Connecter un service externe à Make ou n8n sans coder

Ce qui se branche seul en trente minutes avec Apify et RapidAPI, ce qui demande un vrai développement, et la checklist qui évite les erreurs classiques.

BLYX-AGENCY.COM · 3 AVRIL 2026 · PAR MELVIN BRETON

Un service externe se branche avec un seul module : HTTP dans Make, HTTP Request dans n8n. Tu y mets une adresse, une méthode (GET pour lire, POST pour agir) et une clé. Apify fournit des outils prêts à lancer qui vont chercher des données sur le web, RapidAPI un catalogue de services déjà branchables. Compte trente minutes pour le premier, dix pour les suivants.

1. Ce qu'une API permet vraiment de faire

Tu veux automatiser une tâche, et tu tombes sur le même mur que tout le monde : Make et n8n n'ont pas d'intégration toute faite pour le service qui t'intéresse. C'est exactement le trou que comble une API.

Une API, c'est une prise. Elle permet à deux logiciels de se parler sans que personne ne recopie quoi que ce soit à la main. Ton outil d'automatisation dit au service externe « donne-moi ça », « fais-moi ça », « envoie-moi ça », et le service répond.

Concrètement, une API tient en trois éléments : une adresse, une méthode et une clé. Le reste du vocabulaire est de la décoration.

80 % des services accessibles par API le sont via deux outils spécialisés : Apify et RapidAPI. C'est là que se joue l'essentiel de ce guide. Tu n'as pas besoin d'apprendre à coder pour couvrir cette part-là.

Une remarque avant de continuer. Savoir brancher n'est pas savoir quoi brancher. Si tu n'as pas encore classé tes tâches par gain réel, commence par l'article quelles tâches automatiser en premier, puis reviens ici.

2. Le vocabulaire à connaître en dix minutes

Sept mots suffisent. Tu les croiseras dans toutes les documentations, et ils veulent toujours dire la même chose.

Terme	Ce que ça veut dire
Endpoint	L'adresse exacte que tu appelles, par exemple `/search` ou `/verify`
GET	Je récupère des données, je ne modifie rien
POST	J'envoie des informations ou je déclenche une action
Headers	Les informations techniques de l'appel, dont ta clé
Query params	Les paramètres écrits dans l'adresse, par exemple `?q=hotel&ville=rennes`
Body	Les données envoyées dans un POST, presque toujours en JSON
Response	La réponse du service, presque toujours en JSON elle auss

Le JSON, c'est simplement du texte structuré en paires étiquette et valeur. Claude et n8n savent le lire seuls et te proposent ensuite chaque champ à la souris.

Dernier terme utile : le **rate limit**, la limite de requêtes autorisées par minute ou par mois. Elle décide de ce que ton automatisation coûtera réellement une fois lancée.

Tout appel, quel que soit le service, tient dans ce gabarit. Garde-le sous les yeux la première fois.

```
Adresse   : https://...
Méthode   : GET ou POST
Headers   : ta clé + le type de contenu
Params    : q=... (en GET)
Body      : { ...json... } (en POST)
```

3. Apify ou RapidAPI, lequel pour quoi

Les deux se ressemblent de loin. Ils ne servent pas au même moment.

	Apify	RapidAPI
Ce que tu y trouves	Des outils prêts à lancer, appelés Actors	Un catalogue de services déjà accessibles
À quoi ça sert	Aller chercher des données publiques sur le web	Faire faire un traitement à un service tiers
Exemple type	Récupérer le contenu propre d'un site	Vérifier une adresse email, l'… un PDF

	Apify	RapidAPI
Comment tu paies	À l'usage, selon la durée d'exécution	Par abonnement mensuel, souvent avec un palier offert

Un **Actor** Apify est un petit programme hébergé qui prend des instructions au format JSON, exécute une tâche, et te rend un résultat sous forme de tableau téléchargeable. Tu le trouves dans le catalogue Apify, tu le lances une fois à la main depuis leur interface, tu regardes ce qui sort, et seulement ensuite tu l'automatises.

Procédure vérifiée le 30/08/2026.

Côté RapidAPI, la mécanique est plus simple encore. Tu crées un compte, tu choisis un service, tu récupères ta clé, et tu envoies deux informations dans les headers de chaque appel :

```
X-RapidAPI-Key: TA_CLE
X-RapidAPI-Host: HOST_DU_SERVICE
```

Une fois ta clé créée, la fiche du service affiche l'appel tout prêt, à copier dans Make ou n8n. C'est ce qui rend ce catalogue confortable quand on débute.

