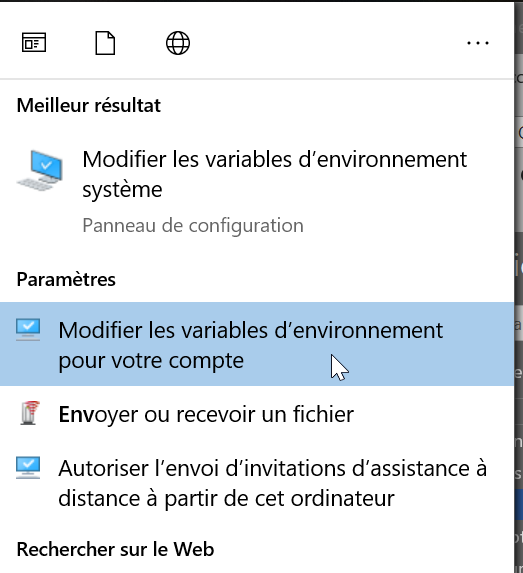
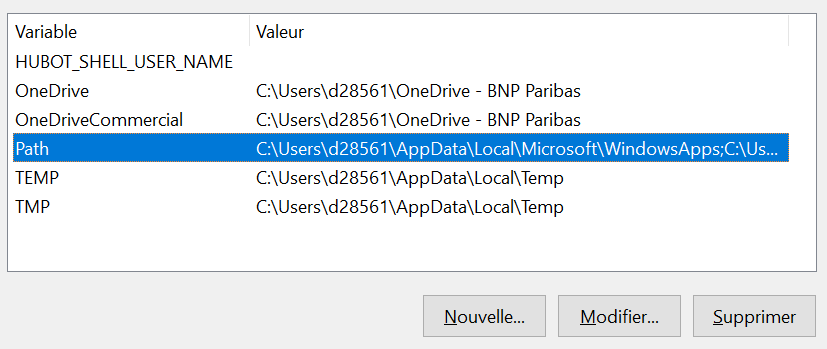
DexBot

# Installation

Dans ce document, sera nommé *<DEXBOT\_HOME>* le répertoire où est installé le chatbot.

## Node js et npm

1. Installer node.js (via le Digital Working Store, le lien est sur le bureau)
2. Config du proxy pour npm :  
   dans un cmd (pas la peine d’aller dans un répertoire spécifique) :  
   npm config set proxy [http://%USERNAME%:PWD@vip-pxp0-std.fr.net.intra:8080](http://%25USERNAME%25:PWD@vip-pxp0-std.fr.net.intra:8080)  
   npm config set https-proxy [http://%USERNAME%:PWD@vip-pxp0-std.fr.net.intra:8080](http://%25USERNAME%25:PWD@vip-pxp0-std.fr.net.intra:8080)  
     
   PWD => Le mot de passe echonet  
   *\* pour les caractères spéciaux dans le mot de passe : https://en.wikipedia.org/wiki/List\_of\_XML\_and\_HTML\_character\_entity\_references)*  
     
   Pour vérifier ce qui a été mis, vous pouvez utiliser la commande :  
   npm config list
3. Ajouter une variable d'environnement pour npm
   1. Appuyer sur <Win>
   2. Taper "env"
   3. Cliquer sur "modifier les variables d'environnement pour votre compte"  
      
   4. Sélectionner la variable "Path" et cliquer sur « Modifier »  
      
   5. Cliquer sur "Nouveau" et mettre la valeur "C:\Users\%USERNAME%\AppData\Roaming\npm"
   6. OK puis OK

## Hubot

(Référence : <https://hubot.github.com/docs/>)

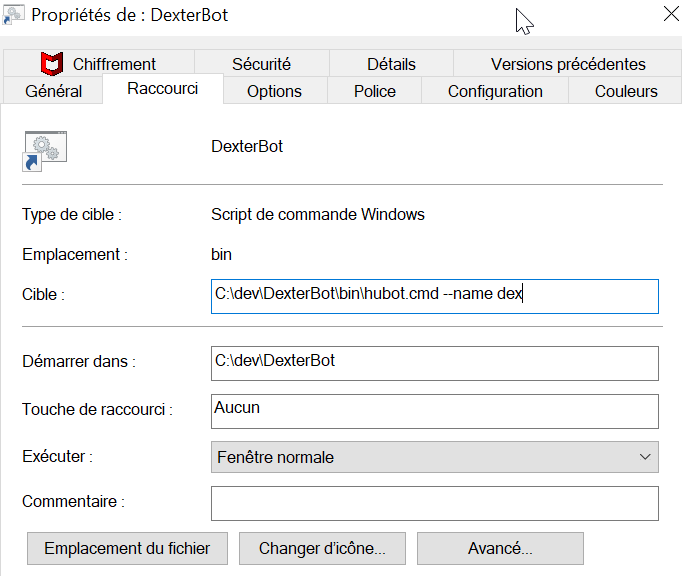
1. Dans un cmd

*npm install -g yo generator-hubot*

* 1. Si vous rencontrez l’erreur :  
     « npm ERR! code SELF\_SIGNED\_CERT\_IN\_CHAIN »  
     🡪 tentez de désactiver l’utilisation du ssl :  
     *SET npm\_config\_strict-ssl=false*

1. Aller dans un répertoire de votre choix (pour moi c:\dev\DexterBot) et dézippez le package (copiez/collez le contenu d’une machine à l’autre).

