

# Rapport de Stage Eurotunnel

## Table des matières

Présentation de l'entreprise.....	2
Quelques chiffres clés : .....	2
WIFI.....	3
Présentation du WIFI :.....	3
Fréquence.....	3
Comment ça marche ? .....	3
Portée.....	3
Tableau présentant les différents WIFI.....	4
Architecture WIFI type .....	5
VLAN.....	5
Trame tag/untag ou Trunk/Access.....	5
Mise en place d'un réseau WIFI pour une sortie internet hors plateforme Meraki.....	7
Monitoring API Meraki.....	10
Definition d'une API .....	10
Script API sous PowerShell .....	10

## Présentation de l'entreprise

Getlink est un acteur majeur des infrastructures de mobilité, des échanges internationaux et un leader du transport éco-responsable en Europe.

Getlink s'attache au quotidien à faciliter les échanges commerciaux, à soutenir les activités économiques entre le Royaume Uni et l'Europe continentale et à créer de la valeur pour toutes ses parties prenantes, en rapprochant les personnes, les entreprises et les cultures.

Le groupe Getlink est composé de quatre entités :

- **Eurotunnel**, leader du transport transmanche de véhicules de tourisme (Le Shuttle), de camions (Le Shuttle Freight), des passagers Eurostar et des trains de fret ferroviaire qui empruntent le tunnel sous la Manche
- **Europorte**, premier opérateur privé de fret ferroviaire en France
- **ElecLink**, l'interconnexion électrique de 1 GW entre le Royaume-Uni et la France
- **CIFFCO**, le centre de formation privé dédié aux formations du ferroviaire.

### Quelques chiffres clés :

- Plus de **32 millions de camions** transportés à bord des Navettes Le Shuttle Freight depuis 1994.
- Plus de **60 millions de voitures** transportées à bord des Navettes Le Shuttle depuis 1994.
- Plus de **465 millions de passagers** ont emprunté le tunnel sous la Manche depuis 1994 à bord des Navettes Eurotunnel et des trains Eurostar.
- **26 % des échanges commerciaux** transitent chaque année entre le Royaume-Uni et l'Europe continentale via le tunnel sous la Manche.
- **2 milliards de tonnes-kilomètres** pour Europorte par an.

## WIFI

### Présentation du WIFI :

WIFI (Wireless Fidelity) : Transmission de données sans fils au sein d'un réseau informatique grâce à un ensemble de protocoles de communication définis par l'IEEE 802.11X.

Chaque norme IEEE fonctionne sur différentes fréquences, offrant une bande passante différente.

Les données sont transmises par radiofréquence. La portée du signal varie selon l'environnement (indoor/outdoor et la présence d'obstacles auquel il est confronté).

Il existe différents types de wifi :

- WIFI 4 (IEEE 802.11n)
- WIFI 5 (IEEE 802.11ac)
- WIFI 6 (IEEE802.11ax)
- WIFI 7 (IEEE802.11be)

### Fréquence

Les données sont transmises par radiofréquence quand on utilise le WIFI. Les principales fréquences utilisées pour cette technologie sont les bandes 2.4 GHz et 5GHz, le WIFI 7 utilise également les bande 6 GHz. Les fréquences basses portent plus loin le signal cependant elles offrent moins de bande passante que les hautes fréquences, ces fréquences sont des UHF (Ultra High frequency).





### Comment ça marche ?

Le WIFI fonctionne grâce à un Access Point et un appareil (ordinateurs, téléphone, tablette...) équipé d'un adaptateur réseau sans fil (carte wifi) afin de convertir les données envoyées en signal radio, les données seront ensuite transmises de l'AP à son commutateur sur lequel il est relié par câble Ethernet.

