

Supposons que nous

Supposons que nous sachions comment accéder à Internet de manière scientifique, alors comment regarder YouTube sur une télévision ? Configurer le routeur peut être un peu compliqué. Ici, nous allons utiliser une application pour y parvenir.

SmartYoutubeTV

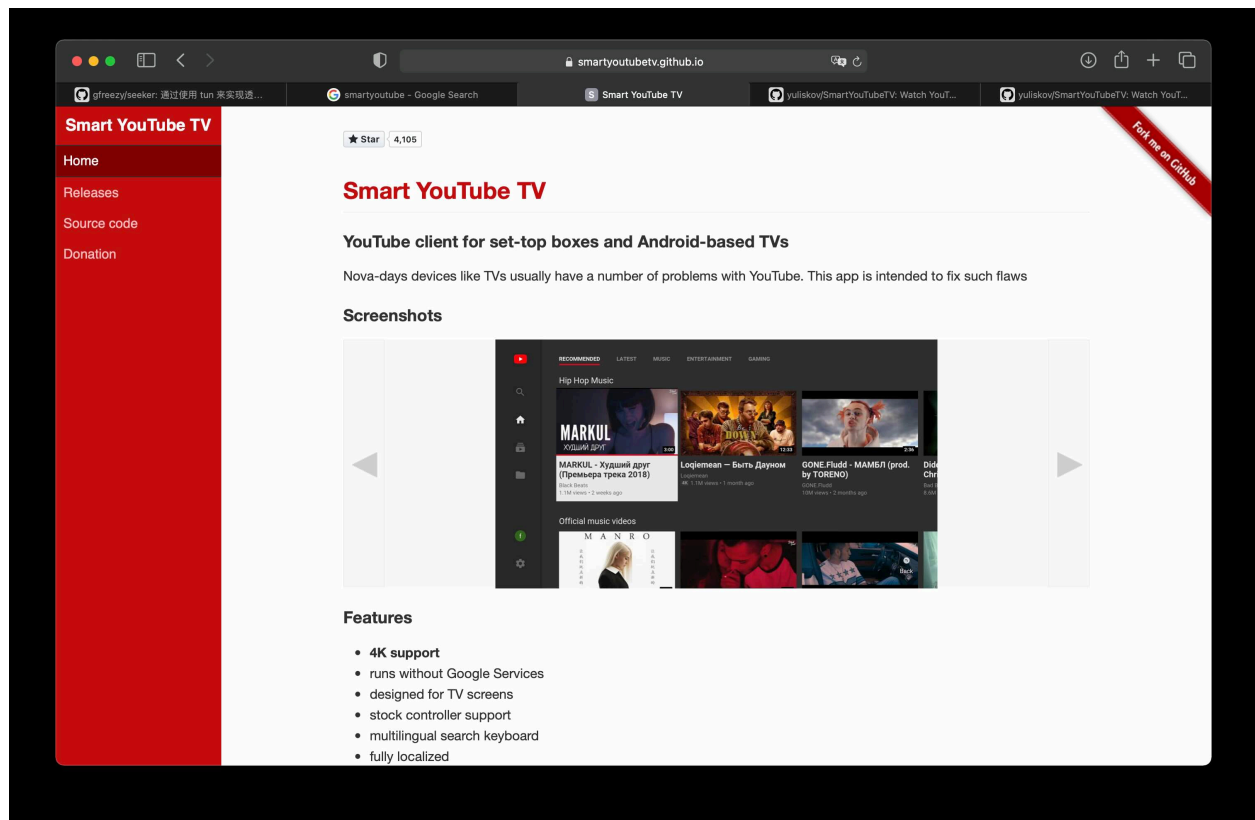


Figure 1: smart

Téléchargez-le. Installez-le sur votre télévision à l'aide d'une clé USB.

Ensuite, dans l'application cliente de l'accès scientifique à Internet, sélectionnez **Allow connect from Lan**. Cela signifie que d'autres appareils sur le réseau local peuvent se connecter à notre appareil pour accéder à Internet.