Pour le démarrer sans avoir à ouvrir un cmd, faire un raccourci avec ces paramètres (vous aurez sûrement à changer les chemins) :



# Description de l’arborescence

Ci-dessous, l’utilité des fichiers créés spécifiquement pour ce bot.

|  |  |  |  |
| --- | --- | --- | --- |
| Rép. | Fichier | Description | Mise à jour |
| data | triaplat.csv | (Certaines) données du triaplat au format csv | A chaque modification du triaplat |
| data | requetes\_sql.json | (typique au pkg Dexter)  Quelques requêtes SQL utiles | A volonté |
| data | ReferentielCft.csv | Copie du référentiel CFT | A chaque modification du referentielCFT.csv |
| data | monitoring.csv | Informations de monitoring de certains traitements | A volonté ou lorsque les traitements sont modifiés |
| data | entries.json | Points d’entrée de quelques traitements | A volonté ou lorsque les traitements sont modifiés |
| data | CIDS.json | (typique au pkg Dexter)  C\_ID (isl) de quelques contributeurs | A décommissionner : en doublon avec le monitoring.csv |
| data | autosys\_scripts.json | Scripts appelés par les jobs autosys | A chaque modification des jobs autosys |
| data | autosys\_calendars.json | Calendriers autosys | A chaque modification des calendriers autosys |
| libs | csvMonitoringLoader.coffee | Librairie de chargement du fichier « monitoring.csv » (format csv ayant ; comme séparateur) | n.a. |
| libs | csvTriaplatLoader.coffee | Librairie de chargement du triaplat au format csv (avec un ; comme séparateur) | n.a. |
| libs | jsonLoader.coffee | Librairie de chargement des fichiers au format json | n.a. |
| libs | referentielCftLoader.coffee | Librairie de chargement du référentielCFT | Si le référentiel cft change de format |
| scripts | autosys.coffee | Commandes liées aux fichiers autosys\_scripts.json et autosys\_calendars.json | n.a. |
| scripts | ISL.coffee | (typique au pkg Dexter)  Commandes liées au fichier CIDS.json | n.a. |
| scripts | smartCalls.coffee | Fonction de reconnaissance de la donnée en entrée (idf, job ou calendrier). | n.a. |
| scripts | triaplat.coffee | Commandes liées au fichier triaplat.json | n.a. |
| scripts | entries.coffee | Commandes liées au fichier entries.json | n.a. |
| scripts | monitoring.coffee | Commandes liées au fichier monitoring.csv | n.a. |
| scripts | sqlRequests.coffee | (typique au pkg Dexter)  Commandes liées au fichier requetes\_sql.json | n.a. |
| scripts | referentielCft.coffee | Commandes liées au fichier ReferentielCft.csv | n.a. |
| scripts | memory.coffee | Commandes permettant de faire persister des valeurs d’une requête à l’autre | n.a. |

# Mise à jour les données

## ReferentielCft

Sur le serveur de prod (s00va9943923), récupérer le fichier :

/apps/sys/scripts/DEXO/batchs/cft/send/ReferentielCft.csv

Et le copier dans le dossier <DEXBOT\_HOME>/data

## Triaplat

1. Récupérer le triaplat.xlsx
2. Ne garder que les colonnes suivantes :  
   jobname  
   job\_type  
   owner  
   start\_times  
   box\_name  
   command  
   alias\_machine  
   condition  
   run\_calendar  
   description  
   run\_window
3. Enregistrer en csv (en utilisant des points virgule comme séparateur) avec le nom « triaplat.csv »
4. Copier le fichier dans *<DEXBOT\_HOME>/data*

## Scripts autosys

[A faire une seule fois]

Créer un script sur l’AIX (dans votre home ?) avec ce contenu :

*#!/bin/sh*

*AS\_SCRIPT\_DIR=/apps/exploit/autosys/DEOP02/script/*

*echo [*

*for f in $(find $AS\_SCRIPT\_DIR -name "\*ksh" -print)*

*do*

*found=`expr match "$f" '.\*/\(IDEO.\*\)'`*

*if [ -n "$found" ]; then*

*calledScript=`cat $f | grep ^[^#].\*apps`*

*calledScript=`echo $calledScript | sed 's/[\\]//g'`*

*calledScript=`echo $calledScript | sed 's/\"/\\\"/g'`*

*echo { \"as\_script\":\"$found\",\"called\_script\":\"$calledScript\" },*

*fi*

*done*

*echo { \"as\_script\":\"\",\"called\_script\":\"\" }*

*echo ]*

Le nommer (par exemple), generation\_scripts\_autosys.sh

Lui donner les droits nécessaires pour s’exécuter :

*chmod 700 generation\_scripts\_autosys.sh*

[Génération du fichier de données]

Lancer le script (de préférence en se connectasnt avec le user root) en redirigeant ce qu’il affiche dans le fichier autosys\_scripts.json.

*./generation\_scripts\_autosys.sh > autosys\_scripts.json*

Puis copier le fichier généré (autosys\_scripts.json) dans le répertoire :

*<DEXBOT\_HOME>/data*

# Utilisation

Dans l’invite du bot, taper « help ». L’écran ci-dessous devrait s’afficher :



Voici un diagramme non exhaustif des commandes de base :