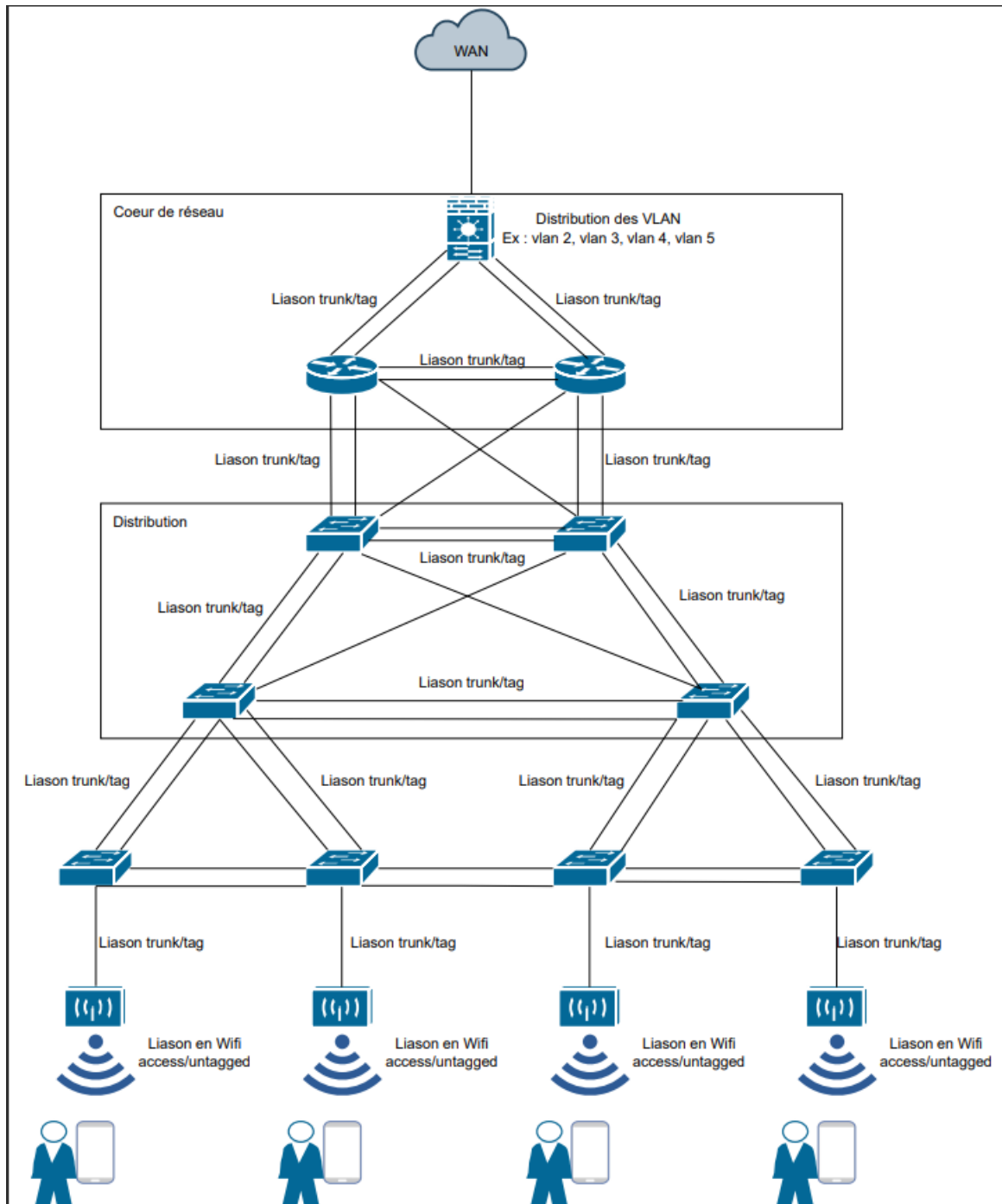
### Portée

La portée du wifi dépend de l'environnement sur laquelle, elle est implantée. La portée du wifi dépend également de la fréquence utilisée en effet si nous utilisons du wifi en 5 GHz la portée sera d'environ 15 mètres, si nous utilisons du 2.4 GHz la portée sera de 20 mètres. En revanche plus la fréquence est réduite plus la bande passante est réduite.

