

# Rapport IA n°7 - Projet n8n n°1 - Rapports Fondas à Partir de Canaux Telegram

25/02/26

(100% rédigé par Jared)

Salut les Taurus, j'espère que vous avez la patate pour cette nouvelle semaine.

ヾ(\*°▽°\*)

Aujourd'hui, on s'attaque à notre premier vrai projet n8n. Le but de faire un tutoriel pas à pas, pour que vous puissiez bien comprendre les mécanismes derrière cette automatisation, et surtout comprendre comment vous pouvez, à partir d'une idée, créer un vrai workflow de toute pièce avec n8n.

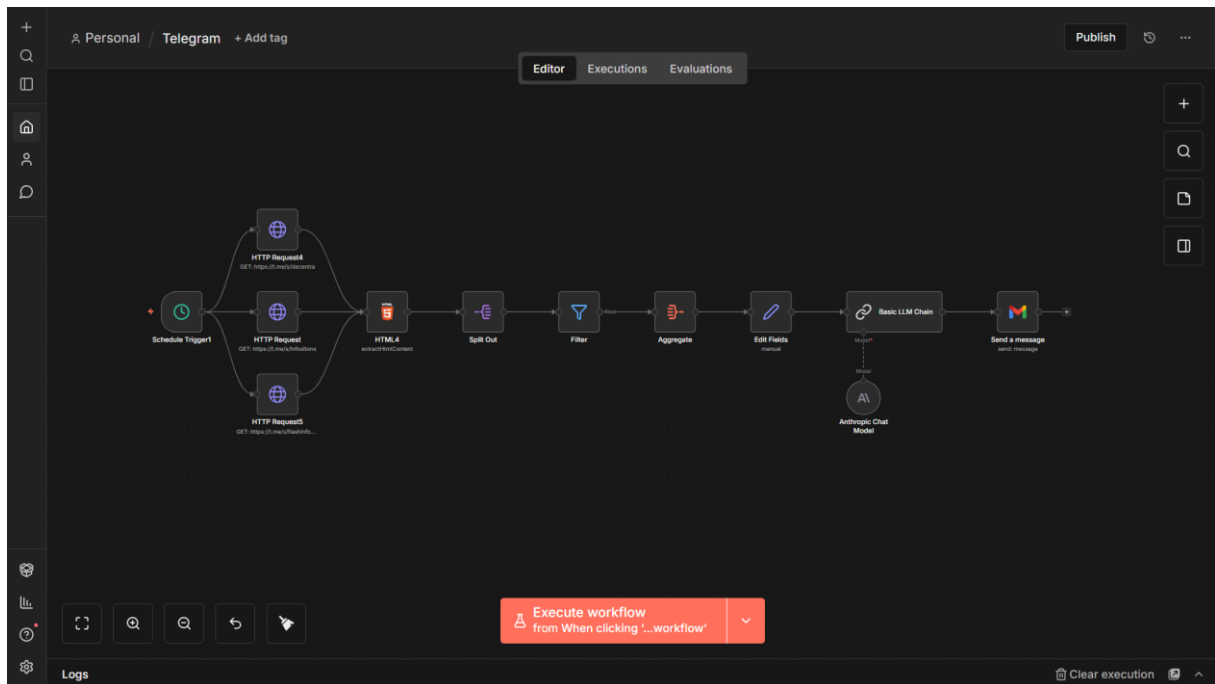
L'idée vient de l'apprenti (merci à lui :D), d'ailleurs si identifiez des tâches répétitives et chronophages dans votre trading, n'hésitez pas à passer en DM, ça fera peut-être l'objet d'un prochain projet >.<

Le problème identifié par l'apprenti est simple : il est très chronophage de suivre le fondamental, surtout quand on a plein de sources différentes à analyser. L'idée est donc de prendre ces sources, d'extraire les informations pertinentes, de traiter ces données, et enfin de les formater.

C'est un schéma classique de traitement de données, ce qui en fait un parfait exemple pour commencer. Dans mon exemple, je vais donc récupérer les actualités diffusées par 3 canaux Telegram, filtrer ces données, les traiter avec l'IA et enfin les formater sous la forme d'une Newsletter quotidienne que je recevrai automatiquement tous les jours à une heure prédéfinie dans ma boîte mail.

Evidement, ce choix est arbitraire et vous pouvez très bien choisir de vous sourcer ailleurs que Telegram, et de formater ça différemment. Il aurait été possible de créer un canal Telegram unique en sortie, dans lequel serait posté automatiquement un résumé d'autres canaux Telegram. Vous voyez que les possibilités sont quasiment infinies avec n8n, pourvu que l'application que vous voulez utiliser possède une API (donc quasiment toutes).

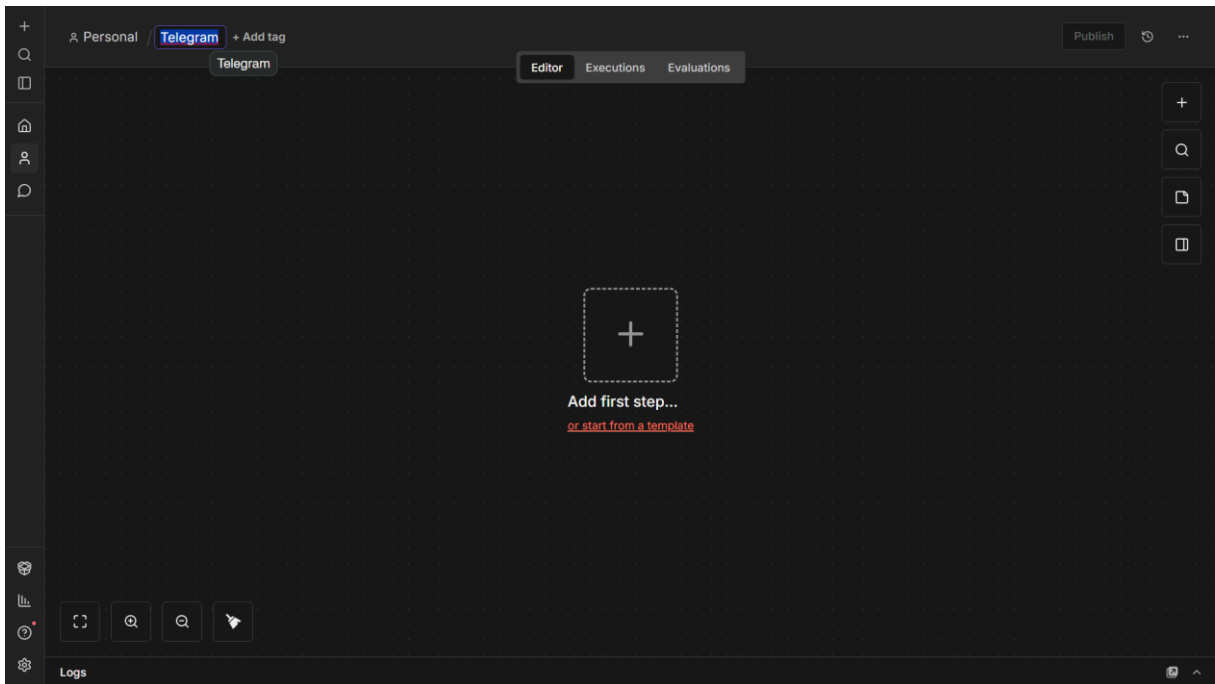
Voici la tête du workflow final (workflow = flux de travail = toute votre automatisation) :



On va tout voir en détail ne vous inquiétez pas. Petit point vocabulaire avant de commencer, chaque morceau du workflow que vous voyez s'appelle un nœud sur n8n. Le sens d'exécution est toujours de la gauche vers la droite, ce qui fait que le premier nœud de ce workflow est « Schedule Trigger », le nœud avec une image d'horloge verte. D'ailleurs comme indiqué dans le nom de ce nœud, il s'agit d'un trigger (déclencheur en français), donc le nœud qui va activer tout votre workflow.

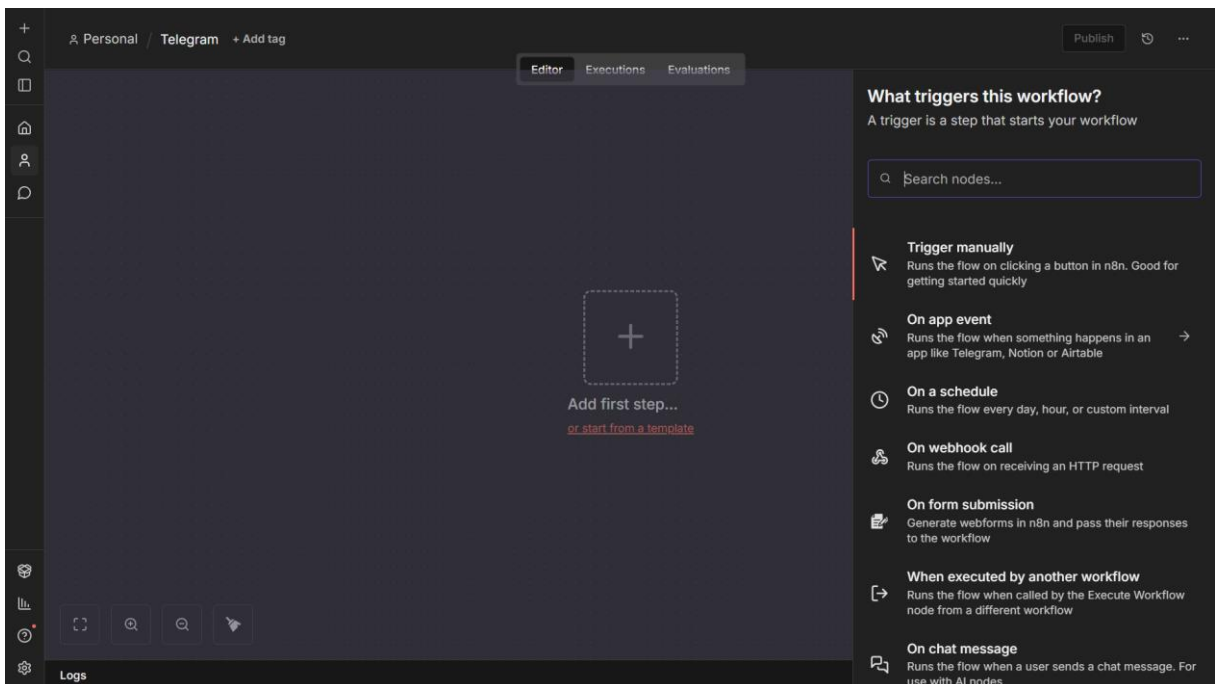
Une des particularités de n8n, c'est qu'on peut avoir plusieurs triggers pour un même workflow. On va en utiliser deux pour la phase de test, mais à la fin vous n'en aurez qu'un, qui déclenche le workflow de façon prédéfini (par exemple tous les jours à 8h), idéal pour une newsletter. Vous verrez que c'est un des triggers les plus utilisés.

