Fiche mémo · Leçon 31 · 28 min

Clés d'API, comptes de service et droits d'accès d'un flux

Un flux tient des clés : celle de l'IA, l'accès au tableur, à la boîte de réception, l'adresse de son webhook. Rangez chaque secret dans les identifiants de l'outil, jamais dans un champ ni un tableur, faites-le détenir par un compte de l'entreprise, donnez-lui le moins de droits possible, plafonnez la dépense et protégez le webhook.

Objectifs

- Ranger chaque secret d'un flux dans les identifiants de l'outil et repérer ceux qui traînent ailleurs
- Attribuer chaque accès à un compte de l'entreprise avec le moins de droits possible et un plafond de dépense
- Protéger un webhook et conduire le remplacement d'une clé qui a fui

À retenir

- Un secret se saisit une fois, dans les identifiants de l'outil ; jamais dans un champ, une expression, un tableur, la fiche ou un export partagé.
- Les accès d'un flux qui doit durer appartiennent à un compte de l'entreprise, reconnectable par au moins deux personnes.
- Chaque flux a sa propre clé d'IA, avec un plafond de dépense et une date d'expiration, et chaque accès le moins de droits possible.
- Un webhook de production a une authentification, et son adresse ne figure dans aucune page publique.
- Une clé qui a fui se remplace et se révoque dans l'heure ; la conduite à tenir est écrite dans la fiche avant d'en avoir besoin.

Les quatre couches d'un accès

Le coffre
Le secret est saisi une fois dans les identifiants de l'outil, qui le chiffrent ou le masquent.

Le compte
L'accès appartient à un compte de l'entreprise, qui survit aux départs et aux congés.

Les droits
Le compte n'ouvre que ce dont le flux a besoin : un fichier, une boîte, un projet.

Le plafond
Une limite de dépense et une date d'expiration bornent les dégâts d'une fuite.

Une seule couche mal réglée suffit : une clé bien rangée mais sans plafond, ou plafonnée mais au nom d'une personne qui part.

Le code à recopier

Générer la clé de chiffrement, puis la déclarer dans le fichier .env BASH

```
# Génère 32 octets aléatoires, écrits en 64 caractères hexadécimaux.
# Testé le 15/11/2026 sur macOS ; même commande sous Linux.
openssl rand -hex 32

# Copiez le résultat dans le gestionnaire de mots de passe de l'entreprise,
# puis dans le fichier .env du serveur (jamais dans Git, jamais dans la fiche) :
#   N8N_ENCRYPTION_KEY=[les 64 caractères générés]
# Le fichier .env se protège en lecture :
chmod 600 .env
```

Changer la clé d'une instance qui a déjà des identifiants n'est pas anodin : faites-le sur une copie, et seulement à l'installation si possible.

Les erreurs fréquentes

L'erreur, et pourquoi elle coûte	La correction
Coller la clé du fournisseur d'IA dans l'en-tête d'un nœud HTTP Request, « en attendant de faire propre ». La clé est lisible par quiconque ouvre le workflow et part dans chaque export. Le provisoire dure, et la clé finit chez un prestataire ou dans une capture d'écran.	Créez dès le départ un identifiant du fournisseur, ou un identifiant « Header Auth », et appelez-le depuis le nœud. Si la clé a déjà été saisie en clair, remplacez-la.
Construire le flux avec son propre compte Google et sa propre clé d'IA, parce que c'est plus rapide. Le flux dépend d'une personne : il s'arrête quand elle part, change de mot de passe ou perd son accès, et il voit tout ce que son compte voit.	Faites détenir chaque accès par un compte de l'entreprise, avec les seuls droits utiles, et notez dans la fiche les deux personnes qui peuvent reconnecter.
Supprimer le message Teams où la clé a été collée, et considérer l'affaire close. Ceux qui ont lu le message ont pu la copier, et l'historique, les notifications ou une sauvegarde la gardent. Tant qu'elle n'est pas révoquée, elle fonctionne.	Créez une nouvelle clé, mettez-la dans l'identifiant, révoquez l'ancienne dans l'heure, puis relisez la consommation depuis la fuite.
Publier un webhook sans authentification et appeler son adresse depuis le code de la page du site. L'adresse est lisible par n'importe quel visiteur. Un robot peut remplir le tableur de fausses demandes, consommer les exécutions et faire appeler l'IA à vos frais.	Réglez « Header auth » sur le nœud Webhook, faites appeler l'adresse par le serveur du formulaire, et testez qu'une requête sans en-tête est refusée.