> ***La limite à connaître avant de commencer.*** *Aller chercher des données publiques ne donne pas le droit de les exploiter. L'extraction automatisée de LinkedIn, des Pages Jaunes ou de Google Maps est hors sujet ici, et pas seulement pour des raisons de conditions d'utilisation : la CNIL a sanctionné en décembre 2024 une société française de 240 000 euros pour de la collecte de coordonnées sur LinkedIn. La ressource prospection*

LinkedIn avec Claude détaille ces deux lignes rouges et ce qui reste
[illegible]ent légal en prospection professionnelle.

4. Six connexions utiles à tester aujourd'hui

Voilà six branchements qui règlent des problèmes réels de TPE et PME. Cherche le nom indiqué dans le catalogue correspondant : les fiches changent d'éditeur et de tarif régulièrement, la recherche vieillit mieux qu'un lien.

Ce que tu cherches	Où	Ce que ça règle chez toi
Website Content Crawler	Apify	Récupérer le texte propre d'un site entier, pour alimenter une base de connaissance interne
`email verification`	RapidAPI	Nettoyer un fichier de contacts avant un envoi, et fiabiliser tes formulaires
`OCR`	RapidAPI	Lire le texte d'une facture scannée ou d'une photo, et sortir les montants
`google serp`	RapidAPI	Suivre ce qui sort sur ton marché et recevoir un résumé chaque matin
`currency exchange`	RapidAPI	Récupérer un taux de change du jour pour tes devis à l'export
`ip geo location`	RapidAPI	Router une demande entrante vers la bonne agence selon la région

Le catalogue des services offerts de RapidAPI est un bon terrain d'apprentissage : tu peux y faire tes premiers appels sans sortir la carte

bancaire.

Le réflexe qui fait gagner le plus de temps : teste toujours un appel à la main, dans l'interface du catalogue, avant de construire quoi que ce soit dans Make ou n8n. Tu sauras alors si le problème vient du service ou de ton montage.

Pour voir une de ces connexions en situation réelle, la veille concurrentielle automatisée applique exactement cette mécanique à la surveillance de tes concurrents.

5. Brancher une API dans Make

Procédure vérifiée le 30/08/2026.

Étape	Action
1	Crée un scénario
2	Ajoute le module **HTTP**, puis l'action **Make a request**
3	Colle l'adresse trouvée dans la documentation, et choisis GET ou POST
4	Ajoute les headers, ta clé en premier
5	Lance une exécution : Make affiche la réponse brute
6	Relie les champs de la réponse à ta destination, tableur, base ou messagerie

Le détail de chaque étape est dans le guide officiel Make.

Si tu passes par Apify, il existe plus court. Le module Apify officiel lance un Actor et récupère ses résultats sans que tu écrives une seule

adresse. La marche à suivre est dans la documentation Apify pour Make.

Un point que tout le monde découvre en production : Make facture à l'opération. Une automatisation qui traite mille lignes consomme mille opérations. Vérifie ce volume avant de brancher, pas après.

6. Brancher une API dans n8n

Procédure vérifiée le 30/08/2026.

Étape	Action
1	Crée un workflow
2	Ajoute le node **HTTP Request**
3	Choisis la méthode, GET ou POST
4	Colle l'adresse
5	Renseigne les headers et l'authentification
6	Exécute le node seul et lis la réponse
7	Envoie le résultat vers ta destination

La documentation officielle du node HTTP Request couvre les cas d'authentification particuliers, qui sont la seule vraie difficulté.

n8n a un avantage net sur Make quand le volume monte : la version que tu héberges toi-même ne facture pas à l'opération. Elle demande e

revanche quelqu'un pour la maintenir, ce qui n'est pas neutre dans une équipe sans personne technique.

7. La checklist avant de lancer

Sept vérifications. Elles couvrent l'essentiel des blocages, et elles prennent deux minutes.

✓	Vérification
☐	Je sais si l'appel est en GET ou en POST
☐	J'ai l'adresse exacte, pas seulement le nom du service
☐	J'ai ma clé, et je sais dans quel header elle se met
☐	Je connais la limite de requêtes et ce qui se passe au-delà
☐	Je sais où vont mes paramètres, dans l'adresse ou dans le body
☐	J'ai testé un appel à la main avant d'automatiser
☐	J'ai prévu quoi faire quand le service répond une erreur

La dernière ligne est celle qu'on saute, et c'est celle qui coûte le plus cher. Une automatisation qui échoue en silence pendant trois semaines fait plus de dégâts qu'une automatisation qui n'existe pas.

8. Les trois erreurs qui bloquent tout

Elles représentent 90 % des blocages des débutants, et elles se diagnostiquent en trente secondes chacune.