Je fais exprès de vous embêter avec le vocabulaire dès le début car toutes les ressources que vous trouverez sur n8n utiliserons ces termes. Donc autant s'habituer dès le départ, surtout qu'il n'y a vraiment pas grand-chose : si vous comprenez ce qu'est un workflow et un nœud, vous avez fait 80% du job.



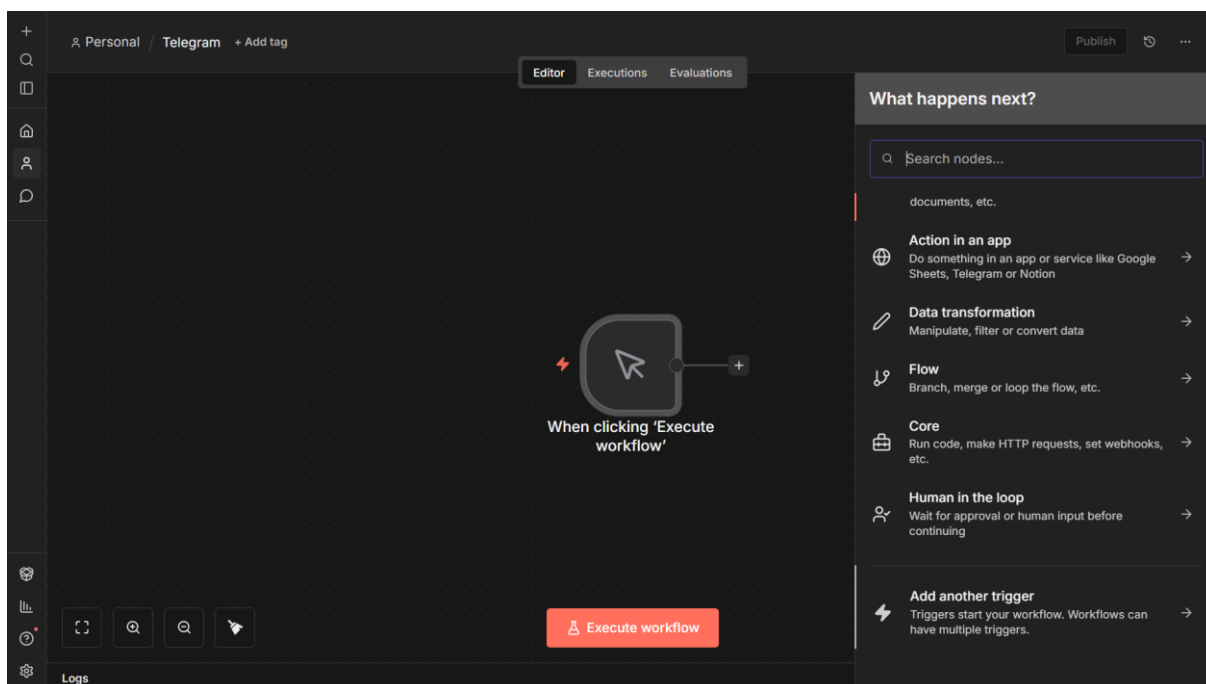
Vous allez démarrer avec un workflow vide, la première chose que je vous conseille de faire, c'est de le renommer en haut à gauche.

Ensuite vous avez un gros +, soit vous cliquez dessus, soit vous cliquez du Tab (désolé la team Mac mais je suis sans le sous UwU ).

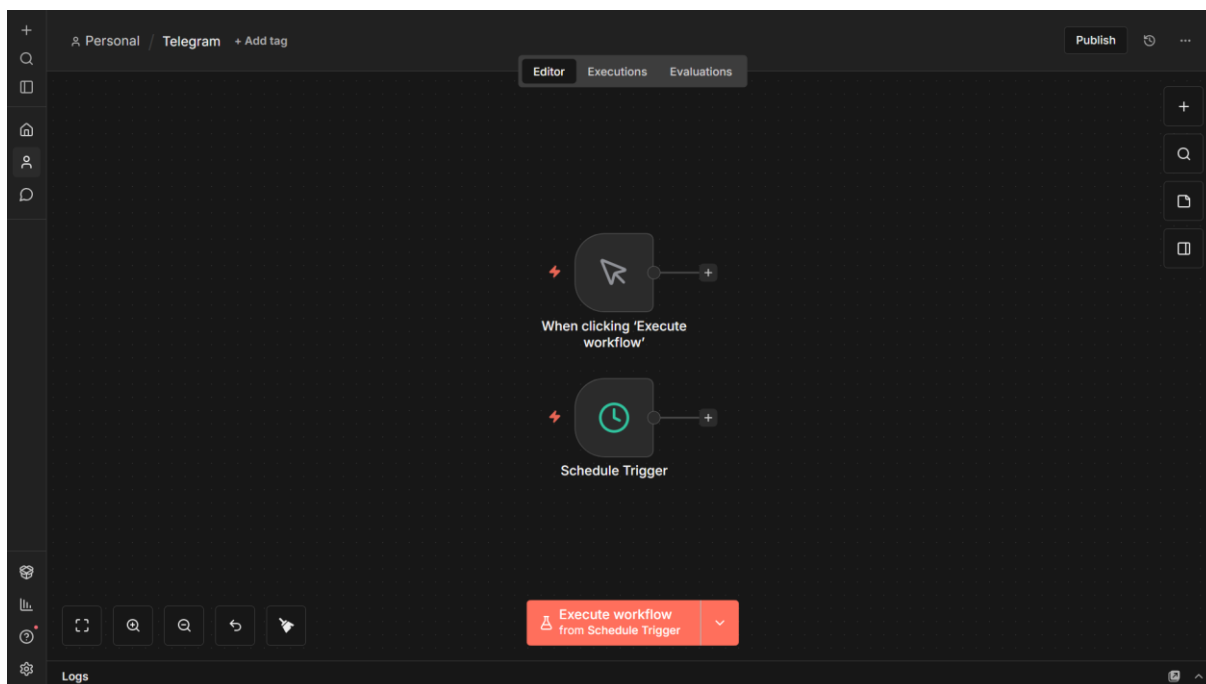


Vous avez ce menu qui s'affiche avec tous les triggers (déclencheurs). On va en prendre deux pour les tests :

- Trigger manually (déclenchement manuel) : idéal pour la phase de test
- On a schedule (pour la version finale) : déclencheur programmé



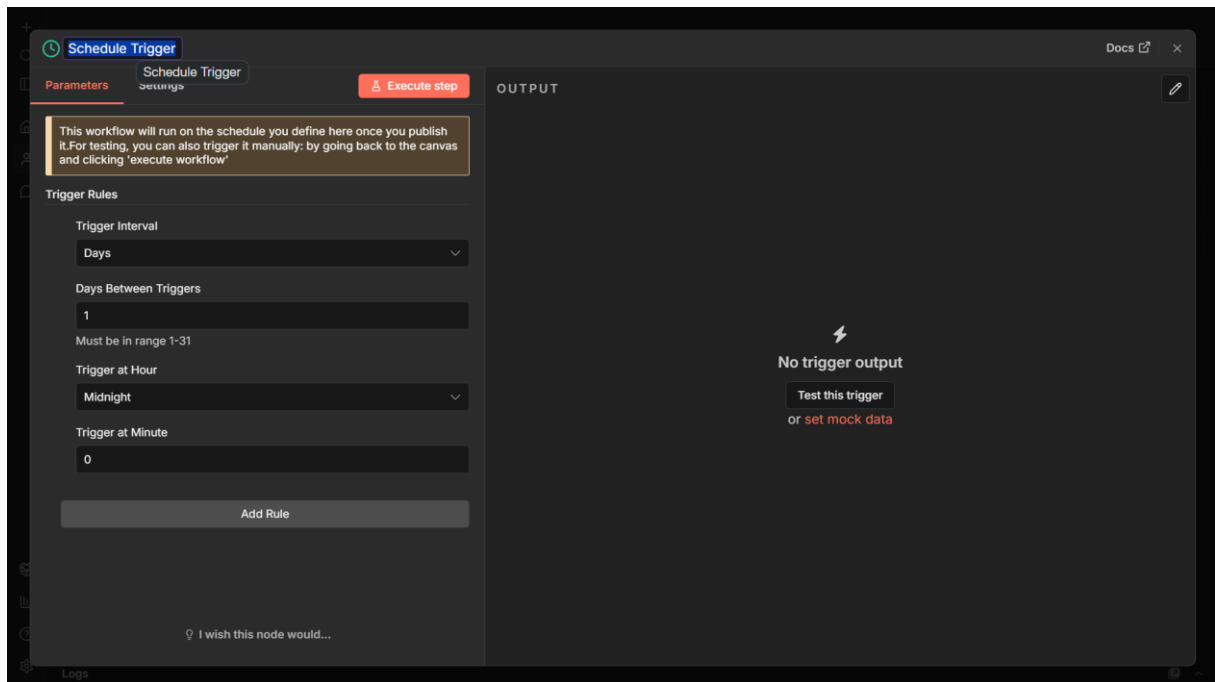
Quand vous avez ajouté un trigger, vous pouvez cliquer sur Tab pour avoir le menu, et cliquer sur Add another trigger pour avoir la même liste que tout à l'heure.



