To add/remove images, no additional changes are needed; the program will auto detect imgur images changed.
3. From project directory, run `python3 album.py modify`; this will modify the current album from `set_album_values.py` and apply changes to database. If there are existing ratings from removed images still in db, these will not be deleted.
4. There is no way to perform a modification where images are deleted from an album in which either i. all collected ratings are reset, or ii. ratings corresponding to deleted images are deleted. The only way to do so would be to start a new imgur album with the same images (except for ones to be deleted), then add it to the database. (Refer to instructions on how to add album)

How to obtain all current rating data:

1. Run `python3 album.py collect album_id` where `album_id` is the album you want to collect statistics about.
2. The output will be in a `album_id_data.csv` file in the `data/` folder. For each image in the album, it will list the image id, number of ratings one, two, ... ,nine, ten in sequential order, separated by a comma, as well as the total number of ratings at the end. (The image ids correspond to the hash at the end of the url of the image, which can be viewed on imgur album website).

How to delete an album:

1. Open `userdata.py` and add the album id to the `filter_list`. This will prevent the album from showing up in mainpage and ignore any new ratings it receives, but it will not delete any old data.
2. There is no way to perform a “true reset” where both album images and old rating data is deleted. The only way to do so would be to start a new imgur album with the same images, then add it to the database. (Refer to instructions on how to add album)

There are extra copies of `album.py`, `userdata.py`, and `set_album_values.py` in default state under `backup/` folder in project directory.