# Few Notes

1. I used Python 3.6 and 3.10 to test the project. The file requirements.txt made with Python 3.10.7.
2. The project uses Flask and SQLAlchemy.
3. Although this is a small project, I used flask blueprint like I do in real project.
4. By using SQLAlchemy config, there is no difference in the code if you use SQLite of MySql. Based on that, I used SQLite to test my project. To work with real remote DB, you can edit the database section in *settings.ini* and use ProductionConfig instead of DevelopmentConfig in the file *app.py*.
5. Most of the files are common to both processes - background service and web server. Because of that I use one file tree.
6. I used *multiprocessing.connection* for internal connection, but it can be changed to any other utility.
7. To execute the background service run the command: *python ./main\_service.py*
8. To execute the web server run the command: *python ./main\_web\_server.py*
9. I the file settings.ini you can set the log level and the to
10. I made two docker files (Dockerfile\_bg, Dockerfile\_web) to create docker images for the background service and for the web server. (didn’t get the ability to build the images on my laptop, so I don’t sure the files are correct).
11. When working with docker containers, you should edit the hostname (under Service) in *settings.ini* with the background service IP address.
12. For client test, please change the filename *client\_test.bat.txt* to *client\_test.bat* (I changed it because of security problem. Google block ZIP file which contains file with BAT extension).
13. Some of the projects had invalid requirements.txt (with invalid characters). In this case *githubAPI2.py* got exception and I set the project status to an error.
14. I add a limit for the number of subprocesses the background service can run at a time. You can easily change it (*main\_service.py*).
15. In case the background service stopped, when it run again, it continues to handle the open requests and projects without *requirements.txt*.
16. When I tested it to much, GitHub blocked my IP address, and *githubAPI2.py* didn’t reply as expected. After changing the IP address (or after a period of time), the background service continues to handle the request and projects.