**Documentation Aircall**

**Objectif:**

Le projet représente une API qui va faire le lien entre le n° de téléphone public fourni par Aircall pour le client Eurofins et un n° de téléphone intermédiaire SVI, selon la logique suivante:

\* Si un appel entrant est fait pendant les heures d’ouverture, il sonnera selon une cascade définie dans le tableau de bord d’Aircall.

\* Sinon, s’il est effectué en dehors des heures d'ouverture, l’appel sera acheminé vers le numéro SVI, offrant un soutien 24/7.

**Numéros Aircall:**

1. **le numéro principale:** (Eurofins +33 4 84 80 07 43)

Ce numéro est visible aux clients, et contient la logique des appels entrants qui sont redirigés vers les agents concernés, pour reconfigurer cette logique il faut se connecter sur un compte administrateur et accéder au workflow directement sur le tableau de bord d'Aircall.

**Remarque:** En cas de changement de ce numéro principale, on doit récupérer son nouveau ID depuis l'url de ce numéro dans le tableau de bord, ensuite, on doit changer la variable MAIN\_NUMBER\_ID dans le fichier "index.js" par le nouvel ID.

**>** Par exemple pour le numéro Eurofins actuelle, on a cet URL "https://dashboard-v2.aircall.io/r/numbers/256235", l'ID sera donc 256235.

2. **le numéro SVI:** (Eurofins-SVI +33 4 85 80 00 97)

C'est le numéro de débordement vers lequel les clients seront redirigés en dehors des heures d'ouverture, et contient un répondeur vocal avec un message spécifique ainsi qu'il permet au client de composer le 1 pour contacter l'astreinte informatique. Sa logique est également modifiable directement depuis le tableau de bord d'Aircall si on a un compte administrateur.

**Remarque:** En cas de changement de ce numéro SVI, on doit modifier la variable "number" dans le fichier "index.js" par le nouveau numéro en annotation E164.

**>** Par exemple pour le numéro SVI actuelle on a sur le code "const body = {number: '+33485800097'};", il faut changer donc +33485800097.

**Webhook Aircall:**

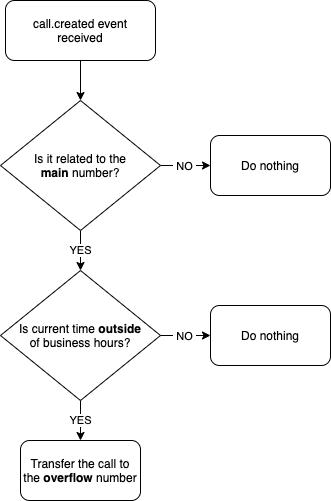
Les Webhooks sont un moyen sécurisé pour récupérer des données associées à un compte Aircall, en temps réel.

Pour notre cas, il sert à envoyer une requête POST vers notre serveur (nodeJS) dès qu'un appel vers le numéro Eurofins a lieu en temps réel, et donc permet de récupérer les données concernant l'heure exacte de l'appel et par la suite la rédiriger vers le numéro Eurofins-SVI en cas de besoin.

**Remarque:** En cas de changement de serveur, on doit modifier l'url sur le webhook, en accédant au tableau de bord Aircall, puis sur la partie "Integrations", on clique sur "configurer" devant le webhook, et on met le nouveau URL avec "/aircall/calls" à la fin.

**Implémentation de la logique de redirection:**

Le fichier "index.js" contient la logique permettant de rediriger les appels entrants effectués en dehors des heures d'ouverture vers le numéro SVI selon le schéma suivant:



Chaque fois que notre serveur Web recevra un événement depuis le webhook, nous pouvons vérifier l’heure actuelle du serveur et prendre une décision en conséquence.

**Remarque:** Pour changer les horaires de disponibilité du numéro principale, il suffit de modifier les valeurs de variable businessHours, avec "from:" l'heure de début, et "to:" l'heure de fin, et chaque ligne pour un jour de la semaine.

**>** **Attention !** les heures sont marquées en fuseau horaire **GMT**, il faut adapter les modifications à ce fuseau par conséquent.

Pour que notre serveur soit lié à notre compte Aircall, et donc à l'API Aircall, on a un ID et un TOKEN qu'on spécifie à notre code, et qui sont générés par le tableau de bord, sous la partie "Compte", puis "Clé API", puis "Générer un clé API".

**Remarque:** En cas de changement du compte administrateur, il faut générer un nouveau ID et TOKEN d'API, et modifier les variables apiId et apiToken par les nouvelles valeurs respectivement dans le fichier "index.js".

**Implémentation du serveur:**

Il faut préalablement installer les dépendances spécifié sur le fichier package.json :

$ npm install

Ensuite, en cas de changements survenus sur le code, il faut l’envoyer vers le repository git:

$ git add .

$ git commit -m « mon message »

$ git push

Et l’envoyer vers l’hébergeur, qui sera Heroku pour notre cas :

$ git push heroku main