## Tableau présentant les différents WIFI

	 Wi-Fi 4 (IEEE 802.11n)	 Wi-Fi 5 (IEEE 802.11ac)	 Wi-Fi 6 (IEEE 802.11ax)	 Wi-Fi 7 (IEEE 802.11be)
Bande passante du canal (en MHz)	20, 40	20, 40, 80, 80 + 80, 160	20, 40, 80, 80 + 80, 160	20, 40, 80, 80 + 80, 160, 320
Bandes de fréquence (en GHz)	2.4, 5	5	2.4, 5, 6	2.4, 5, 6
Modulation	64-QAM	256-QAM	1024-QAM	4096-QAM
Nombre de flux avec différentes positions dans l'espace	1	4	8	Jusqu'à 16
Débit maximum	150 Mbps	3.5 Gbps	9.6 Gbps	> 30 Gbps

## Architecture WIFI type



## VLAN

Un VLAN (Virtual Local Area Network) est un groupe de périphérique connectés logiquement avec toutes les fonctionnalités d'un LAN. Cependant les périphériques d'un VLAN n'ont pas besoin d'être connectés physiquement ou sur le même réseau pour fonctionner contrairement au LAN. En général, les VLAN sont utilisés pour segmenter un réseau, le sécuriser et le faire évoluer.

## Trame tag/untag ou Trunk/Access

Pour identifier les flux de chaque VLAN, on tague la trame Ethernet en ajoutant un identifiant de (1 à 4094) afin d'identifier à quel appareil correspond le trafic.

Si une trame Ethernet est « Tagguée » elle est alors identifiée dans un VLAN, c'est le switch qui va ajouter l'identifiant dans la trame et le switch pourra alors reconnaître l'appartenance à son VLAN et rediriger correctement le trafic. Cela a pour bût de reconnaître l'appartenance au bon réseau (au bon virtual LAN). Quand le port est en untagged, il partage toutes les trames alors que lorsqu'il est en en tagged, il partage uniquement les tags spécifiés.

Tagged : le port du switch envoie le trafic sans avoir retiré le tag du VLAN

Untagged : le port du switch envoie le trafic après avoir retiré le tag du VLAN.

## Mise en place d'un réseau WIFI pour une sortie internet hors plateforme Meraki

(Répondre aux incidents et aux demandes d'assistance et d'évolution : Traiter des demandes concernant les services réseau et système, applicatifs).

(Mettre à disposition des utilisateurs un service informatique : Réaliser les tests d'intégration et d'acceptation d'un service, Accompagner les utilisateurs dans la mise en place d'un service).

(Travailler en mode projet : Analyser les objectifs et les modalités d'organisation d'un projet).

Mission demandée : Fournir une connectivité internet en dehors de la plateforme cloud de meraki, afin de libérer un SSID global et utiliser un SSID.

On m'a fourni un Access Point MR42 de chez Cisco afin de pouvoir fournir du wifi de la LiveBox.

Le switch fournis nous sert juste à alimenter l'AP à l'aide du Po



Sur la plateforme Meraki on peut voir nos différents SSID que les bornes wifi diffusent sur tous le site :

Network

Eurotunnel

Network-wide

Wireless

Insight

Two-factor Authentication is not currently enabled on your Meraki

### Configuration overview

SSIDs Showing 15 of 15 SSIDs. [Hide disabled SSIDs](#)

	SSID 1	SSID 2	SSID 3	SSID ALEX
Enabled	<a href="#">enabled</a>	<a href="#">enabled</a>	<a href="#">enabled</a>	<a href="#">enabled</a>
Name	<a href="#">rename</a>	<a href="#">rename</a>	<a href="#">rename</a>	<a href="#">rename</a>
Access control	<a href="#">edit settings</a>	<a href="#">edit settings</a>	<a href="#">edit settings</a>	<a href="#">edit settings</a>
Encryption				
Sign-on method				
Bandwidth limit				
Client IP assignment				
Clients blocked from using LAN				
Wired clients are part of Wi-Fi network				
VLAN tag				
VPN				
<a href="#">Splash page</a>				
Splash page enabled				
Splash theme				

Voici l'état de mon AP :

Network

Eurotunnel

Network-wide

Wireless

Insight

test\_alex

MR42

Formation HACCP & Permis d...

Siege Exploitation Eurotunnel

Google

Map data ©2024 Google

ADDRESS

Siege Exploitation, Coquelles

SSIDs

Summary

Event log

Timeline

Location

Connections

Performance

Tools

LAN

### Live data

Current clients 0

There are no currently connected clients.

Current mesh routes 0

Route	Avg Mbps	Avg metric 0	Usage
test_alex >	75.3	2872	100%

Radius and VLAN request status 0

No RADIUS or VLAN data available for this node.