Vous avez maintenant vos deux triggers comme ça. Petite précision :

- Pour vous déplacer dans le workflow : appuyez sur la molette en même temps de bouger la souris
- Pour zoomer/dézoomer : Ctrl + molette

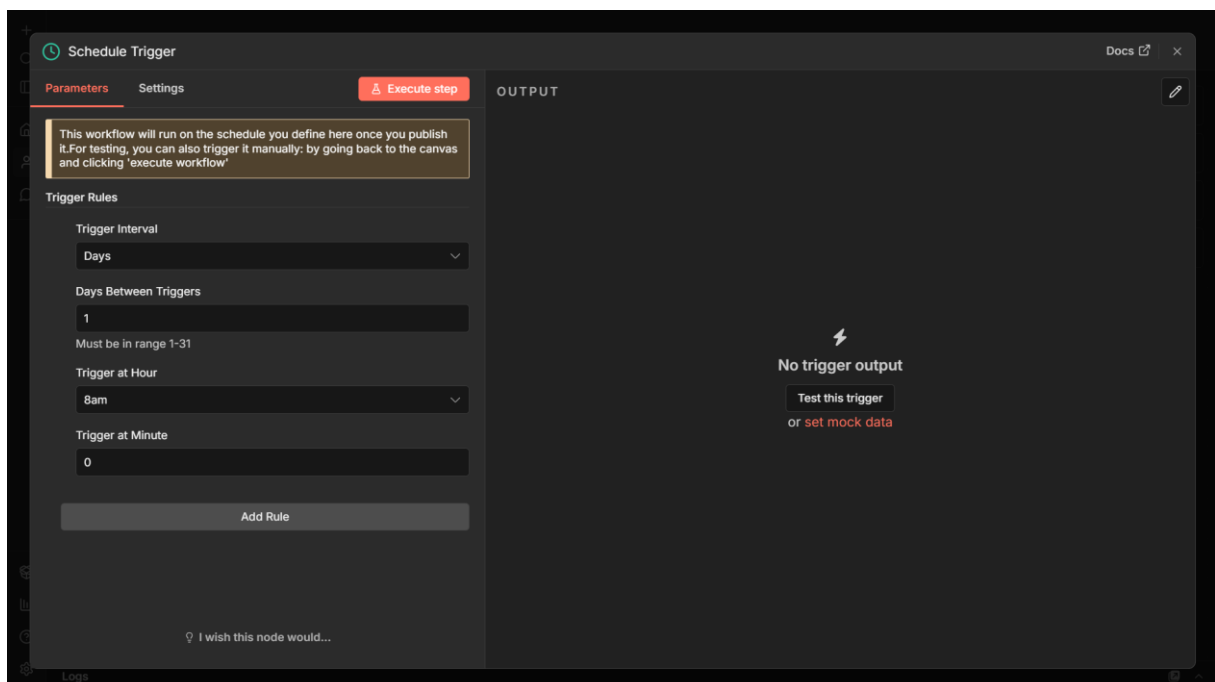
On va maintenant configurer le nœud Schedule Trigger. Pour cela, double-cliquez dessus.



Vous arriverez sur cette page. Comme pour votre workflow, vous pouvez renommer le nœud si vous le souhaitez en cliquant en haut à gauche.

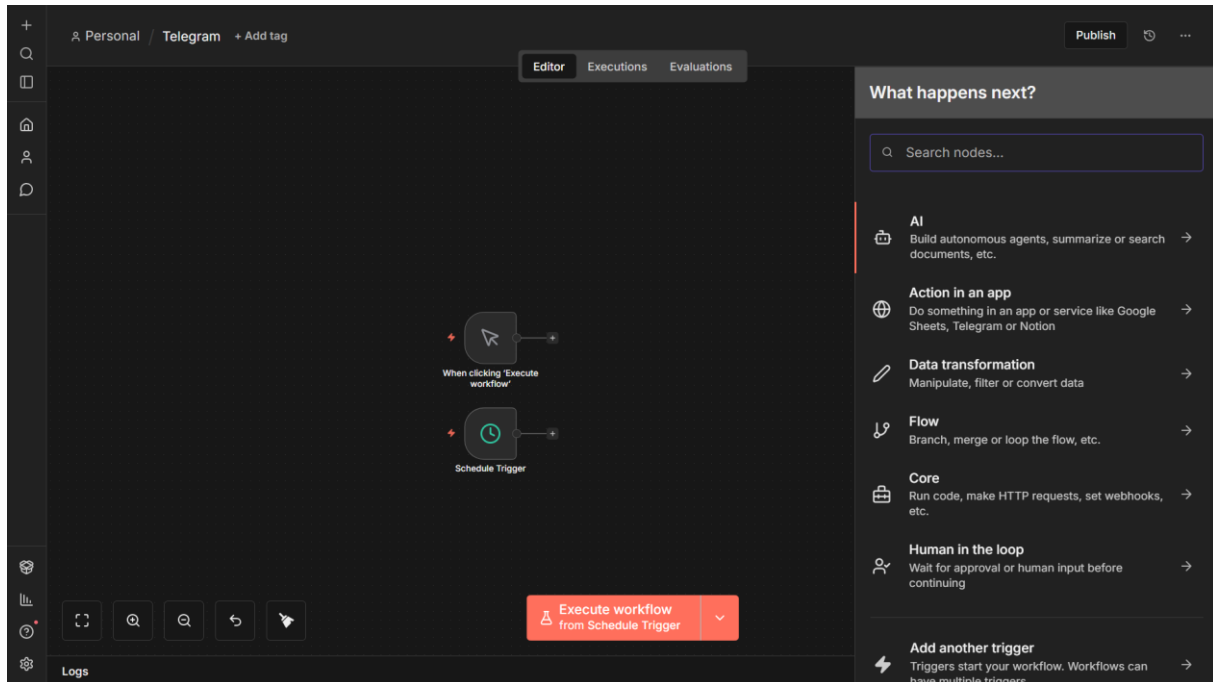
Vous voyez que n8n est entièrement en anglais, ça demande donc un peu d'adaptation si vous n'êtes pas à l'aise avec cette langue. Ici je vais laisser « Trigger Interval » sur Days, car je veux une newsletter quotidienne. Pour « Days Between Triggers », je laisse sur 1, ça fait que votre workflow va s'exécuter tous les jours.

Par contre "Trigger at Hour", je vais changer et mettre 8am (8h du matin).

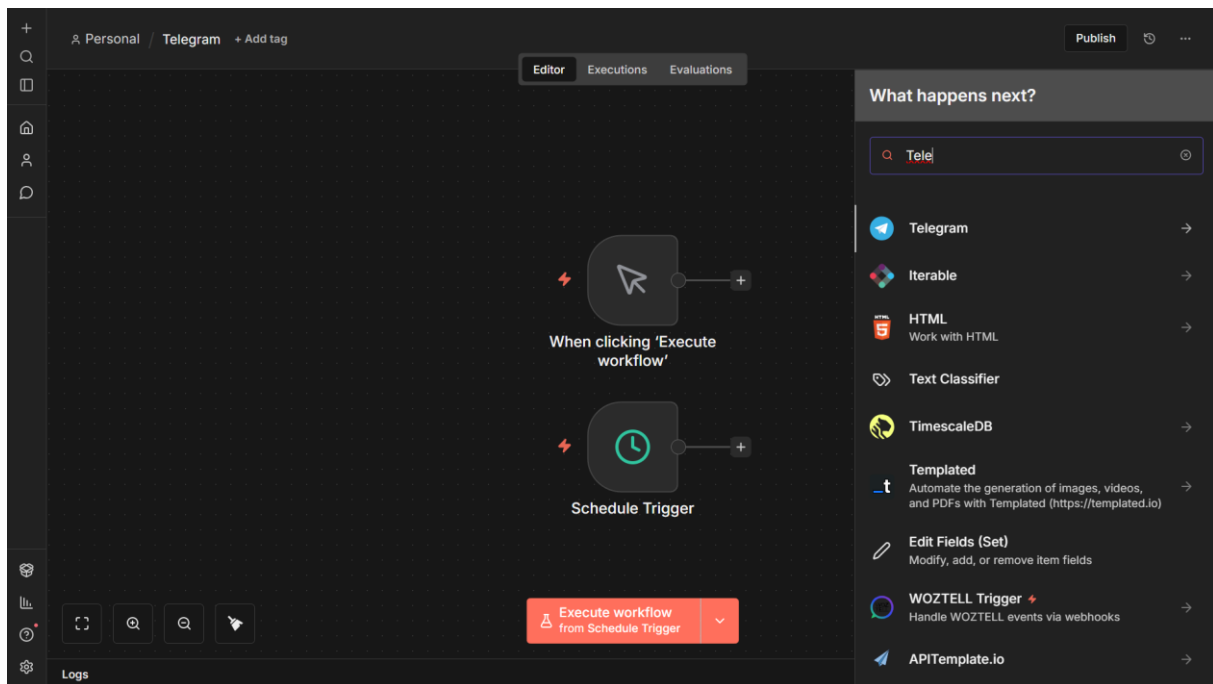


A noter que si vous avez installé n8n en local comme moi, vous devez choisir une heure où votre hébergeur sera en marche. C'est un des défauts de la version auto-hébergé, donc prenez-le en compte, sinon votre workflow ne s'exécutera pas.

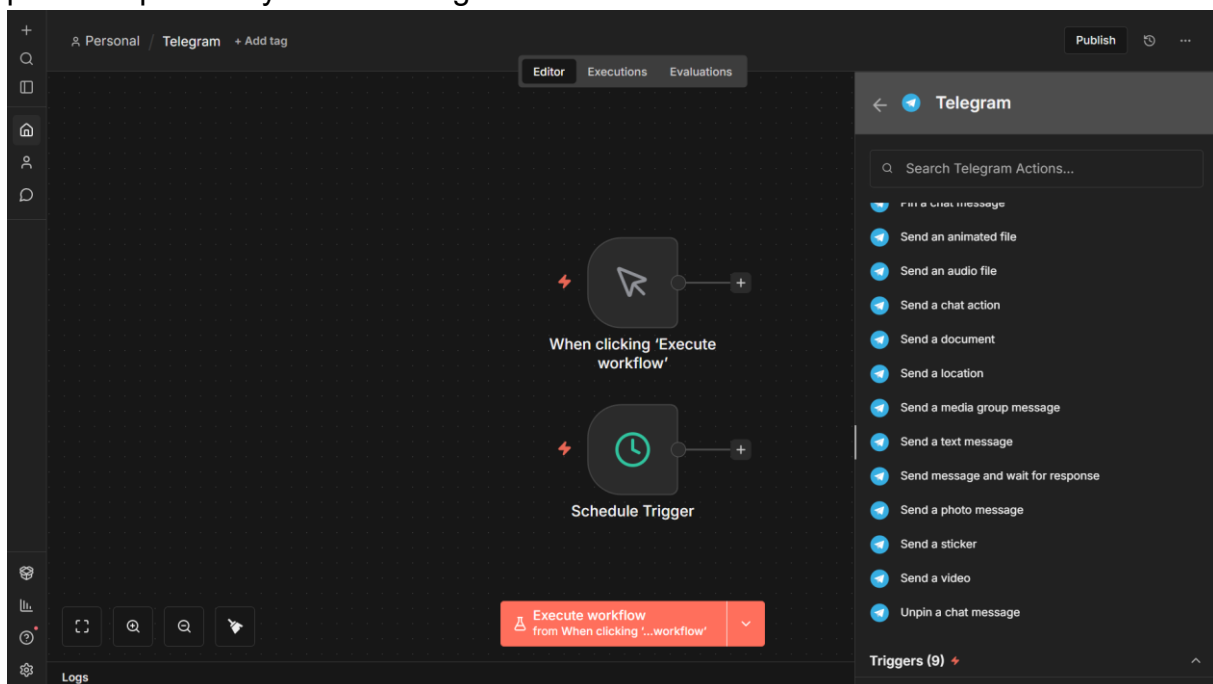
Ok donc pour la phase de construction, on ne va pas connecter le nœud Schedule trigger pour éviter que ce soit trop le bordel, on le branchera à la fin quand les tests seront finis.



