

Lors de la réalisation de mon projet de scraping de données web relatives aux annonces de location de biens immobiliers sur la région lyonnaise je me suis confronté à un problème : l'impossibilité pour la machine de trouver mon profil firefox ; embêtant lorsqu'on veut scraper des données à l'aide de selenium. Après quelques recherches, il s'avère que le problème rencontré provient de l'installation de firefox en mode snap plutôt que deb. En effet, avec Ubuntu22.04, firefox est conteneurisé dans un snap. Cela entrave l'accès de notre application au profil, sauf si vous exécutez votre script sur pycharm ou vscode. Par conséquent, le lancement du navigateur web Firefox est impossible. Alors vous avez deux solutions : préciser le chemin vers votre profil firefox à selenium ou désinstaller le snap pour réinstaller une version deb de firefox. La première solution étant évidente à réaliser, nous allons nous concentrer sur la seconde. On va donc désinstaller le snap et le fichier de transition Firefox, puis, installer la version deb de Firefox depuis l'APT mozilla. Enfin, pour assurer que les mises à jour seront faites depuis apt-mozilla (deb), et non pas depuis le snap, on va utiliser un fichier de priorisation des mises à jour. Voici le [lien](#) vers le site qui m'a permis de réaliser ce billet.

Téléchargement du geckodriver.

Dans un premier temps, pour permettre à selenium d'accéder à firefox, vous devez disposer du geckodriver adéquat. Vous trouverez ce qu'il vous faut [ici](#). Une fois votre geckodriver téléchargé et extrait du fichier, vous pouvez le déplacer dans le fichier **'/usr/local/bin'** afin que python puisse le trouver facilement. Si vous ne savez pas quel geckodriver télécharger, vérifiez avant l'architecture de votre système ubuntu :

```
uname -m
```

```
x86_64
```

D'après mon architecture ubuntu je dois télécharger la version 'v0.35.0-linux64.tar.gz' de geckodriver.

Pour accéder à la page github contenant le lien de téléchargement et télécharger notre geckodriver, on peut utiliser la commande "curl" suivi de l'option "-O".

```
curl -O  
https://github.com/mozilla/geckodriver/releases/download/v0.35.0/geckodrive  
r-v0.35.0-linux64.tar.gz
```

Cela téléchargera directement le fichier zipé dans notre dossier courant. Observez que l'option "-o" vous permet de spécifier un nouveau nom de fichier :

```
curl -O geckodriver.tar.gz  
https://github.com/mozilla/geckodriver/releases/download/v0.35.0/geckodrive  
r-v0.35.0-linux64.tar.gz
```

La commande 'curl' vous permet également de chaîner plusieurs téléchargements de fichier - e.g.
`curl -O monfichier.tar.gz -O unautre fichier.tar.gz -O blablabla.xlsx`

cURL me semble être plus complet, en terme d'options et donc de possibilités, que wget. Cependant wget à l'avantage d'être simple et rapide à utiliser :

```
wget
https://github.com/mozilla/geckodriver/releases/download/v0.35.0/geckodriver-v0.35.0-linux64.tar.gz
```

Maintenant, il faut décompresser le dossier pour obtenir le geckodriver, le rendre exécutable et l'envoyer dans le dossier 'usr/local/bin' afin de le rendre accessible à python-selenium.

```
tar -tf /home/jordan/Téléchargements/geckodriver-v0.35.0-linux64.tar.gz
tar -xf /home/jordan/Téléchargements/geckodriver-v0.35.0-linux64.tar.gz
chmod +x geckodriver
mv geckodriver /usr/local/bin
```

le commande **tar** permet de compresser ou de décompresser des fichiers d'extension tar. Les options **-tf** permettent de lister le contenu du fichier tar, puis **-xf** l'extraction du fichier spécifié. Pour une explication complète et détaillée de l'extension .tar ou de la double extension .tar.gz aller voir cette [article](#) ! Vous remarquerez la commande **chmod** suivi de l'option **+x**. Cela permet de rendre le geckodriver exécutable, de sorte que l'ordinateur et les programmes tel que python puissent l'exécuter.

Désinstallation du snap firefox. Nous allons passer à la désinstallation du snap firefox puis à sa réinstallation via le deb. D'ici quelques minutes vous n'aurez plus accès à firefox. Si vous avez besoin d'un navigateur pour poursuivre la lecture de cet article, pensez à en installer un autre. Par exemple : `sudo apt install epiphany-browser`.

On passe en mode administrateur, cela nous évitera de spécifier la commande sudo.

```
sudo -i
```

Mise à jour dépôt apt et suppression Firefox snap.

On met à jour notre système.

```
apt update
apt full-upgrade -y
```

On supprime le snap firefox ainsi que le fichier de transition firefox.

```
snap remove firefox  
apt remove firefox -y
```

Ajout des dépôts apt-Mozilla.

Import de la clé publique signant les paquets Mozilla dans notre système.

```
wget https://packages.mozilla.org/apt/repo-signing-key.gpg -O  
/etc/apt/keyrings/packages.mozilla.org.asc
```

Ajout du dépôt Mozilla.

```
echo "deb [signed-by=/etc/apt/keyrings/packages.mozilla.org.asc]  
https://packages.mozilla.org/apt mozilla main" >  
/etc/apt/sources.list.d/mozilla.list
```

Rafraichissement du cache apt.

```
apt update
```

Mise en place de la priorisation des paquets apt.

Création du fichier firefox/nosnap.

```
vim /etc/apt/preferences.d/firefox-deb-nosnap
```

```
Package: firefox*  
Pin: release o=Ubuntu*  
Pin-Priority: -1  
Package: *  
Pin: origin packages.mozilla.org  
Pin-Priority: 1000
```

Installation de firefox depuis le deb.

Installation de firefox et du package de langue français.

```
apt install firefox  
apt install firefox-l10n-fr
```

Vérification de l'installation.

```
apt show firefox
```

On à bien une version firefox.deb. Notre geckodriver est désormais efficace ; on peut utiliser selenium avec un webdriver Firefox sans risque qu'il ne trouve pas notre profil.