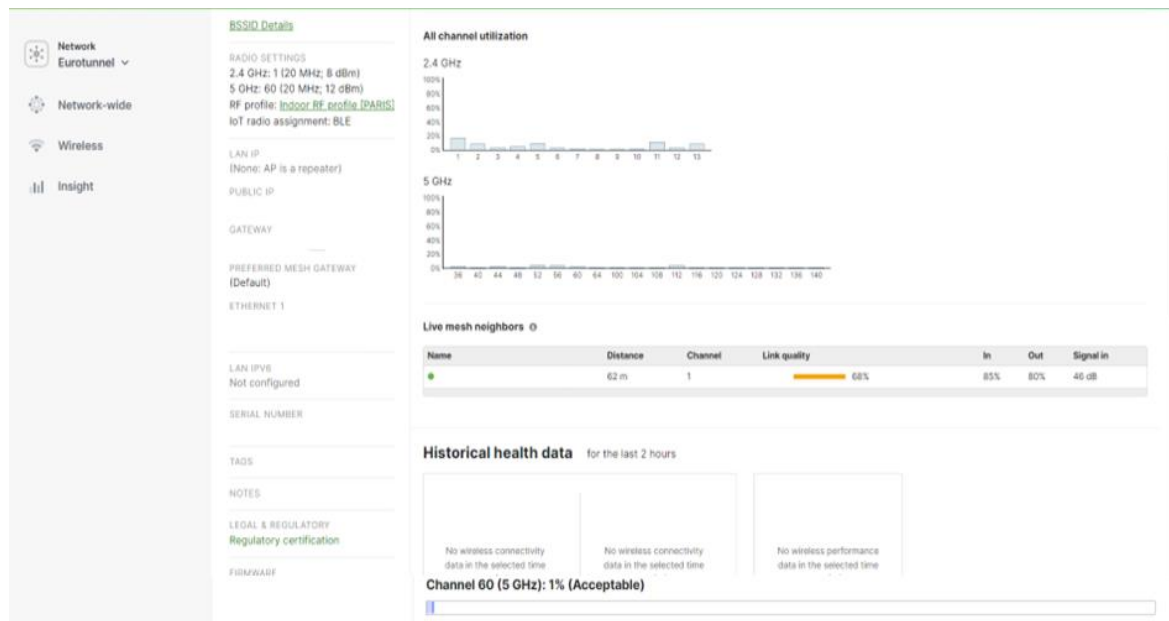
Utilization on current channels

802.11 Traffic non-802.11 Interference

Channel 1 (2.4 GHz): 12% (Acceptable)

Channel 60 (5 GHz): 1% (Acceptable)





Suite à plusieurs recherches et en se documentant sur les points d'accès MR42, j'ai remarqué que ces AP sont administrables uniquement sur la plateforme Meraki de façon centralisée sur un cloud et qu'ils fonctionnent en mesh c'est-à-dire qu'ils détectent les AP autour d'eux et va donc se connecter directement aux autres AP en chargeant leurs configurations quand bien sûr ils sont préalablement déjà paramétrés dans le réseau.

À la suite de cela on remarque que l'on ne peut pas fournir du wifi de la Box internet sans gagner 1 SSID car ses AP MR42 sont faits pour qu'ils fonctionnent en mesh et que par la plateforme Meraki.

Solution envisagée pour remédier à cela : Acheter un autre point d'accès d'une autre marque et qui fonctionne hors technologie mesh et de le brancher directement sur la box internet.

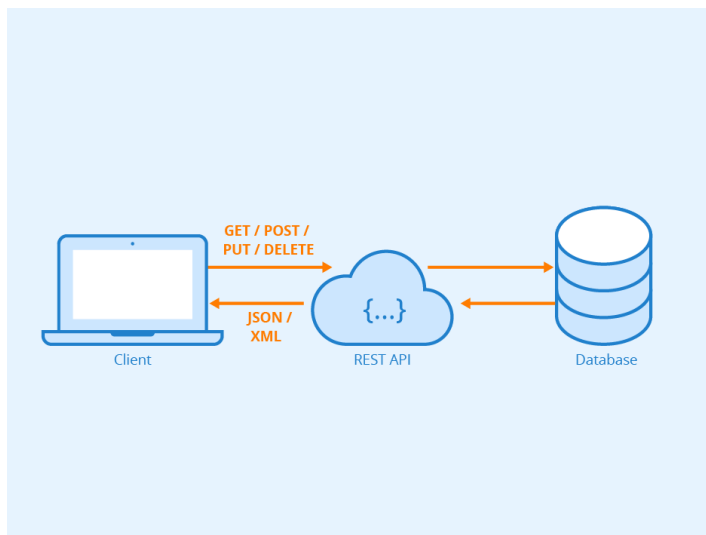
## Monitoring API Meraki

(Gérer le patrimoine informatique : Vérifier les conditions de la continuité d'un service informatique).

