Haro Listener

By: Zane Alumbaugh

Description

Haro Listener is used to scrub Haro daily update emails for the links that contain specific keywords. If it finds any of these links, it will notify the user with an automated email containing all the links that it found. By default, it is set to check for new emails every 30 minutes.

To avoid inbox clutter and improve runtime efficiency, the program makes use of Gmail's label and filter system. It first creates a filter that sends all new emails from Haro to a label with the name "haro_unprocessed." If the label does not exist, the program will automatically create one. Additionally, it creates a label names "haro_processed" to store emails that have already been scrubbed.

Once initialization is complete, every 30 minutes the program will check for any emails in the "haro_unprocessed" label. If it finds any, it will scrub them for links containing the keywords. If it finds any, it will notify the user, otherwise it will wait another 30 minutes and repeat the process.

After an email is processed, it will be archived and added to the label "haro_processed" incase the user needs to refer back to it later.

Installation

System Requirements

- Python 2.7, 3.4 or higher (Python3 preferred)
  - `sudo apt-get install python3`
  - `sudo apt-get install python`
- pip for your python version
  - `sudo apt-get install python3-pip`
  - `sudo apt-get install python-pip`
- Google api python client
  - `pip3 install --upgrade google-api-python-client`
  - `pip install --upgrade google-api-python-client`

Usage

Step 1

The program requires a few arguments that should be put in a file name `args` with the following format:

`notification_email`

`user_email`

`link_key1, link_key2, ..., link_keyN`

In other words, line 1 should only contain the email the HARO notification with found links should be sent to.

Line 2 should only contain the email the HARO emails are sent to and it should be the same email address the user intends to log into in step 2.

Finally, line 3 should contain a comma separated list of link keywords/phrases that the program will search for in HARO emails.

Step 2

Running Haro Listener is very easy. Simply open up a `Terminal` and type `python3 harolistener.py` and, if prompted, login to your Gmail account (same account as `user_email` above) and accept all the permissions the program requests.

The program will begin logging to the terminal window and, once setup is complete, go into the process of checking for new emails every 30 minutes.

Step 3

To quit the program, navigate back to the terminal window where you ran it and type `Cmd + C` (mac)/`Ctrl + C` (windows/linux) at the same time.

Module Descriptions

- **harolistener.py** - Contains the `main()` method and `HaroListener` class
    - `main()` establishes a connection to the gmail service and sets the params needed for `HaroListener` it then creates a `HaroListener` instance and calls `messageLoop()`
    - `HaroListener` is the main object for the program. Its stores all the parameters for the program such as the various email addresses as well as the connected gmail `service`.

  Additionally, it handles the processing of messages and notification of the `notification_email` if any relevant links are found.

- **inboxmanager.py** - Handles any modification/management of the `user_email`'s inbox such as retrieving/creating labels and filters as well as assigning/udating email messages' labels through the `InboxManager` class.

- **messageutils.py** - Utilities that deal directly with email messages such as converting from raw to a MIME formatted message and scrubbing MIME messages for relevant links.

- **utils.py** - Misc. helper functions and useful utilities that do not fit in any of the above modules.