Ensuite, dans les options de configuration de SmartYoutubeTV, il suffit de définir le port.

Après avoir effectué les réglages, cliquez sur le bouton **Tester** pour essayer. Notez que j'ai utilisé ici un proxy de type **SOCKS**. J'ai essayé plusieurs fois avec **HTTP** sans succès. Une fois le

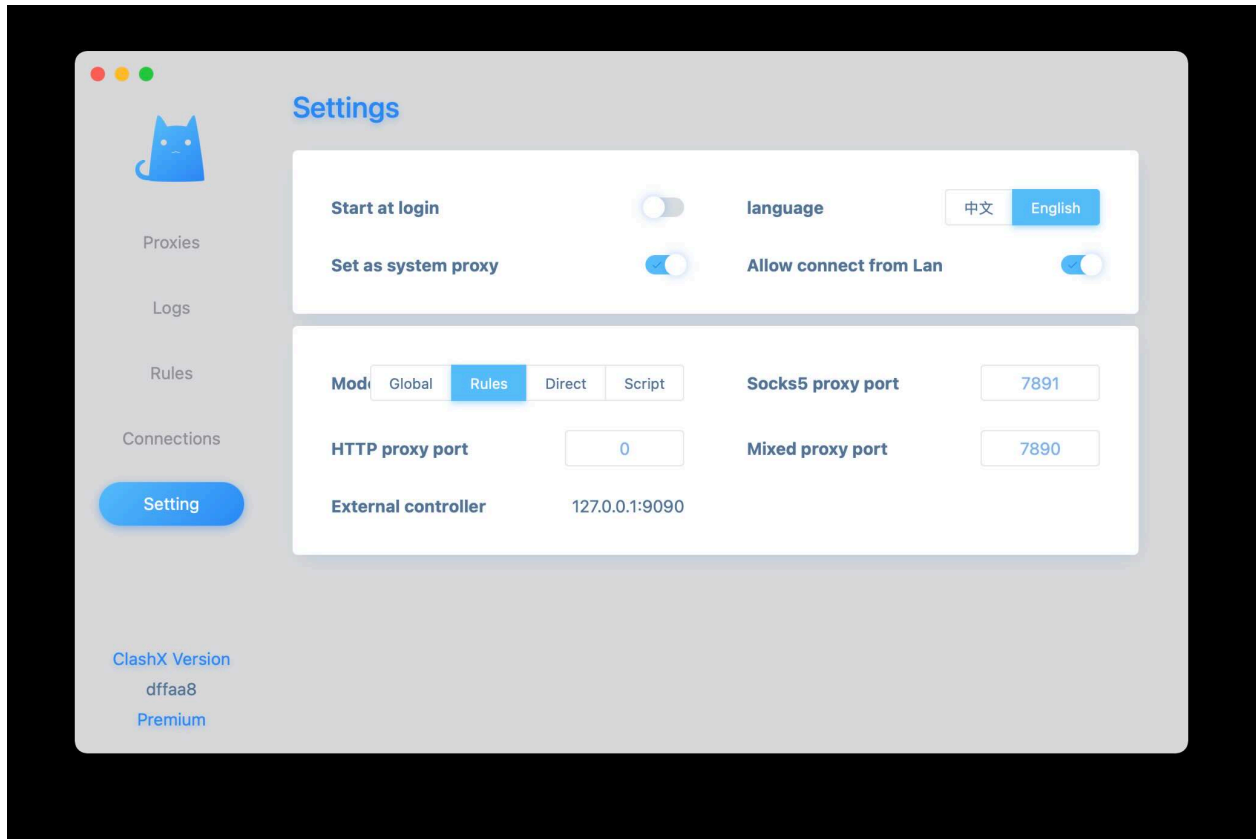