### Definition d'une API

Une API (*application programming interface* ou « interface de programmation d'application ») est une interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.

Une API REST est une interface de programmation d'application (API) qui permet d'établir une communication entre plusieurs logiciels. Grâce à elle, des logiciels d'applications utilisant différents systèmes d'exploitation peuvent interagir et partager des informations par l'intermédiaire du protocole HTTP.

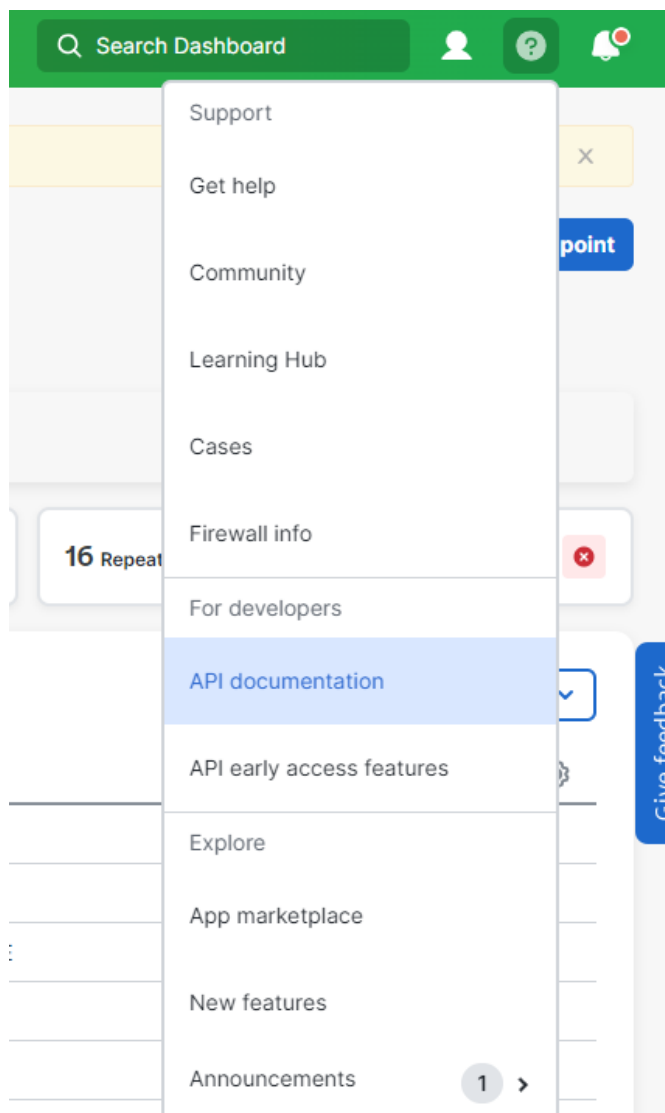


### Script API sous PowerShell

Mission demandée : Créer un script PowerShell afin d'envoyer des requêtes url « GET » pour pouvoir obtenir des informations sur les différents statuts des Access points et de leurs Nom ainsi que de savoir combien d'AP sont présents dans chaque tag. La finalité de cette mission a pour bût que les épiques RUN puissent avoir une visualisation sur les problèmes des points d'accès du site directement sur leur outil de monitoring.

