Le programme aggreg.py

Ricardo SUPRICE 5 juin 2024

Prérequis

- Une machine agrégateur sans interface graphique
- Un serveur http apache2 installé avec un hôte virtuel configuré
- Un utilisateur aggreg qui est l'utilisateur qui exécutera l'aggrégateur (le programme)
- L'utilisateur aggreg n'est pas administrateur de la machine (il n'a pas le droit de faire des sudo)

Résumé du programme

Le programme aggreg.py lit un flux RSS, extrait les informations importantes et les affiche clairement sur une page web. Pour plus de détails, veuillez continuer la lecture de ce document.

Guide d'installation et de paramétrage

1. Récupéré le programme :

   ➢ Pour pouvoir récupérer le programme via GitLab :

   - apt install git
   - git clone https://etulab.univ-amu.fr/s23001073/sae-203.git
   - ls sae-203

   ➢ Récupéré le programme en téléchargé l'archive :

   - Archive.tar.gz
   - scp archive.tar.gz <ip_aggregateur>: (il faut avoir ssh dans sa machine : apt install ssh)
   - Tar -xvzf archive.tar.gz
   - Ls archive

Ensuite avant tout lancement/action sur le programme, téléchargé les dépendances :

Yaml et feedparse

- apt-get install python3
- apt install python3-pip

- pip3 install PyYAML
- pip3 install feedparser

2. Paramétrage

   ➢ Crée un répertoire contenant un fichier de config dans /etc, demander les droits à votre administrateur si vous ne pouvez pas crée un répertoire dans /etc :

      - sudo mkdir /etc/aggreg

   ➢ Ensuite crée un fichier dans le répertoire crée juste précédemment.

      - sudo touch /etc/aggreg/aggreg.conf

Sinon crée un fichier de config dans le répertoire courant.

- Touch toto.conf

Compléter le fichier toto.conf comme dans la section plus basse.

➢ Dans le fichier config écrire :

- #sources: --> source des page rss
  # - http://serveur1/rss.xml
  # - http://server2/rss.xml
  # - url
  #rss-name: rss.xml --> nom du fichier
  #destination: /var/www/superviseur/index.html --> destinantion pour la création de la page web
  #tri-chrono: true --> pour trier:True = evenement les plus récent et False = evenement les plus critique

  #Exemple:
  sources:
  - http://serveur1/rss.xml
  - http://server2/rss.xml
  rss-name: rss.xml
  destination: /var/www/superviseur/index.html
  tri-chrono: True

Guide d'utilisation

3. Utilisation

   ➢ Une fois un fichier de config compléter lancer la commande :

      - ./aggreg.py

- Ou si le fichier aggreg.conf (/etc/aggreg/aggreg.conf) n'existe pas faire exécuter le programme avec un fichier de conf de qu'importe l'endroit, donc la commande finale
  - --> ./aggreg.py <path_file_config>

- Pour plus d'information voire la page d'aide du programme
  - ./aggreg -h ou ./aggreg.py --help

Explication du programme

Dans ce programme il y a 3 fonctions principales :

1. charge_urls(liste_url)
2. fusion_flux(liste_url, liste_flux, tri_chrono)
3. genere_html(liste_evenements, chemin_html)

La première fonction charge_urls(liste_url) a pour paramètre une liste d'URL, ses URL pointes vers la page web qui contient le flux RSS de chaque serveur. Donc la fonction récupère les documents RSS de chaque URL, chaque page est chargée et renvoie un dictionnaire ensuite les rangent dans une liste. Si une URL est inaccessible, on range la valeur None dans la liste.

La deuxième fonction fusion_flux(liste_url, liste_flux, tri_chrono) prend en paramètres une liste d'url, la même que celle dans la première fonction, une liste de flux, ce que renvoie la première fonction charge_urls et un booléen True ou False qui sert à choisir à partir de quoi la fonction va se basait pour trier les flux (Flase = la criticité d'un évènement, True = La date par ordre chronologique inverse).Cette fonction produit une liste unique d'événements contenant les événements de n flux, éventuellement triés par ordre chronologique inverse (c'est-à-dire les événements les plus récents en premier).

Et la troisième fonction genere_html(liste_evenements, chemin_html) prend en paramètres une liste d'événement structurée donc ce que renvoie la deuxième fonction fusion_flux et le chemin du fichier qui contiendra la page HTML générée. La fonction renvoie trois fichiers et un répertoire. Pour les fichiers, un fichier html, un fichier css et un autre javascript. Le répertoire créé par la fonction contient le fichier css.

Dans le programme principal main() j'utilise les fonctions précédemment définit. Tout d'abord définition d'une option que l'on peut utiliser en lançant la commande, cette option donne une aide à l'utilisateur. C'est ici qu'on regarde aussi s'il y a un fichier de

configuration par défaut ou si le fichier de configuration est placer en argument à côté de la commande. Ensuite on ouvre le fichier de config et on applique les fonctions précédentes. On a pour résultat à la fin de l'exécution du programme, une page web décrivant les évènements de chaque serveur. Dans le main() j'ai fait en sorte que le programme gère certaine erreur et aide l'utilisateur a les réglées.

Résolution des problèmes

4. Problèmes survenus ?
   - Si un problème survient quand la commande est lancée, lire l'erreur et faire ce qu'il y a marqué dans le terminal (le programme gère certaines erreurs et vous guide dans la résolution du problème)

   - Si ce n'est pas réglée, vérifier 1 par 1 les informations ci-dessous :
     - Fichier de config
     - Url des flux rss
     - Bonne installation des paquets
     - Bon droit sur les fichiers
   - Si erreur persistante contacter moi --> ricardo.suprice@etu.univ-amu.fr je me ferais un plaisir de vous aider !