Si vous réappuyez sur Tab maintenant, vous voyez que l'entièreté des nœuds s'affichent, car vous avez au moins un Trigger dans votre workflow. Lorsque vous souhaitez utiliser une application spécifique pour faire une action, par exemple envoyer un message Telegram, je vous conseille de passer par les nœuds « Action in an app », car c'est les intégrations natives d'n8n. Ces nœuds sont les plus faciles à utiliser. Vous cherchez votre application, par exemple Telegram ici :



Si vous cliquez dessus, vous voyez que vous avez accès à des actions prédéfinies, par exemple envoyer un message.

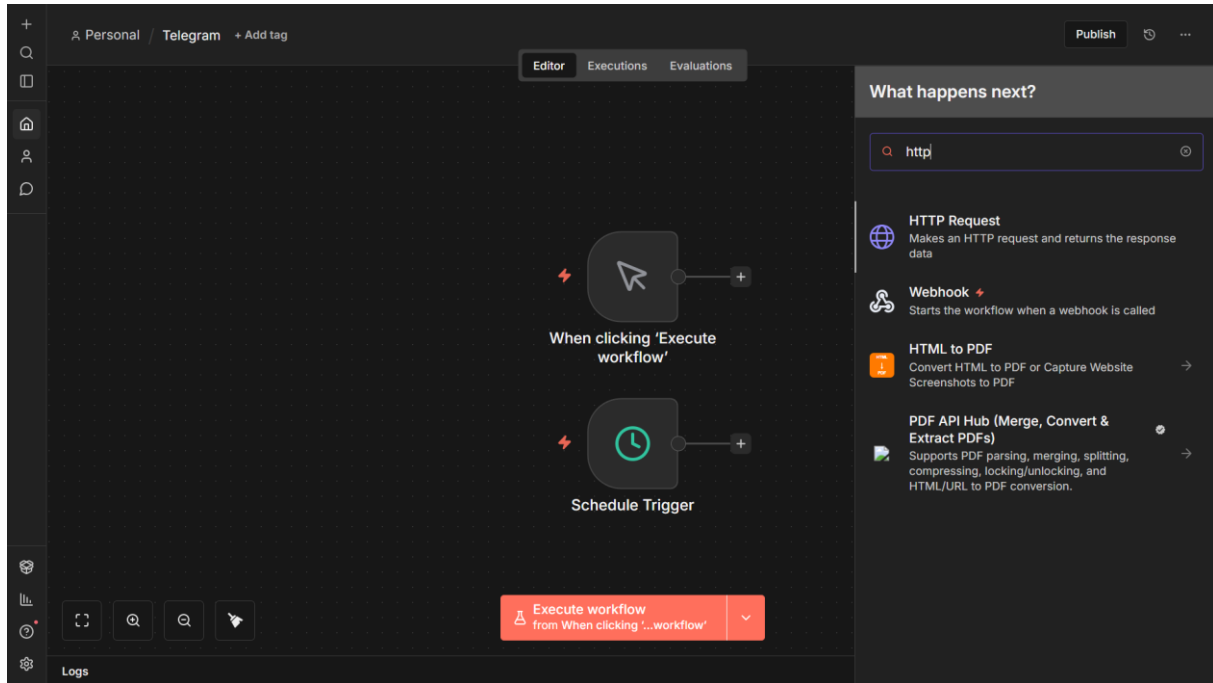


En bas de l'image, vous voyez qu'il y a même des triggers. C'est très pratique quand vous voulez déclencher votre workflow lors d'un événement se produisant dans l'application. Par exemple lorsqu'un message est reçu (ce qu'a fait l'apprenti pour le bot du salon fondamental d'ailleurs :D)

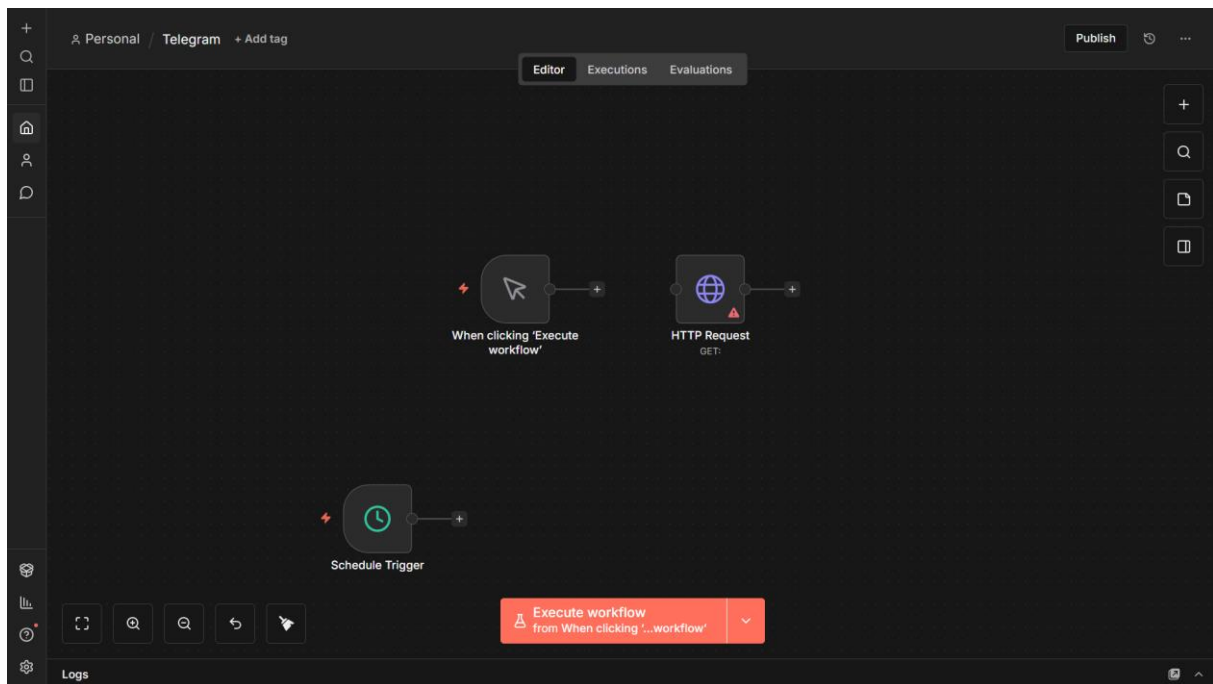
Ces nœuds d'action sont très cools, mais ils ne couvrent pas tous les cas possibles. Des fois, vous allez utiliser une application qui ne figure pas dans la liste des applications natives, ou vous allez vouloir effectuer une action qui n'est pas dans les propositions natives.

Dans mon cas, je n'ai pas trouvé de moyen de récupérer les messages d'un groupe sans être admin de celui-ci avec le nœud Telegram natif. Dans ce cas pas de panique, il y a un plan B, et c'est une des forces d'n8n

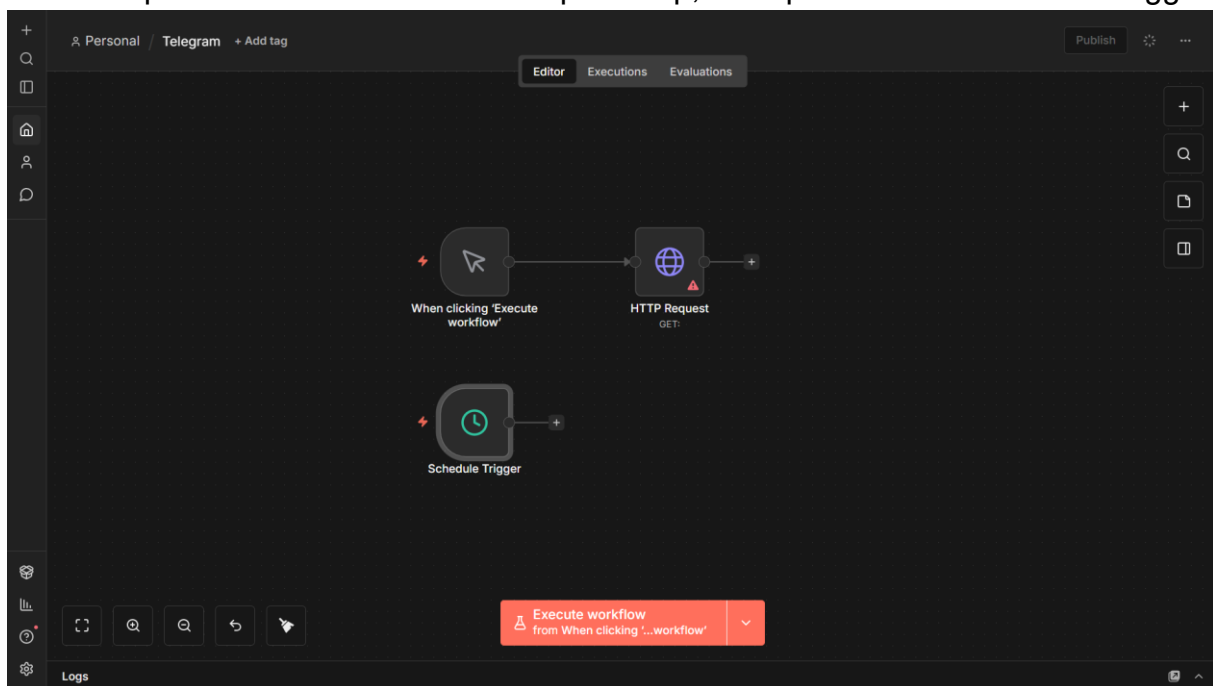
Au lieu de choisir le nœud Telegram, revenez en arrière et tapez « http Request » dans la barre de recherche :