Je dois créer ce script pour des Access points utilisant la plateforme Cloud Meraki (cisco). Pour pouvoir réaliser cela, je me suis renseigné directement sur la documentation meraki afin de savoir quelle requête effectuer :

Tout d'abord, afin d'obtenir la documentation API sur meraki, il faut se connecter à la plateforme et cliquer sur API documentation :



Une fois cliqué sur cela on peut voir sur leur documentation des exemples d'url :

Par exemple, ici cette url c'est pour voir les différents historiques d'alertes :

The screenshot shows the Cisco DevNet API reference page for the 'Get Network Alerts History' endpoint. The page is divided into several sections:

- API Reference**: Overview, API, Platform, configure, liveTools, monitor, adaptivePolicy, administered, alerts, history.
- Get Network Alerts History**: Operation ID: getNetworkAlertsHistory, Description: Return the alert history for this network. The endpoint is `GET /networks/{networkId}/alerts/history`.
- Request Parameters**: Path, Query.
- Path**: `networkId` (required) | string. Network ID.
- Query**: `perPage` | integer. The number of entries per page returned. Acceptable range is 3 - 1000. Default is 100. `startingAfter` | string. A token used by the server to indicate the start of the next page. When this is a timestamp or an ID.
- Configuration**: Parameters, Code Snippets.
- Code Snippets**: Meraki Python Library, Curl, Python, Nodejs.

The Python code snippet is as follows:

```
import requests

url = "https://api.meraki.com/api/v1/networks/{networkId}/alerts/history"

payload = None

headers = {
    "Authorization": "Bearer 75d45334bef4d2bc96f26138c163c0a3fa0b5ca6",
    "Accept": "application/json"
}

response = requests.request('GET', url, headers=headers, data=payload)

print(response.text.encode('utf8'))
```

On remarque que pour la requête s'effectue bien on n'a besoin d'un token (cle\_API), pour pouvoir avoir cela, il faut se rendre sur la plateforme meraki puis cliquer sur my profile :

The screenshot shows the Meraki dashboard user profile dropdown menu. The menu is open, showing the user's email address, customer ID, and options to view the profile or sign out.

- Search Dashboard
- alexandre.lespagnol@getlinkgroup.com
- Customer 7801-3514
- My profile
- Sign out

Et ensuite cliquer sur Generate new API key :

**Your account**

View your account settings.

**Change your password** ⓘ

Your new password will be synced to all Dashboard accounts with this email address.

Current password:

New password:

Confirm password:

[Change password](#)

**Two-factor authentication**

Two-Factor authentication is **OFF**. [Set up two-factor authentication](#)

**API access**

API keys

Key	Created at	Last used	
*****3c85	Jan 31 2024 09:37 UTC	Feb 16 2024 15:08 UTC	<a href="#">Revoke</a>

[Generate new API key](#)

**Color blind assist mode (OFF)**

Enables an alternative color palette for various Dashboard elements.

[Enable Red/Green assist mode](#)

**Sample of elements affected by color blind assist mode:**

Device status icons: Active: Alerting: Unreachable: Dormant:

Map pins: Gateway Repeater Alerting Offline

Connectivity:

Connectivity icons:

Labels: Success Alert

**Dashboard language** BETA

Preferred language:  [Save](#)

Ensuite une fois notre clé API récupéré, pour pouvoir faire des requêtes, il faut le network id de l'entreprise ainsi que l'organisation id :

Pour pouvoir obtenir ses informations on peut procéder de deux façons différentes, la première est de faire une requête url directement afin d'obtenir les informations et l'autre est d'aller dans la page source de la page de la plateforme et rechercher le mkiconf.org\_id pour le org\_id et le mkiconf.ng\_id pour le network\_id :

```
Mkiconf.org_id = " ";
Mkiconf.org_eid = " ";
Mkiconf.org_name = "GETLINK";
Mkiconf.locale_id = null;
Mkiconf.ng_id = " ";
Mkiconf.ng_eid = " ";
Mkiconf.network_name = "Eurotunnel";
Mkiconf.base_url = " ";
Mkiconf.shard_origin_url = "https: ";
Mkiconf.is_federal_cluster = false;
Mkiconf.min_password_length = 8;
```

Attention, pour le Network id, il faut rajouter « N\_(net\_id) »

