

Explication du script d'envoi de notifications :

```
1  #!/bin/bash
2  urlcdr='https://192.168.21.245:9298/1.0/cdr?from=2018-03-14&limit=1&order=id'
3  token=$(curl -k -X POST --header 'Content-Type: application/json' --header 'Authorization: Basic
4  cmVzdC1hcGktZGVzdDpidHNzaW8=' -d '{ "backend": "xivo_service", "expiration": 120000 }' 'https://192.168.21.245:9497/0.1/token' | jq -r .data.token)
5  echo $token
6  namedest=$(curl -k -X GET --header 'Accept: application/json' --header X-Auth-Token:$token $urlcdr |jq -r .items[0].source_name)
7  echo $namedest
8  numdest=$(curl -k -X GET --header 'Accept: application/json' --header X-Auth-Token:$token $urlcdr |jq -r .items[0].source_extension)
9  echo $numdest
10 IDdest=$(curl -k -X GET --header 'Accept: application/json' --header X-Auth-Token:$token $urlcdr |jq -r .items[0].destination_user_uid)
11 echo $IDdest
12 IDappel=$(curl -k -X GET --header 'Accept: application/json' --header X-Auth-Token:$token $urlcdr |jq -r .items[0].id)
13 echo $IDappel
14 adrmmail=$(curl -k -X GET --header 'Accept: application/json' --header X-Auth-Token:$token https://192.168.21.245:9486/1.1/users/$IDdest | jq -r .email)
15 echo $adrmmail
16 statutappel=$(curl -k -X GET --header 'Accept: application/json' --header X-Auth-Token:$token $urlcdr |jq -r .items[0].answered)
17 echo $statutappel
18
19 if [ ! -f "/home/btssio/listeappel/$IDappel" ] && [ $statutappel = false ] ;then
20 ifconfig eth0 down;
21 echo "Vous avez un appel en absence de la part de $namedest num : $numdest. Pour entendre le message veuillez appeler le *99"|mail -s "Appel en absence de
22 $numdest" $adrmmail;
23 touch /home/btssio/listeappel/$IDappel;
24 ifconfig eth0 up;
25 sleep 12
26 else
27 echo "le fichier existe déjà donc pas de mail"
28 fi
```

Ce script va permettre d'envoyer des notifications lorsqu'un appel est manqué. Par exemple l'utilisateur A appelle l'utilisateur B. L'utilisateur B ne peut pas répondre à l'appel. Un mail va être envoyé à l'utilisateur B afin de le prévenir que l'utilisateur A a voulu le joindre et a pu potentiellement lui laisser un message.

Cette fonctionnalité est déjà disponible nativement dans Wazo, cependant les mails sont envoyés via le réseau du lycée qui ne permet pas d'envoyer des mails rapidement. Ces derniers semblent bloqués par le pare-feu.

Le script est ainsi exécuté via le Raspberry qui va utiliser le réseau 4G pour envoyer le mail au lieu d'utiliser le réseau du lycée.

Dans un premier temps, on récupère un token qui va être utilisé pour récupérer les informations dans l'API. Ces informations sont récupérées sous la forme de fichier JavaScript (grâce à la commande Curl), le fichier est ensuite traité pour obtenir une valeur précise (grâce à la commande jq qui permet de fragmenter le fichier js).

Une fois les informations récupérées, un IF vérifie si l'appel est en absence grâce à la variable \$statutappel, de plus lors de l'envoi d'un mail un fichier unique est créé (à l'emplacement /home/btssio/listeappel/\$IDappel) pour spécifier que l'appel a été traité ainsi si ce fichier existe on ne rentre pas dans le IF

Si l'appel est en absence et n'a pas déjà été traité par le script, on rentre dans le if, on éteint l'interface réseau qui correspond au réseau du BTS et on envoie un mail à l'aide des différentes informations qui ont été récupérées au préalable (adresse mail du destinataire, nom de l'appelant, numéro de l'appelant)

En outre au moment de l'envoi du mail on désactive l'interface réseau connecté au réseau du BTS afin d'être sûr que le mail va être envoyé via le réseau 4G (ifconfig eth0 down)