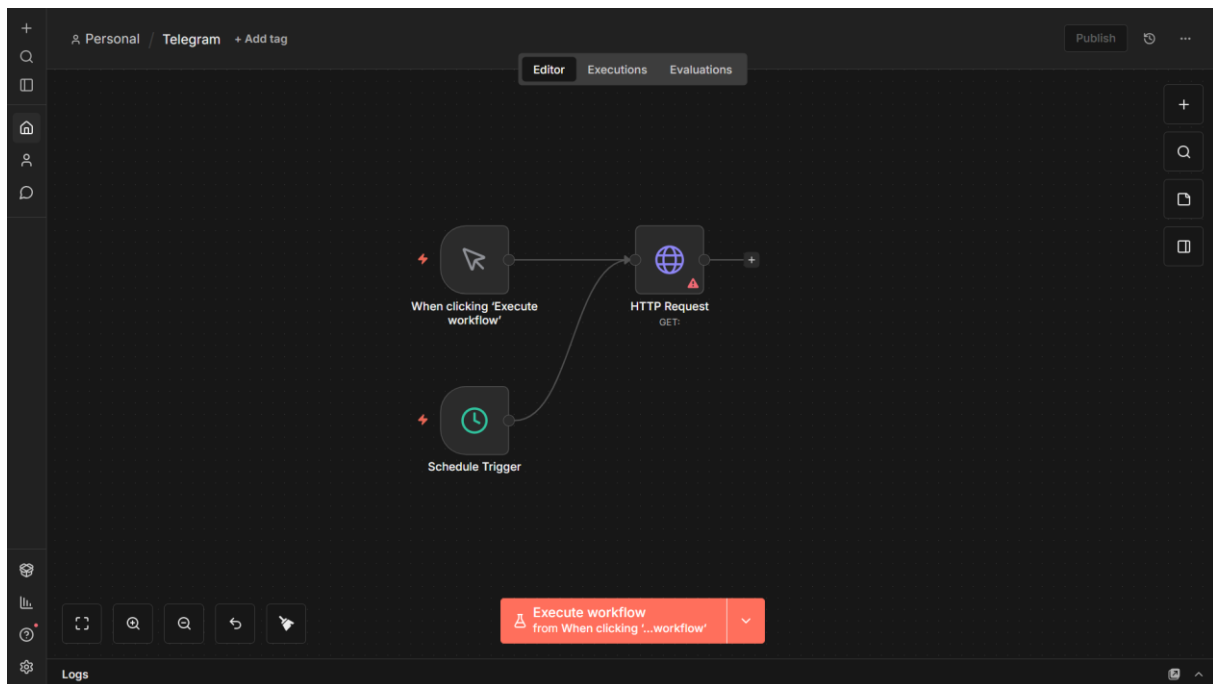
Prenez le premier nœud. Ce nœud est extrêmement utile, car il permet d'étendre le champ des possibles à toutes les applications possédant une API. Ici le problème n'était pas tant l'intégration de Telegram, mais plus les limitations du nœud Telegram en termes de fonctionnalités. En fait, voyez les nœuds d'actions comme des nœuds http Request pré-configurés en rendre l'accès plus intuitif. Ici, on va rentrer direct dans la requête http pure, comme ça vous saurez quoi faire le jour où vous serez limité pas les nœuds d'action.



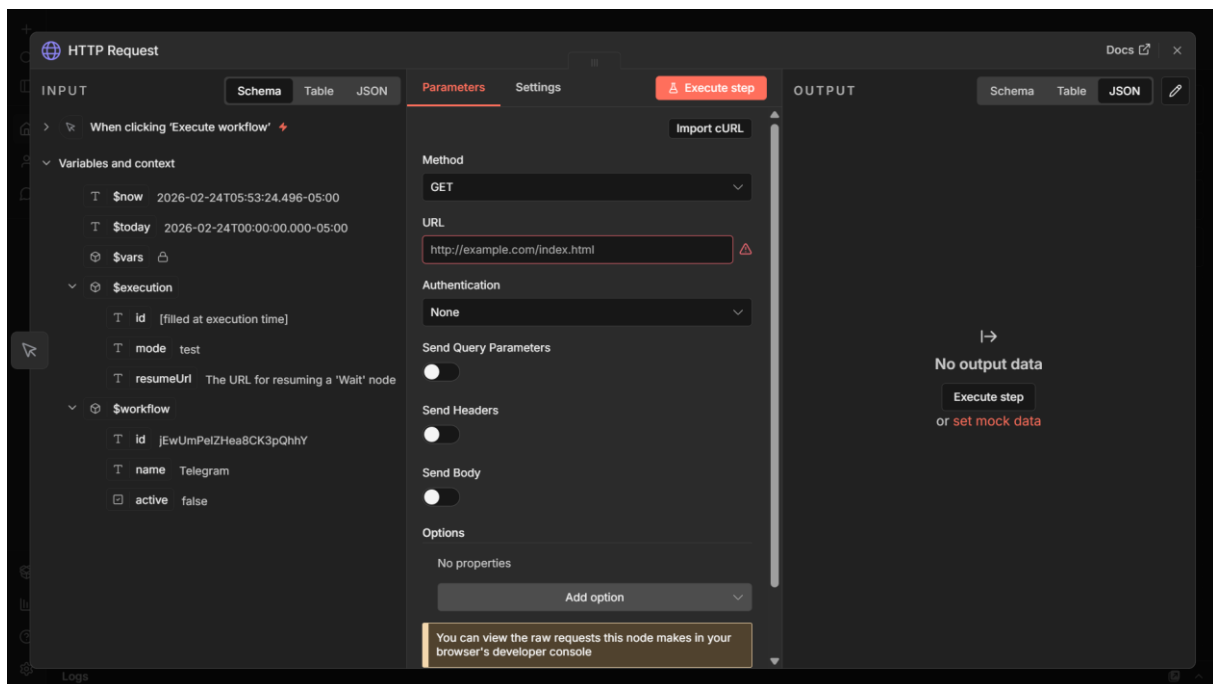
Une fois que vous avez votre nœud requête http, vous pouvez le relier à votre trigger.



A noter que comme décrit tout à l'heure, vous pouvez très bien connecter les deux triggers directement de cette façon :



Mais bon pour un premier workflow restons simple, connectez juste le Trigger manuel pour la phase de test. Vous voyez que le nœud http Request affiche une erreur, normal il n'est pas encore configuré. Pour le faire, double cliquez-dessus.



Vous arrivez sur cette page, elle peut paraître intimidante, mais c'est en fait très intuitif quand on comprend la logique.

Je l'expliquerai qu'une fois car tous les nœuds, sauf les triggers se présentent de cette manière.

La fenêtre se décompose en trois parties :

- La partie de gauche (INPUT) : les données qui rentrent dans votre nœud, vous voyez en haut le nœud déclencheur manuel, et en dessous des clés qui vous pouvez utiliser (on va revenir sur ce concept un peu plus loin)
- La partie du milieu (en gris plus clair) : les paramètres de votre nœud, comme sur le nœud Schedule Trigger
- La partie de droite (OUTPUT) : les données qui sortent de votre nœud une fois qu'il a été exécuté

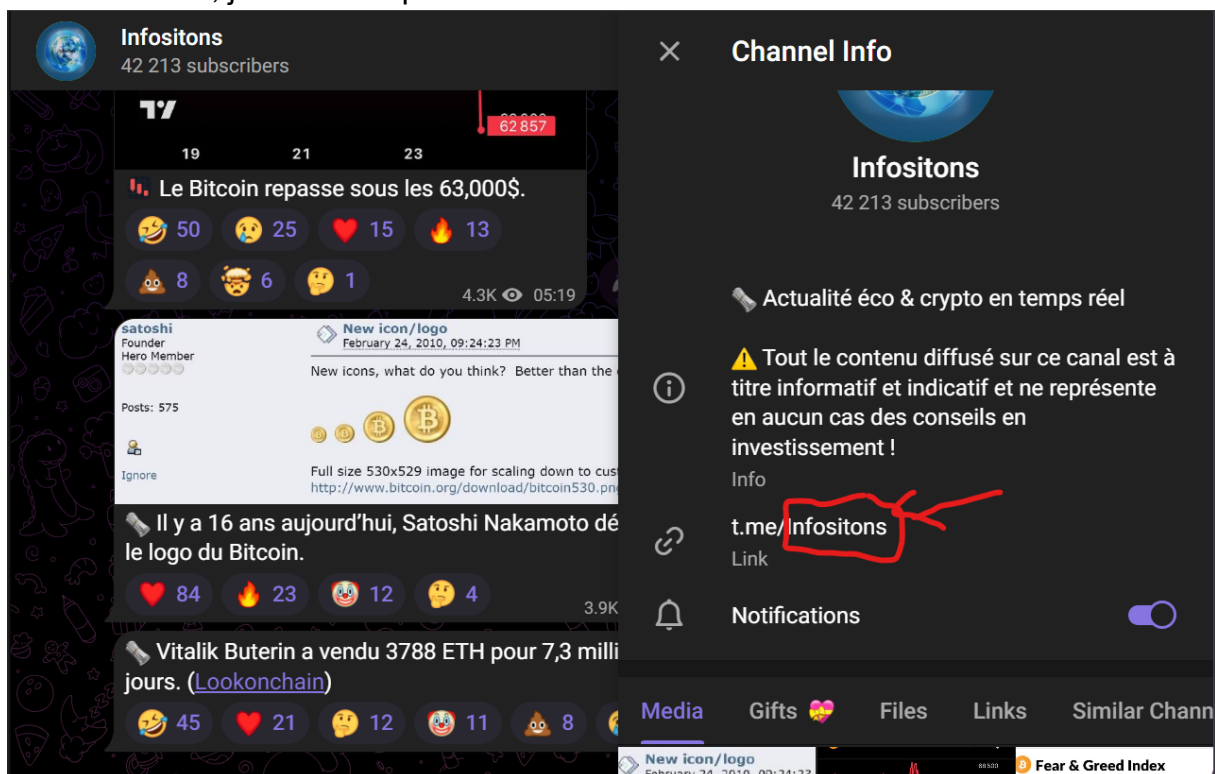
Au final c'est assez logique et ça suit le sens d'exécution du workflow de la gauche vers la droite. C'est comme si vous zoomiez dans le nœud.

