**HLD sbc\_auto\_check**

|  |  |  |
| --- | --- | --- |
| Version | Mise à jour par | Date |
| 1.2 | SUEUR Emmanuel | 02/10/2021 |

**1/But du document**

Explication des différentes partie du programme pour la reprise et la modification par des développeur ou toute personne autorisées.

**2/Environnement**

Le développement a été effectué pour l’architecture Oracle SBC et ECB en 8.4 ou ultérieur. La version 8.4 a vu le développement des API REST permettant l’automatisation de certaines tâches non permis par les feature implémentées dans l’interface web.

Le programme a été réalisé sous python 3

Les librairies utilisé sont :

import requests  
import re  
import xml.etree.ElementTree as ET  
import phonenumbers  
**import time**

**/ !\ Si l’une de ses librairies n’est pas disponibles dans l’environnement de production le script ne pourra s’exécuter correctement.**

**3/Architecture sbc\_auto\_check**

Le programme ce constitue dans sa version actuel de 3 partie distinct.

3.1 API

Le développement a été effectué en s’appuyant sur la documentation Oracle Officiel, <https://docs.oracle.com/cd/E95618_01/html/sbc_scz810_monitoring/title.htm>

Une requête post pour la récupération du token . Dans la version de travail le login est en claire, il est recommandé lors de la mise en production de passer par un échange de clés publics.

response\_tok = requests.post(url\_tok, headers=headers, verify=False, auth=('admin', 'Ytreytre'))

Une regex permet la récupetation du Token

**re.search('<accessToken>(.+?)</accessToken>', response\_tok.text).group(1)**

**/ !\ la récupération du token est important car utilisé dans le header lors du get**.

Une requête get pour la récupération en xml du fichier de configuration local-policy.

url\_poli = "https://" + sbc1ip + "/rest/v1.1/configuration/configElements?elementType=local-policy"  
resp = requests.get(url\_poli, headers=headers, verify=False)

Cette requête peut-être modifier en fonction de la configuration à vérifier (toute modification de ce champ demandera une adaptation de la suite du code).

3.2 Parse XML

Lors de la requête get la réponse est un fichier xml reprenant l’ensemble de la configuration pour l’objet demandé. Il est donc nécessaire de faire un filtre sur la description de pour permettre d’extraire seulement la configuration à analyser et non l’ensemble. Pour cela la variable isblocage est définie à faux.

isblocage = False

Cette variable n’a pour but que de matcher sur la description souhaité, ce qui permet d’implémenter la liste verif.number avec seulement les entrées souhaitées.

if name == "description":  
  
 # white space correction  
 attrib\_from\_address = re.findall(r"^\w+", attribute.find('value').text)  
  
 for blocage in attrib\_from\_address:  
 isblocage = blocage == "blocage"

Un regex est là pour éviter tout problème si un espace est présent dans le fichier xml.

3.3 Vérification des entrées

Lorsque la partie 3.2 est terminé la librairie phonenumbers va permettre d’incrémenter 2 listes. Une liste avec les numéros possible et une liste avec les numéros ne répondant pas aux règles de numérotation E164.

for numbers in verif\_number:  
 valid = phonenumbers.is\_valid\_number(phonenumbers.parse(numbers))  
 if valid == True:  
 valid\_number.append(numbers)

**4/Fichier de log**

Par default les différents fichiers de log seront créées à la racine du script lors de son exécutions.

Ils seront au nombre de 3 :

* ValidNumber\_ « date ».txt

Généré avec la liste des numéros valide dans la configuration de blocage

verifFile = open("ValidNumber"+"\_"+time.strftime("%Y%m%d-%H""h""%M""m""%S")+".txt", "w+")

* UnvalidNumber\_ « date ».txt

Généré avec la liste des numéros non valide dans la configuration de blocage

unvalidFile = open("UnvalidNumber"+"\_"+time.strftime("%Y%m%d-%H""h""%M""m""%S")+".txt", "w+")

* Auto\_check\_policy.log

Permutant de log les différentes étapes pouvant être en erreur et bloquante résultante d’une mauvaise utilisation ou d’un problème de configuration. Et permettant d’horodater la fin d’exécution du script

logprocessing = open("auto\_check\_policy.log", "w")