Voici le script que j'ai créé pour pouvoir avoir les différents statuts des access points offline et en alertes :

```

D:\> lespagnola > stage > API > & .\apimerakitotal.ps1
1  <#
2  .DESCRIPTION
3  Recherche des points d'accès par tags et renvoie ceux en statut "alerting" ou "offline".
4  .PARAMETER tag
5  Spécifie les tags que vous souhaitez utiliser pour filtrer les points d'accès.
6  .PARAMETER apexclus
7  Spécifie les points d'accès à exclure (séparés par des virgules).
8  .EXAMPLE
9  C:\PS> apimerakitotal.ps1 -apexclus "api,ap2" -tag "tag1"
10 C:\PS> apimerakitotal.ps1 -apexclus "api" -tag "tag1,tag2"
11 C:\PS> apimerakitotal.ps1 -apexclus "api,ap2" -tag "tag1,tag2"
12 C:\PS> apimerakitotal.ps1 -apexclus "" -tag "tag1,tag2,tag3"
13 C:\PS> apimerakitotal.ps1 -apexclus "" -tag "tag1"
14 .NOTES
15 Auteur: Alexandre LESPAGNOL
16 Date: 07 février 2024
17 #>
18 # Demander à l'utilisateur de saisir le tag et les points d'accès à exclure
19 param(
20     [Parameter(Mandatory=$true)]
21     [String] $tag,
22     [String] $apexclus
23 )
24 # Déclaration de la fonction pour pouvoir l'appeler plus tard dans le script
25 function Get-OfflineAlertingAps{
26     param(
27         [Parameter(Mandatory=$true)]
28         [String] $tag,
29         [String] $apexclus
30     )
31     # Définition de l'URL de l'API Meraki
32     $url = "https://api.meraki.com/api/v1/organizations/org_id/devices/statuses/"
33     # Définition de l'en-tête de la requête HTTP
34     $headers = @{
35         "Authorization" = "Bearer cle-api"
36         "Accept" = "application/json"
37     }
38     # Envoi de la requête GET et stockage de la réponse
39     try {
40         $response = Invoke-RestMethod -Uri $url -Headers $headers -Method Get -ErrorAction Stop
41         # Vérification si la réponse est vide
42         if (-not $response) {
43             Write-Host "La réponse de l'API est vide."
44         } else {
45             # Convertir la réponse en JSON en utilisant la méthode ToJson de PowerShell
46             $jsonString = $response | ConvertTo-Json
47             # Convertir la chaîne JSON en objet PowerShell
48             $data = $jsonString | ConvertFrom-Json
49
50             # Filtrer les accès points dans le tag spécifié
51             $devices_tag = $data | Where-Object { $_.tags -contains $tag }
52             # Filtrer les accès points exclus
53             if ($apexclus) {
54                 $excluded_aps = $apexclus.Split(',')
55                 $devices_tag = $devices_tag | Where-Object { $excluded_aps -notcontains $_.name }
56             }
57             # Filtrer les accès points hors ligne ou en alerte dans le tag spécifié
58             $offline_alert_tag = $devices_tag | Where-Object { $_.status -eq 'offline' -or $_.status -eq 'alerting' }
59             # Afficher les résultats pour le tag spécifié
60             Write-Host "Nombre total d'accès points dans le tag (0): (1)" -f $tag, $devices_tag.Count
61             Write-Host "Nombre total d'accès points hors ligne ou en alerte dans le tag (0): (1)" -f $tag, $offline_alert_tag.Count
62             Write-Host "Access points hors ligne ou en alerte dans le tag : " -f $tag
63             $offline_alert_tag | ForEach-Object { Write-Host ("Nom: {0}, Statut: {1}" -f $_.name, $_.status) }
64             Write-Host "-----"
65             # Retourner les résultats pour concaténation
66             return [PSCustomObject]{
67                 NumDevicesTag = $devices_tag.Count
68                 NumOfflineAlertTag = $offline_alert_tag.Count
69                 OfflineAlertTag = $offline_alert_tag
70                 All = $devices_tag
71             }
72         }
73     } catch {
74         # Si problème pour récupérer les données depuis l'API cela retourne :
75         Write-Host "Une erreur s'est produite lors de la récupération des données depuis l'API : $_"
76     }
77 }
78 # Initialiser un dictionnaire pour stocker les résultats uniques
79 $uniqueResults = @{}
80 # Initialiser le compteur pour le nombre total d'accès points
81 $total_devices = 0
82 # Initialiser le compteur pour le nombre total d'accès points hors ligne ou en alerte
83 $total_offline_alert = 0
84 # Appel de la fonction et séparation des tags par une "," si plusieurs tags sont entrés dans le terminal
85 $splitted_tags = $tag -split ","
86 foreach ($different_tag in $splitted_tags){
87     # Ajouter les résultats de chaque tag au dictionnaire $uniqueResults
88     $tagResults = Get-OfflineAlertingAps -apexclus $apexclus -tag $different_tag
89     # Ajouter le nombre d'accès points dans le tag spécifié au compteur total
90     $total_devices += ($tagResults.All | Where-Object { $_.Name -notin $uniqueResults.Keys }).Count
91     # Ajouter le nombre d'accès points hors ligne ou en alerte dans le tag spécifié au compteur total
92     $total_offline_alert += ($tagResults.OfflineAlertTag | Where-Object { $_.Name -notin $uniqueResults.Keys }).Count
93     foreach ($result in $tagResults.OfflineAlertTag) {
94         if (not $uniqueResults.ContainsKey($result.Name)) {
95             $uniqueResults[$result.Name] = $result
96         }
97     }
98 }
99 # Ajouter les résultats uniques à la variable $results
100 $results = $uniqueResults.Values
101 # Afficher les résultats concaténés pour tous les tags
102 Write-Host ("Résultats concaténés pour tous les tags rentre en parametre :)")
103 if ($results.Count -eq 0) {
104     Write-Host "Aucun access point hors ligne ou en alerte trouvé."
105 } else {
106     $results | ForEach-Object { Write-Host ("Nom: {0}, Statut: {1}" -f $_.name, $_.status) }
107 }
108 # Afficher le nombre total d'accès points sur tous les tags
109 Write-Host "Nombre total d'accès points sur tous les tags rentre en parametre : $total_devices"
110 # Afficher le nombre total d'accès points hors ligne ou en alerte sur tous les tags
111 Write-Host "Nombre total d'accès points hors ligne ou en alerte sur tous les tags rentre en parametre : $total_offline_alert"
112 Write-Host "-----"