Dans « Method », on va laisser GET, car on veut récupérer des informations.

Pour ce qui est de l'URL, rentrez simplement :

<https://t.me/s/> + nom du canal Telegram (il doit être public)

Dans mon cas, je veux récupérer les données de ce canal :



Donc mon lien complet sera : <https://t.me/s/Infositons>

Comment j'ai trouvé ce lien ? Gemini m'a aidé ~\_~

Qu'on soit bien clair là-dessus, pour les requêtes http, ne perdez pas votre temps à lire et recracher les documentations d'API, un LLM le fait très bien.

Ce n'est plus là qu'est la valeur, par contre, imaginer des workflows au préalable, lui donner des pistes de réflexions et le plus important : lui donner un jugement

(feedback) de son travail grâce à un regard critique, c'est là que se trouve la valeur en 2026. Quand vous faites une tâche, essayez toujours de vous demander : est-ce que l'IA peut la faire mieux et plus vite que moi ? Où est-ce que je peux déplacer mon attention pour l'aider au mieux ? Quelles compétences inhérentes à mon humanité puis-je utiliser pour l'assister au mieux ?

Le jugement est une réponse à la dernière question, alors apportez lui votre œil d'humain, et laissez-la faire ce qu'elle sait faire de mieux.

Pareil pour le code, je ne suis pas spécialement partisan du vibe coding pur (demander tout le code à l'IA sans comprendre), car le jour où ça coince, vous êtes foutu. Par contre, l'association du vibe coding et du jugement est très puissante, voici ce que je fais et qui marche très bien :

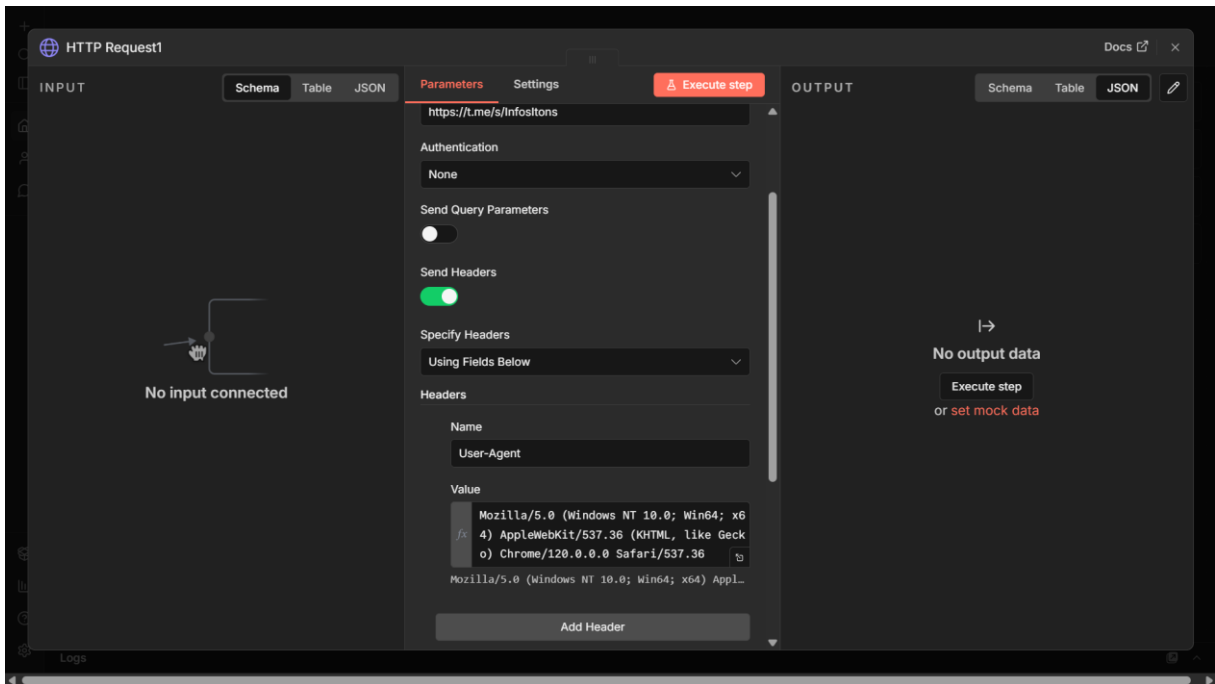
1. Demander à un LLM : code-moi ça (en JSON pour n8n)
2. Explique-moi pas à pas ton raisonnement (Chain of Thought si vous vous rappelez du rapport sur le Prompt Engineering)
3. Lire son code en prenant compte de ses explications, et si vous voyez une incohérence, lui en informer

Pas besoin d'être un expert en code pour faire ça, étant moi-même novice complet avant d'avoir commencé à travailler comme ça, je peux vous dire que ça marche très bien. La plupart du temps, les erreurs seront de l'ordre de la logique pure : le LLM peut vous faire prendre un chemin qui soit n'est pas optimisé, soit mène à une impasse. C'est là que le jugement entre en jeu, et encore une fois, vous n'avez pas besoin d'une expertise en JSON, juste d'un cerveau en état de marche.

Vous verrez qu'au fil du temps, vous aurez de moins en moins besoin de demander à l'IA de vous expliquer son code, car vous arriverez à en lire les grandes lignes.

Cette parenthèse était un peu longue, mais je voulais vraiment clarifier ce point, car visiblement, même certains « doctorants », « PhD », « enseignants chercheurs » ou que sais-je (en France au hasard) ne sont visiblement pas capable de comprendre ça, peut-être sont-ils juste trop intelligents pour s'abaisser à ça 😊

Toujours est-il qu'une fois l'URL rentré, vous pouvez cliquer sur « Send Headers », et remplir les champs Name et Value de telle manière :

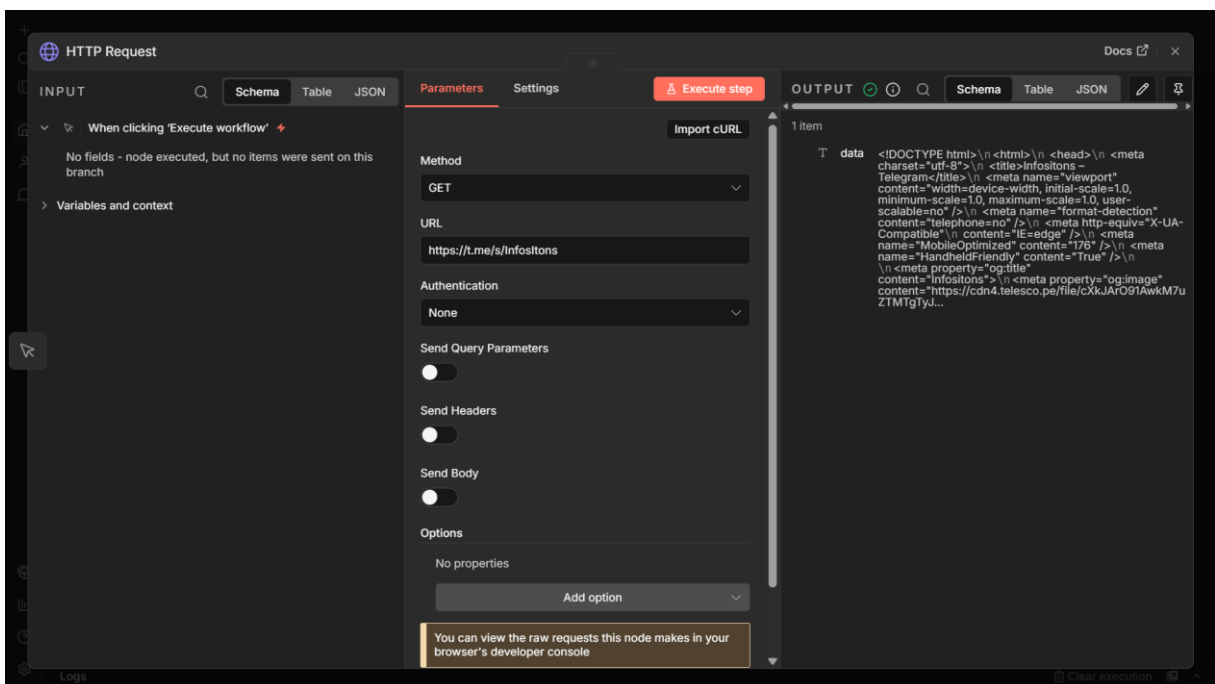


Name : User-Agent

Value : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36

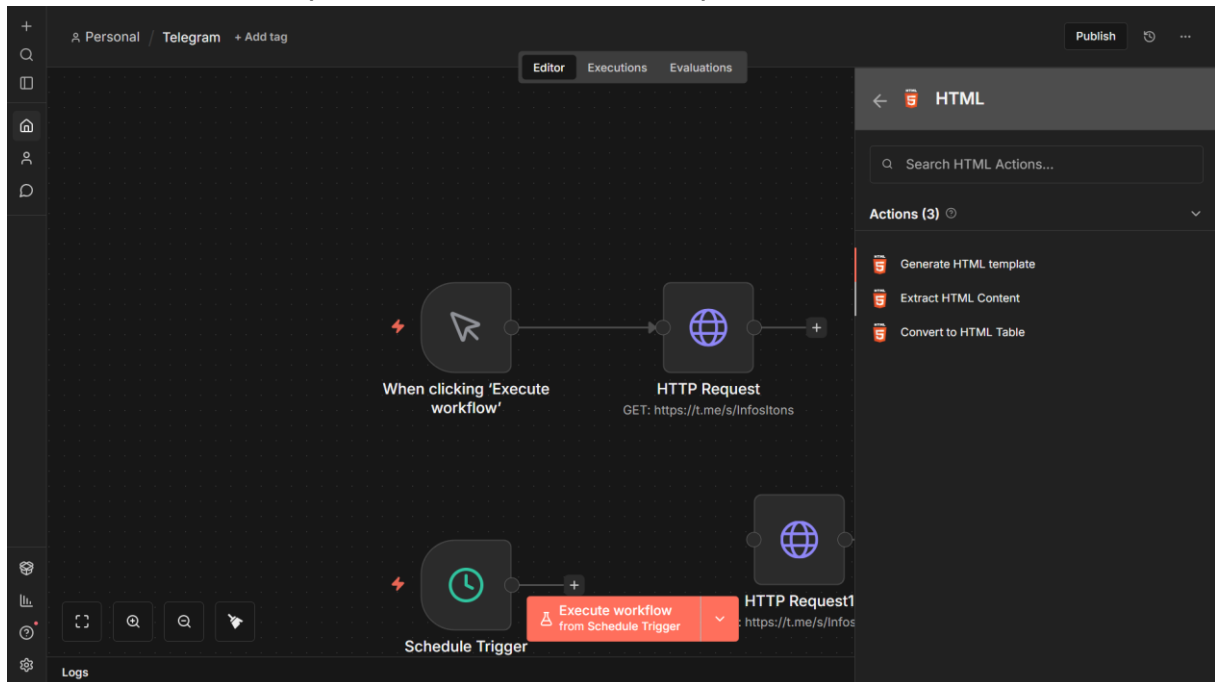
Une fois cela rempli, vous pouvez tester votre nœud en cliquant en haut sur le bouton orange «Execute step », ce bouton va exécuter votre nœud et tous ceux d'avant.