Figure 2: clash

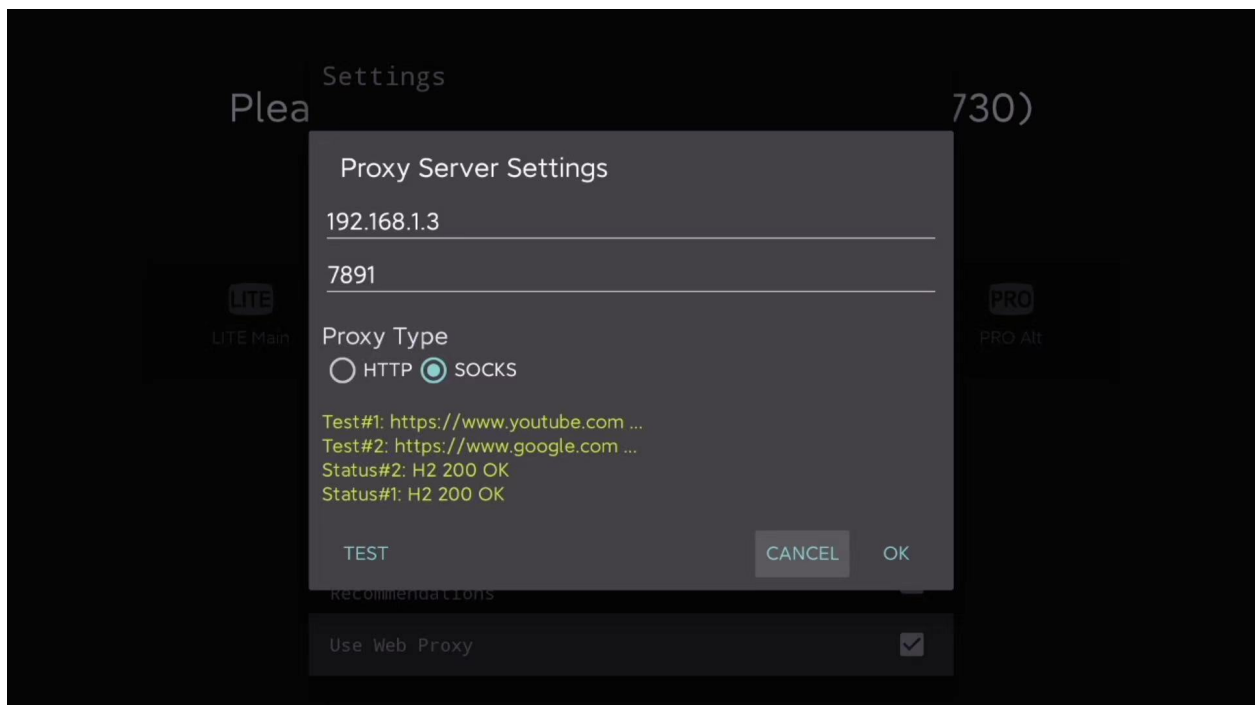


Figure 3: proxy1

test réussi, cliquez sur OK, puis testez à nouveau. Ensuite, il n'est pas nécessaire que vous configuriez 192.168.1.3, cela dépend de l'adresse de votre réseau local sur votre ordinateur.

C'est vraiment pratique, on peut voir tout de suite.

