

CheeseShop

Michel Casabianca
casa@sweetohm.net

Sur Github: <https://github.com/c4s4/cheeseshop>

CheeseShop est un repository de packages Python, soit une version locale du célèbre <http://pypi.python.org>. Ceci est utile pour les utilisateurs en entreprise qui ont besoin de partager des bibliothèques Python privées entre développeurs.

Pour dire à PIP où se trouve votre CheeseShop privé, vous devez éditer votre fichier `~/.pip/pip.conf` :

```
[global]
index-url = http://my.shop.host/simple
trusted-host = my.shop.host
```

Où `my.shop.host` est le nom d'hôte de la machine exécutant CheeseShop. PIP appellera votre CheeseShop pour obtenir des paquets. Si CheeseShop n'héberge pas ce paquet, il redirigera PIP vers Pypi standard.

Pour dire à `setup.py` où télécharger votre paquet, vous devez éditer le fichier `~/.pypirc`:

```
[distutils]
index-servers =
    pypi
    cheeseshop

[pypi]
username: ***
password: ***

[cheeseshop]
username: spam
password: foo
repository: http://my.shop.host/simple/
```

`setup.py` appellera votre CheeseShop si vous le nommez sur la ligne de commande :

```
$ python setup.py sdist upload -r cheeseshop
```

Où `-r cheeseshop` est l'option qui indique la connexion que vous voulez utiliser. Il doit y avoir une entrée correspondante dans votre fichier de configuration `~/.pypirc`. N'oubliez pas d'ajouter `cheeseshop` dans la liste `index-server` au début du fichier.

CheeseShop est capable de fonctionner sur HTTP et/ou HTTPS et effectue une authentification de base si nécessaire.

Installation

Téléchargez l'archive binaire sur <https://github.com/c4s4/cheeseshop/releases>, décompressez-la et copiez l'exécutable binaire de votre plate-forme (nommé *cheeseshop-system-platform* dans le répertoire *bin*) quelque part dans votre *PATH* et le renommer *cheeseshop*. Cet exécutable n'a besoin d'aucune dépendance ni d'aucune machine virtuelle pour s'exécuter.

Il existe des binaires pour les plates-formes suivantes :

- Linux 386, amd64 et arm.
- FreeBSD 386, amd64 et arm.
- NetBSD 386, amd64 et arm.
- OpenBSD 386 et amd64.
- Darwin (MacOSX) 386 et amd64.
- Windows 386 et amd64.

Usage

Pour lancer CheeseShop, tapez sur la ligne de commande :

```
$ cheeseshop
```

Il recherchera un fichier de configuration aux emplacements suivants :

- *~/cheeseshop.yml*
- */etc/cheeseshop.yml*

Vous pouvez également passer le chemin d'accès au fichier de configuration sur la ligne de commande :

```
$ cheeseshop /path/to/cheeseshop.yml
```

Configuration

Le fichier de configuration devrait ressembler à ceci :

```
# The root directory for packages
root: /home/cheeseshop
# Path to the server certificate
cert: /etc/ssl/certs/cheeseshop-cert.pem
# Path to the server key
key: /etc/ssl/private/cheeseshop-key.pem
# The HTTP port CheeseShop is listening
```

```
http: 80
# The HTTPS port CheeseShop is listening
https: 443
# The URL path
path: simple
# Redirection when not found
shop: pypi.python.org/simple
# Tells if we can overwrite an existing package
overwrite: false
# List of users and their MD5 hashed password
# To get MD5 sum for password foo, type 'echo -n foo | md5sum'
# To disable auth when uploading packages, set auth to ~
auth:
  spam: acbd18db4cc2f85cedef654fccc4a4d8
  eggs: 37b51d194a7513e45b56f6524f2d51f2
```

Il y a un exemple de configuration dans le fichier *etc/cheeseshop.yml* de l'archive.

root

Le répertoire racine où se trouvent les paquets Python. Sous cette racine, il y a un répertoire pour chaque paquet. Les fichiers pour les versions de ce paquet sont dans ces sous-répertoires. Ainsi, si notre repository héberge des paquets *spam* (dans les versions *1.0.0* et *1.1.0*) et *eggs* (dans les versions *1.0.0* et *1.0.1*) nous aurions la structure de répertoire suivante :

```
$ tree
.
├── spam
│   ├── spam-1.0.0.tar.gz
│   └── spam-1.1.0.tar.gz
└── eggs
    ├── eggs-1.0.0.tar.gz
    └── eggs-1.0.1.tar.gz
```

Vous devez créer ce répertoire et vous assurer que l'utilisateur qui exécute le serveur a le droit d'écrire dans ce répertoire.

Il est fortement conseillé de sauvegarder ce répertoire.

key

C'est le chemin de la clé privée CheeseShop. Pour générer une telle clé, vous pouvez taper :

```
$ openssl genrsa -out cheeseshop-key.pem 2048
```

Cela va générer un fichier *cheeseshop-key.pem* que vous devez copier dans le répertoire */etc/ssl/private*, qui est l'endroit standard.