Voilà l'output (les données de sorties) :

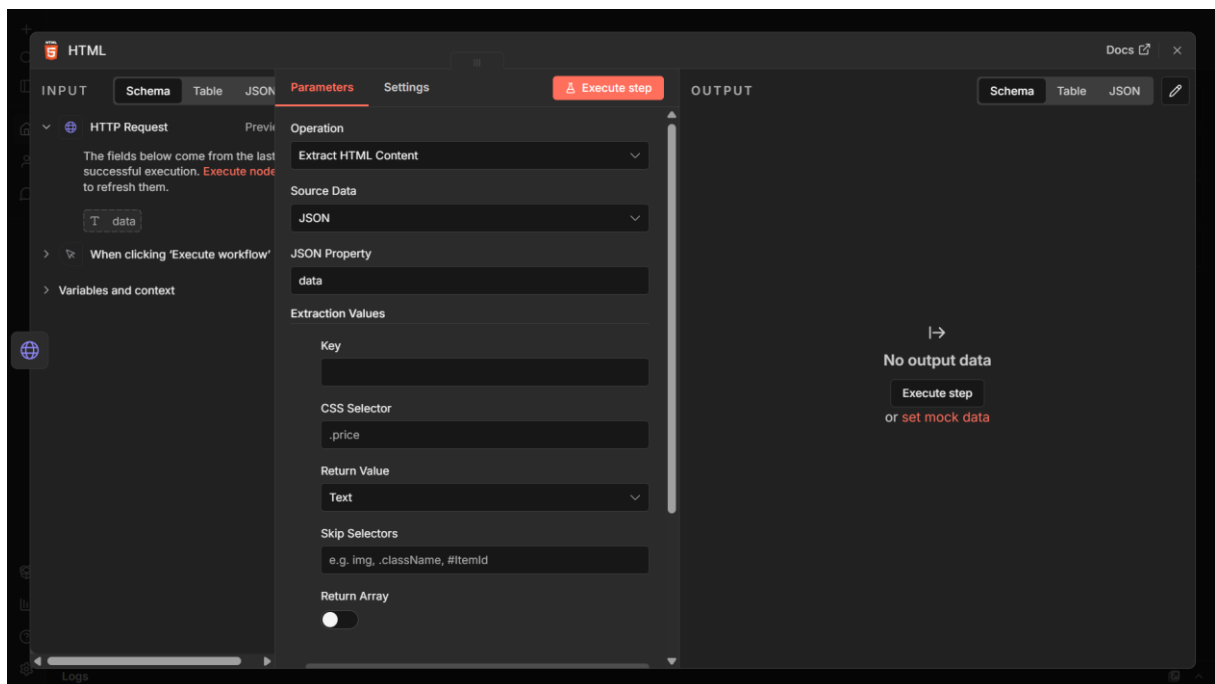


Maintenant qu'on a la donnée brute en HTML, on va la formater un peu pour faciliter

l'étape de filtrage. Pour cela, appuyez de nouveau sur Tab, et chercher le nœud HTML. Choisissez l'option extract HTML content pour formater le code HTML.



Double cliquez sur le nœud pour le configurer :

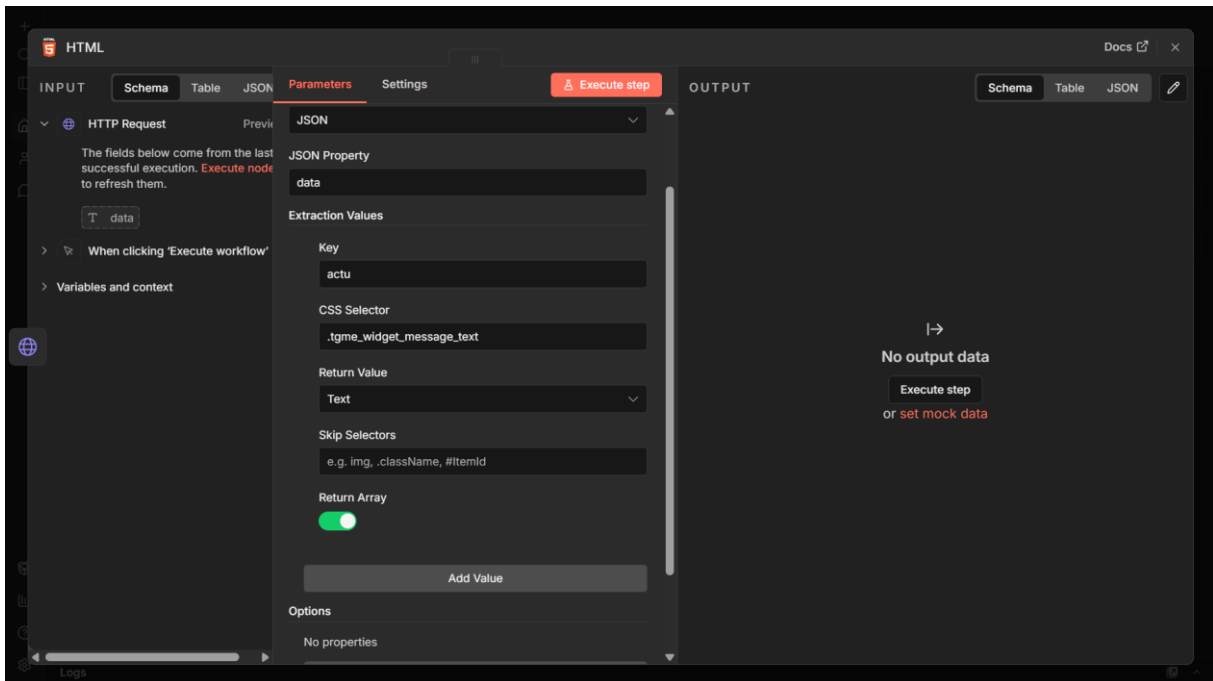


Là ce qu'on va faire, c'est non seulement extraire le code HTML qui nous intéresse (l'info pertinente) dans une clé que je vais appeler actu, mais aussi extraire la date de publication du message dans une autre clé que je vais appeler date. De telle manière, on pourra filtrer la donnée pour garder seulement les messages vieux d'un jour maximum.

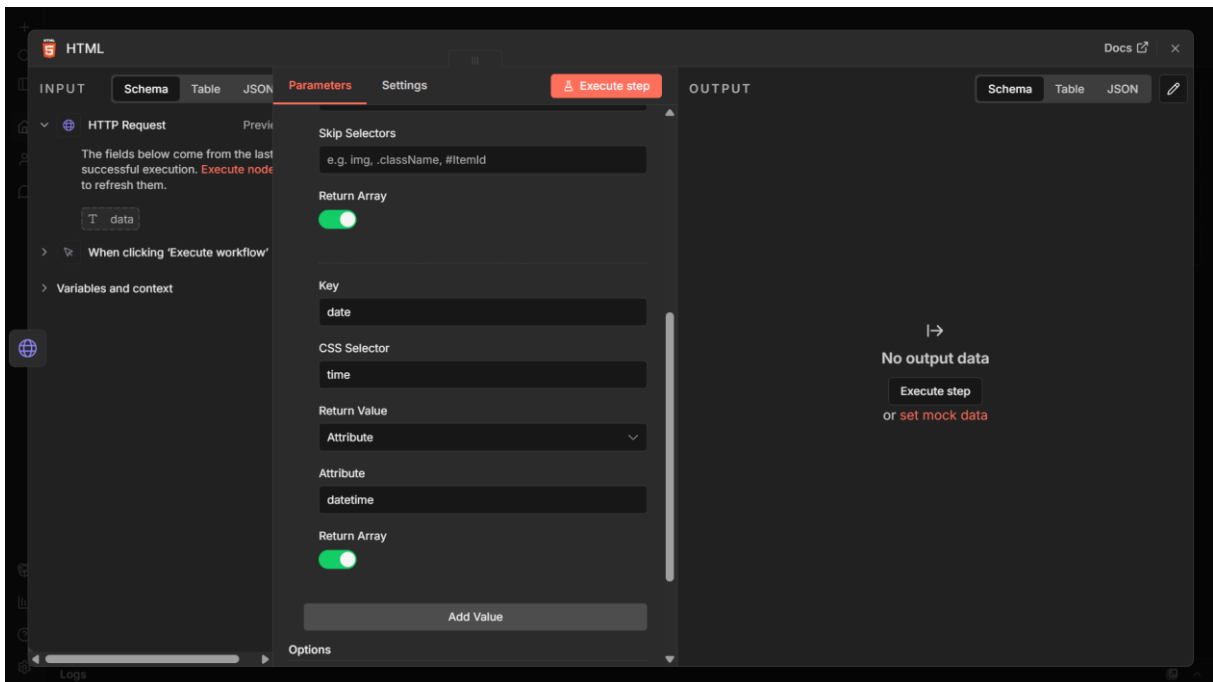
Donc dans Key on a dit qu'on l'appellerait la première clé « actu ».