```

Résultat du script :

[illegible]

Afin que les épiques RUN puissent superviser cela, il faut modifier le script pour qu'il soit intégrable dans l'outil de supervision :

(script ou diagramme du script)

Après interfaçage du script dans l'outil de supervision, voici les résultats :

Service Check\_WIFI access point PAX on [REDACTED]meraki.com

Service

WIFI\_PAX

Host

[REDACTED]meraki.com (158.115.144.221)

Current Status

CRITICAL

(for 0d 0h 21m 4s)

Nom: AP\_UKF\_Pos23\_5, Statut: offline, Last alarm date: 20/02/2024 on 04:19:42

Nom: AP\_FRF\_FrontTools\_2, Statut: offline, Last alarm date: 22/02/2024 on 12:54:14

Nom: AP\_UKF\_U51\_PAFLanes\_3, Statut: offline, Last alarm date: 19/02/2024 on 13:52:49

Nom: AP\_UKF\_Pos23\_4, Statut: offline, Last alarm date: 20/02/2024 on 04:19:42

Nom: AP\_UKF\_Pos23\_3, Statut: offline, Last alarm date: 20/02/2024 on 04:19:42

On remarque ici que l'outil nous ressort bien les noms des points d'accès offline, il nous ressort également le niveau de criticité en fonction des paramètres rentré dans le script.

Autre résultat pour différents afin de pouvoir superviser les points d'accès en fonction de leurs zones.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
neraki.com	WIFI_CO	OK	14:59:47	0d 0h 18m 28s	1/3	OK - 38 APs good
	WIFI_FR	OK	15:07:09	0d 0h 2m 14s	1/3	OK - 540 APs good
	WIFI_Fed	OK	14:59:47	0d 0h 18m 28s	1/3	OK - 234 APs good
	WIFI_I	OK	15:09:20	0d 0h 0m 3s	1/3	OK - 209 APs good
	WIFI_Swec	OK	14:59:27	0d 0h 18m 28s	1/3	OK - 42 APs good
	WIFI_Swv	OK	14:59:11	0d 0h 18m 28s	1/3	OK - 1 APs good
	WIFI_UK	OK	15:09:20	0d 0h 0m 3s	1/3	OK - 248 APs good
	WIFI_Maintenance	OK	14:59:34	0d 0h 18m 28s	1/3	OK - 157 APs good
8 of 8 Items Displayed						