1. La mauvaise méthode. Tu envoies un GET là où le service attend un POST. Le service répond une erreur qui ne dit pas ça clairement. Relis la documentation : la méthode est toujours écrite en tête de la fiche.

2. La clé absente ou mal placée. Une clé collée dans l'adresse au lieu des headers ne sert à rien. Le code d'erreur 401 veut dire « je ne sais pas qui tu es », le 403 veut dire « je sais qui tu es et tu n'as pas le droit ».

3. Les paramètres au mauvais endroit. En GET, ils vont dans l'adresse. En POST, ils vont dans le body, au format JSON. Les mélanger produit une réponse vide, sans message d'erreur, ce qui est le cas le plus déroutant.

Un quatrième cas, plus rare mais qui fait perdre une soirée : la réponse arrive bien, mais ton outil ne trouve pas les champs. C'est presque toujours que le résultat est imbriqué dans un niveau supplémentaire. Regarde la réponse brute avant de conclure que le service ne marche pas.

9. Où le no-code s'arrête vraiment

Ce guide t'emmène loin, et il faut être honnête sur l'endroit où il s'arrête.

Le montage Apify plus RapidAPI plus Make ou n8n couvre bien les tâches en ligne droite : je récupère une donnée, je la traite, je la range quelque part. C'est déjà beaucoup, et c'est souvent tout ce dont une TPE a besoin la première année.

Trois signaux indiquent que tu as atteint le plafond.

Le volume. Quand le nombre d'opérations mensuelles grimpe, la facture à l'opération finit par dépasser le coût d'un développement fait une fois. Le point de bascule se calcule sur ta grille tarifaire, pas au jugé.

Les règles métier. Dès que ton automatisation doit arbitrer, croiser plusieurs sources, gérer des exceptions nombreuses, le scénario devient un plat de spaghettis que personne ne saura reprendre.

La criticité. Le jour où l'entreprise ne peut plus travailler si le scénario tombe, tu as besoin de journaux d'exécution, de reprises sur erreur et de tests. Aucun outil no-code ne te donne ça sérieusement.

L'arbitrage complet entre les deux approches est traité dans no-code ou sur mesure. Et si tu veux la carte de ce qui, chez toi, se branche seul et de ce qui demande un vrai développement, c'est précisément le travail d'un audit IA : on regarde tes outils, tes volumes et tes tâches, et on sort une liste priorisée plutôt qu'une intuition.

Pour la vue d'ensemble, côté méthode d'introduction de l'IA dans une entreprise, la ressource intégrer l'IA dans une PME pose le cadre dans lequel ces branchements viennent s'inscrire.

Questions fréquentes

Faut-il savoir coder pour suivre ce guide

Rien de technique. Il faut savoir lire une documentation en anglais, copier une adresse, et accepter de faire un premier test qui échoue. Ce qui bloque le plus souvent n'est pas la compréhension, c'est la peur de casser quelque chose. Un appel en GET ne casse rien : il lit.

Combien coûte réellement ce montage par mois

Apify facture à la durée d'exécution et offre un crédit mensuel qui suffit pour apprendre. RapidAPI propose presque toujours un palier offert, suffisant pour tester, puis des abonnements mensuels dont le montant est affiché sur la fiche de chaque service. Le vrai coût vient de ton outil d'automatisation : Make facture à l'opération, et c'est lui qui grimpe quand le volume arrive.

Que faire si le service que je vise n'est dans aucun des deux catalogues

Regarde d'abord si l'éditeur publie sa propre documentation développeur : la plupart des logiciels métier en ont une, et le module HTTP sait l'appeler exactement pareil. Si rien n'existe, c'est un signal fort. Un service sans accès par programme est un service dont tu ne sortiras pas tes données automatiquement, ce qui pèse lourd au moment de le remplacer.

Qu'est-ce qui est permis et qu'est-ce qui ne l'est pas

Consommer un service qui t'a donné une clé, sous les conditions qu'il affiche : permis, c'est le principe même de ces catalogues. Extraire automatiquement des données personnelles depuis une plateforme qui l'interdit dans ses conditions : interdit, sanctionné, et le sujet est traité dans la ressource sur la prospection LinkedIn. Entre les deux, la question à se poser est simple : est-ce que je pourrais expliquer ce montage à la personne dont je traite les données.

Comment garder ce montage en état sur la durée

Une automatisation branchée sur un service tiers casse toujours un jour. Le service change son adresse, modifie son format ou ferme. Prévois deux choses dès le départ, une alerte quand le scénario échoue, et une note écrite qui dit à quoi il sert et qui l'a monté. C'est la différence entre une automatisation qui tient trois ans et une automatisation que personne n'ose toucher six mois plus tard.