Ceci est seulement nécessaire lors de l'exécution du serveur HTTPS. Si vous exécutez uniquement HTTP, vous pouvez définir cette valeur sur `~`.

cert

C'est le chemin vers le certificat CheeseShop. Pour générer un certificat auto-signé, vous pouvez taper :

```
$ openssl req -new -x509 -key cheeseshop-key.pem -out cheeseshop-cert.pem -days 365
```

Cette commande vous demandera beaucoup de champs, mais le seul qui soit nécessaire est le nom *FQDN* qui est le nom d'hôte de la machine qui exécute CheeseShop. Un fichier nommé *cheeseshop-cert.pem* sera généré; vous devriez copier ce fichier dans le répertoire */etc/ssl/certs*, qui est l'emplacement standard.

Notez que si vous avez un certificat généré par une autorité de certification, vous n'aurez peut-être pas besoin d'ajouter un *trusted-host* dans votre configuration PIP. Mais je n'ai un tel certificat et n'ai donc pas été incapable de le tester.

http

C'est le numéro de port que le serveur HTTP écoutera pour les connexions entrantes. Réglez-le sur *0* pour désactiver HTTP (et exécuter uniquement sur HTTPS). Notez que ce n'est pas une bonne idée d'effectuer une authentification de base sur HTTP, car quiconque intercepte des requêtes HTTP peut connaître votre nom d'utilisateur et votre mot de passe. Le port standard pour HTTP est *80* mais le serveur doit fonctionner en tant que root pour pouvoir écouter sur ce port. Si vous n'exécutez pas le serveur en tant que root, vous devez écouter sur un numéro de port supérieur à *1024*.

https

C'est le numéro de port que le serveur HTTPS écoute. Réglez-le sur *0* pour désactiver HTTPS. Si HTTPS est activé, vous devez fournir une clé privée et un certificat (dans les champs de configuration *key* et *cert*). Le port standard pour HTTPS est *443* mais le serveur doit fonctionner en tant que root pour pouvoir écouter sur ce port. Si vous n'exécutez pas le serveur en tant que root, vous devez écouter sur un numéro de port supérieur à *1024*.

path

C'est le chemin de l'URL que le serveur va écouter. La valeur par défaut est *simple*, donc pour lister tous les paquets, vous devez ouvrir l'URL <http://my.shop.host/simple>. Pour lister la version disponible du paquet *spam*, vous devez ouvrir l'URL <http://my.shop.host/simple/spam>. Pour télécharger la version *1.2.3* de ce paquet, vous devez ouvrir <http://my.shop.host/simple/spam/spam-1.2.3.tar.gz>. Cette valeur ne devrait pas être modifiée.

shop

C'est l'URL du repository de paquets public sans protocole (*http* ou *https*), tel que *pypi.python.org/simple*. Cette valeur ne devrait pas être modifiée.

overwrite

Indique si nous pouvons écraser un paquet existant lors du téléchargement (avec `setup.py upload`). Si défini sur *false* (la valeur par défaut), vous devez télécharger le paquet manuellement sur le serveur pour modifier une version (ce qui n'est **pas** recommandé), et une tentative aboutira à un status code *400*. Si la valeur est *true*, il est possible d'écraser un package.

auth

Ceci est la configuration d'authentification de base. Si vous ne voulez pas d'authentification, définissez cette valeur sur `~`. Ceci est une liste de noms d'utilisateur et du hash *MD5* de leur mot de passe. Pour obtenir le hachage *MD5* d'un mot de passe donné, vous pouvez taper la commande suivante :

```
$ echo -n foo | md5sum
acbd18db4cc2f85cedef654fccc4a4d8 -
```

Notez que si vous modifiez cette configuration, vous devez redémarrer le serveur car cette configuration est chargée au démarrage.

Service

Pour installer CheeseShop en tant que service *System V*, éditez le fichier de configuration d'exemple se trouvant dans le fichier *etc/cheeseshop.init*. Vous devez éditer la variable *SCRIPT* pour définir le chemin d'accès à la commande *cheeseshop*. Puis copiez ce fichier dans */etc/init.d/cheeseshop*.

Vous devez également modifier le fichier de configuration *etc/cheeseshop.yml* pour définir l'emplacement du référentiel dans la variable *root*. Copiez ce fichier à l'emplacement */etc/cheeseshop.yml*.

Vous pouvez ensuite démarrer le service avec :

```
$ sudo service cheeseshop start
```

Et arrêtez avec :

```
$ sudo service cheeseshop stop
```

Vous pouvez voir les logs dans le fichier */var/log/cheeseshop.log*.

Pour lancer le service au démarrage de la machine, vous devez taper:

```
$ sudo update-rc.d cheeseshop defaults
```

Et pour désactiver le lancement au démarrage :

```
$ sudo update-rc.d -f cheeseshop remove
```

Builder CheeseShop

Pour construire CheeseShop, vous devez installer [Goyaml](#) et [GOX](#) avec les commandes suivantes :

```
$ go get gopkg.in/yaml.v2
$ go get github.com/mitchellh/gox
$ gox -build-toolchain
```

Ensuite, vous pouvez utiliser le fichier makefile pour créer la version binaire pour votre plate-forme :

```
$ make build
```

Pour créer des binaires pour toutes les plateformes, tapez :

```
$ make compile
```

Enjoy!