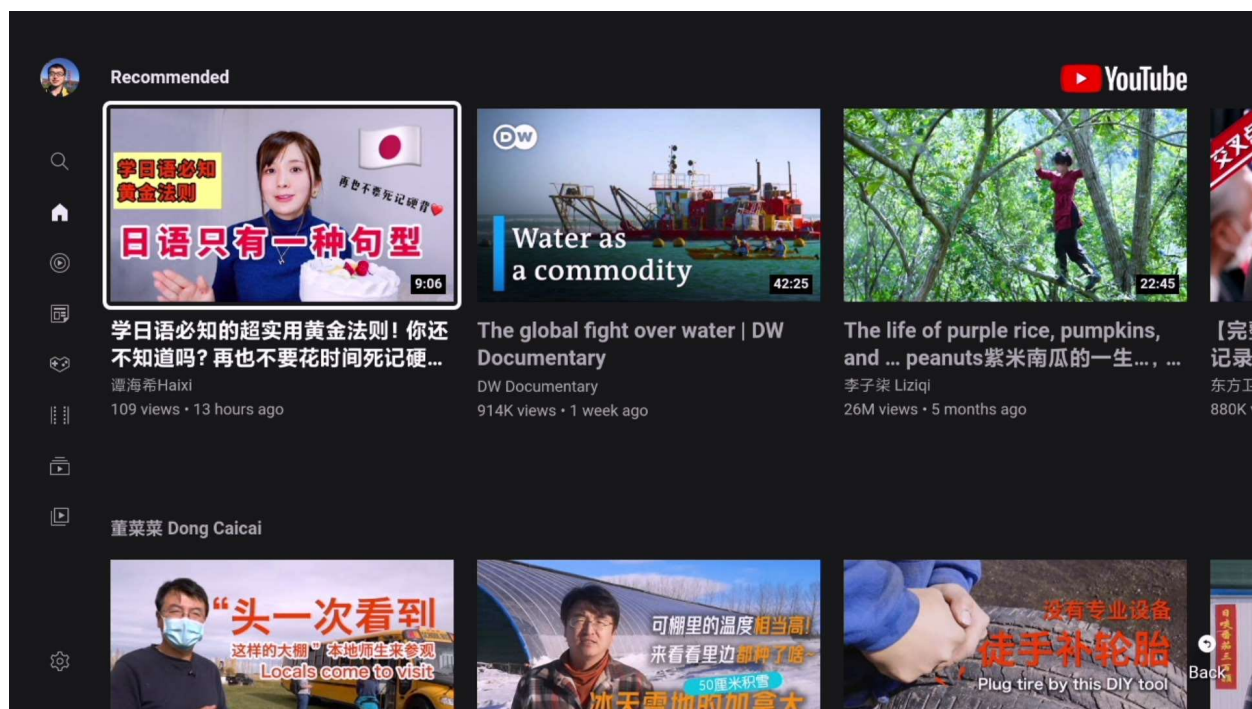


Figure 4: tan

gfreezy/seeker

Voici un projet GitHub. La page d'accueil du projet contient des instructions d'utilisation. Ici, nous allons principalement ajouter quelques points supplémentaires.

Il utilise tun pour implémenter un proxy transparent. Il réalise des fonctionnalités similaires au mode amélioré et au mode passerelle de Surge.

Dès le début, j'ai utilisé seeker pour transformer mon ordinateur en routeur de navigation scientifique. Voici ma configuration :

```
verbose: true
dns_start_ip: 10.0.0.10
dns_servers:
  - 223.5.5.5:53
  - 114.114.114.114:53
```

