(Email based computer task management system)

Ishan Shanware

January 2020

1 Introduction (Problem Faced and Proposed solution)

A few weeks earlier during the 1st semester, I had an idea of creating a task management system for your computer which performs tasks for you even when you are away. The whole idea was that a person can get to manage his system even when he is not in its vicinity.A few important tasks that such a system could perform were changing file permissions when your are away,updating information or simply putting information into a database(say MySQL) when you send an excel sheet to your computer,performing file uploads when you send a particular files location and performing torrent downloads when you send magnet links to your computer. I discovered that sending instructions to your computer through email was easy to build with certain python libraries and also provided security since the application was programmable in a way,such that your computer only responded to your emails.
RALF has no meaning it's literally a short form for my project.

2 Tech Stack

Pythonic Libraries I am using are:
Imapclient
IMAPClient is an easy-to-use, Pythonic library. IMAPClient is straight forward it use, but it can be useful to have at least a general understanding of the IMAP i.e, it allows the system to manage your email,you can read email data,manage email folders, Delete Emails etc. It Basically uniquely handles each email and ang generates fully parse server responses.

Dropbox API
The Dropbox API v2 is a set of HTTP endpoints that help your app integrate with Dropbox. I used dropbox api to upload files.Dropbox API is very easy to

integrate into APPS and websites.
Why I didn't use Google Drive?
The google drive api has become very difficult to use if you don't intend to build a website or mobile app.Google's new security guidelines require proper details before they release the oauth credentials or access tokens required to build the project.I spent two days trying to get the oAuth client keys for Gdrive ,therefore i started using the dropbox API which i found extremely easy to use and further developed my application upon it.

3 Progress and Completeness

My final goal for this application is to be able develop a complete email based task manager. That manages tasks according to the instructions sent by the computer. For example:There can be sub-categories of tasks such as download,upload,terminal and database management. when one of these categories is mentioned in the subject the program reads further instructions from the body of the email and work accordingly.

However so far, I have only been able to develop a email based torrent downloading system and email instruction based file upload system using dropbox API and I still need to integrate these scripts so that main script reads the subject and perform the task accordingly.

4 Future Implementation

Dropbox API is somewhat is somewhat easy to use, however there is no clarity provided in the documentation to upload large files that exceed 100mb. I intend to build an application which is very similar to this however it performs zipping and unzipping operations based on requests sent from the user email. for eg:if the user demands a large folder to be zipped and uploaded it will do so.

This projects somewhat resembles the working of a server, i.e. it takes the requests from clients and then performs actions based on the user requests. So maybe we can backup your data onto a digital ocean server and you can then access data as per the requests sent to a server. The reason,I emphasize on the word server is because this application obviously cannot work when your computer is in sleep mode,it is supposed to function like a server and therefore i decreased the sleep duration and set sleep to never. I customized a crontab to run the file every 15 minutes. You can do it too just follow the instructions in the readme file on github.

References

[IMAPclient Documentation]: https://imapclient.readthedocs.io

[Dropbox API Docs]: https://www.dropbox.com/developers/documentation/http/documentation