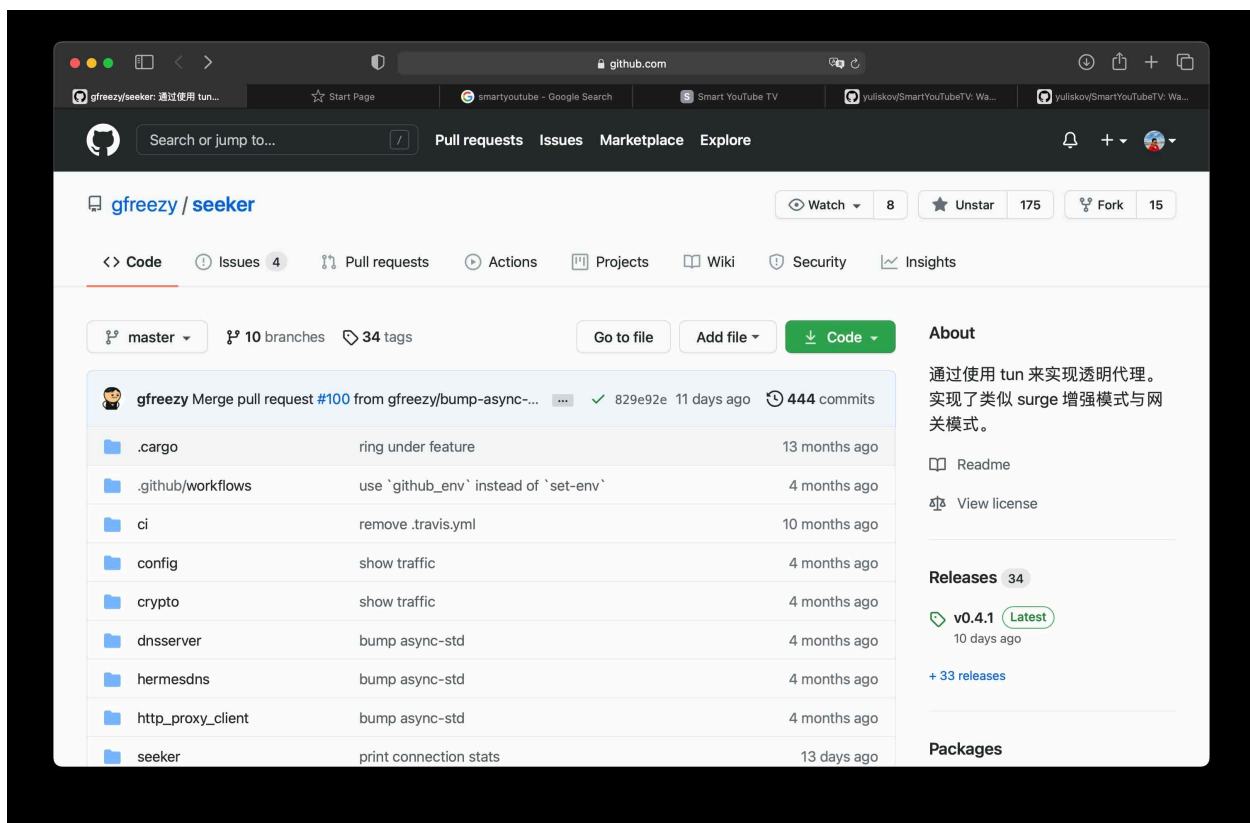


Figure 5: seeker

```
dns_timeout: 1s
tun_name: utun4
tun_ip: 10.0.0.1
tun_cidr: 10.0.0.0/16
dns_listen: 0.0.0.0:53
gateway_mode: true
ping_timeout: 2s
probe_timeout: 30ms
connect_timeout: 1s
read_timeout: 30s
write_timeout: 5s
max_connect_errors: 2
```

servers: - name: serveur proxy HTTP addr: 0.0.0.0:7890 protocol: Http

```
- name: serveur proxy https
  addr: 0.0.0.0:7890
  protocol: Https
```

rules: - 'MATCH,PROXY'

Au début, j'utilisais un proxy `socks5`. J'ai configuré comme ceci :

```
```yaml
servers:
 - name: serveur proxy socks5
 addr: 0.0.0.0:7891
 protocol: Socks5
```

Cependant, il y a pas mal de problèmes. La connexion échoue souvent. La documentation contient ce passage :

Lorsque vous utilisez un proxy socks5, vous devez configurer tous les domaines en connexion directe dans le fichier de configuration. Si vous utilisez ss ou vmess, vous devez également ajouter le domaine du serveur ss ou vmess dans le fichier de configuration. Sinon, cela pourrait entraîner une boucle infinie et empêcher une utilisation normale.

C'est peut-être la raison.

Utiliser `seeker` implique qu'il faut avoir un ordinateur qui le fait tourner, en l'utilisant comme un routeur. En revanche, la configuration via `proxy` offre beaucoup plus de flexibilité. Je peux utiliser un iPhone ou un téléphone Android pour partager le port du proxy.

## Capture d'écran de télévision

En écrivant cet article, j'ai réfléchi à la manière de capturer des captures d'écran sur une télévision. Chez moi, nous utilisons une télévision Xiaomi. Vous pouvez appuyer deux fois de suite sur le bouton `Home` de la télécommande pour faire apparaître le menu de gestion des applications.



Figure 6: tv\_screen

Vous voyez le bouton de capture d'écran ? Ensuite, il est très pratique de partager directement sur WeChat. Ici, vous pouvez également fermer toutes les applications. Si certaines applications se bloquent, vous pouvez les gérer de cette manière.

Très bien. Utilisons une télévision grand écran pour observer le monde.