Dans CSS Selector veuillez rentrer : `.tgme_widget_message_text`

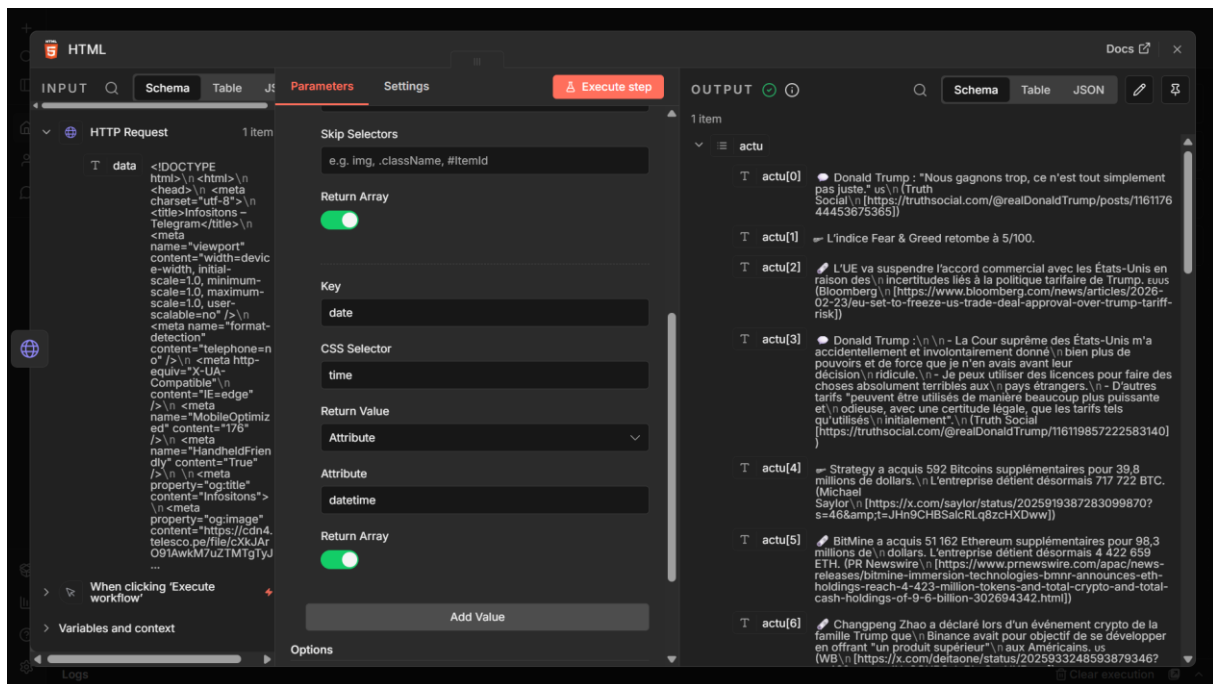
Enfin veuillez à bien cocher Return Array pour faciliter le traitement de donnée



Cliquez ensuite sur Add Value pour ajouter une deuxième clé de sortie, la métrique de temps.



Voilà ça suffira, maintenant cliquez sur Execute step pour tester le nœud :



Voici les données de sorties, comme vous le voyez c'est beaucoup plus clair que tout à l'heure, on a les actus, et si vous descendez vous verrez date[0], date[1], date[2], etc...

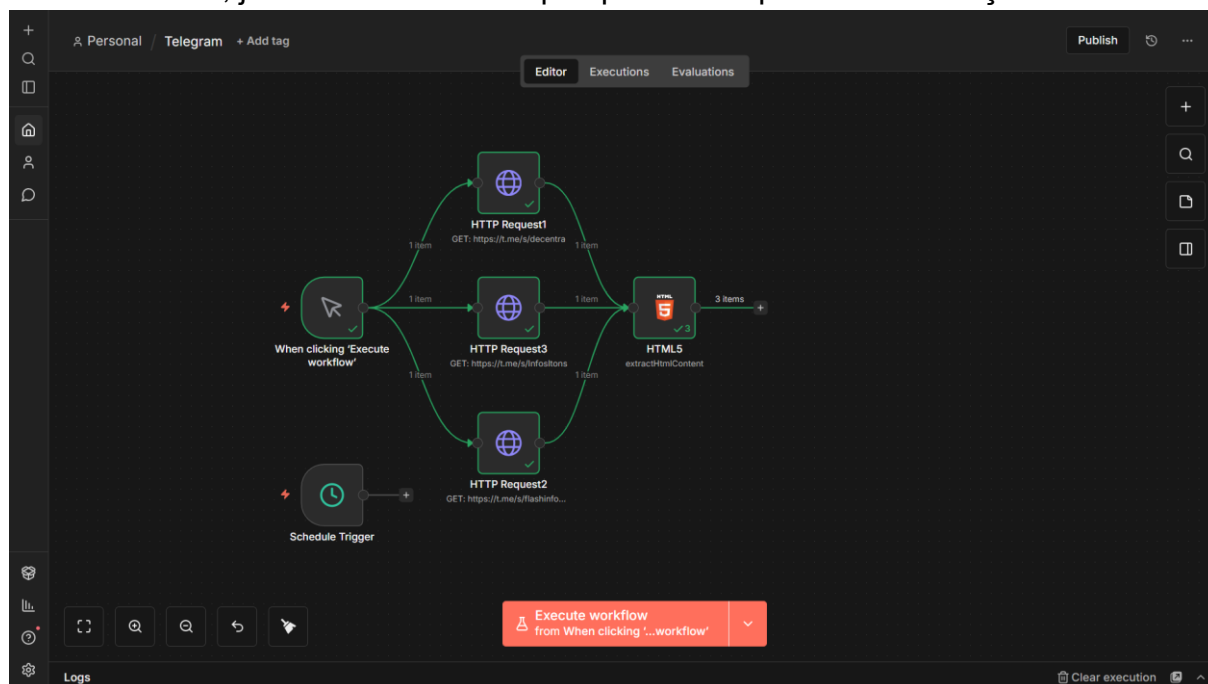
C'est ça que j'appelle les clés depuis tout à l'heure, c'est ce qui permet de faire circuler des données d'un nœud à un autre. Derrière se cache du code JSON, mais pas de panique, n8n a rendu ça super simple, dans 80% des cas, vous aurez juste besoin de drag and drop les clés que vous voulez, on verra ça un peu plus loin.

On a fait le travail pour un canal Telegram, maintenant comme l'intérêt est d'en résumer plusieurs, je vais cloner les nœuds http Request et HTML, en changeant seulement le lien du nœud http Request pour récupérer les informations d'autres canaux.

J'ai choisi de résumer 3 canaux, mais vous pouvez vous amuser à en ajouter plus. Vous pouvez même utiliser d'autres sources (X, Discord, Google, etc..), vous avez l'embaras du choix, et si vous ne trouvez pas votre bonheur dans les nœuds d'action natifs n'oubliez pas : le nœud requête http est votre meilleur ami !

Demandez simplement à votre LLM préféré de vous aider à le remplir en suivant la méthode décrite plus haut.

Dans mon cas, je me retrouve avec quelque chose qui ressemble à ça :



Donc pour résumer rapidement : on a extrait les données brutes soit forme de code HTML indigeste, puis on a formaté ça avec le nœud HTML et on a extrait la date de chaque message. La prochaine étape, c'est de filtrer les messages pour garder ceux vieux de 24h maximum.

Mais avant de faire ça, laissez moi vous introduire la notion d'item dans n8n. Si vous regardez bien sur les branches entre les nœuds, vous verrez marqué « 1 item » et enfin « 3 item » à la sortie du nœud HTML. Ce qu'ils appellent un item, c'est un paquet de nœuds. Ce paquet peu être plus ou moins gros, et dans notre cas ça va poser problème si on ne fait rien. Car regardez bien ce qu'il se passe à l'intérieur du nœud HTML :

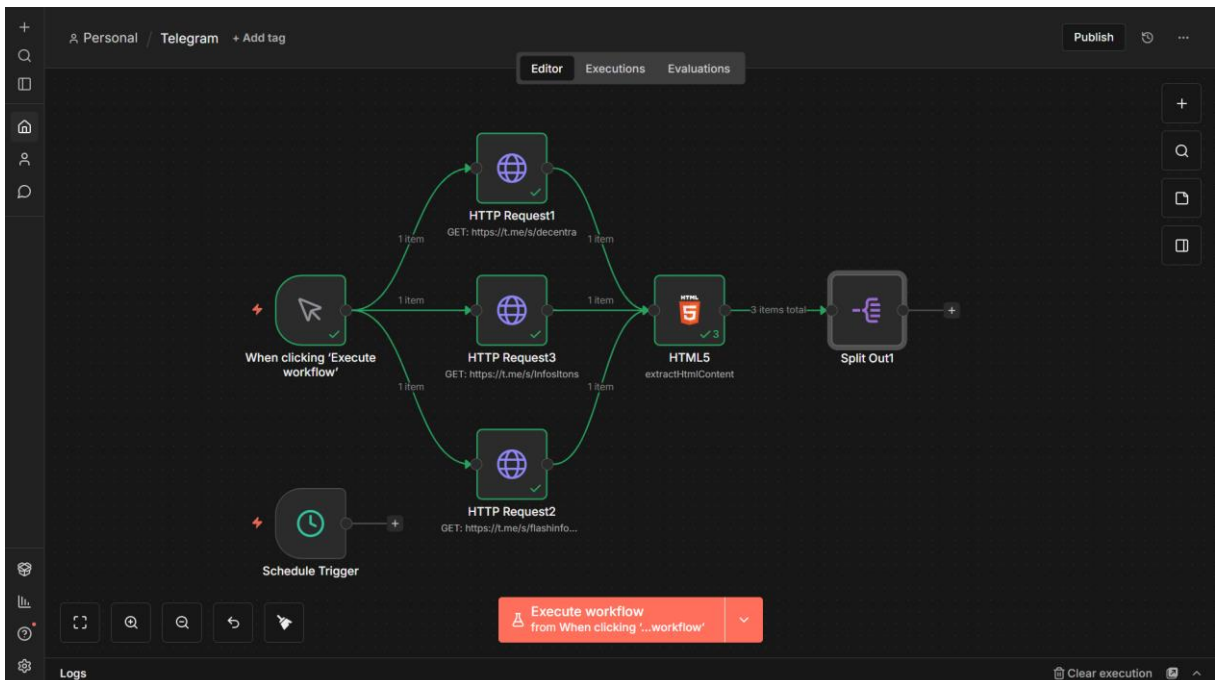
The screenshot shows the configuration of an 'Extract HTML Content' node in the HTML5 tool. The 'Source Data' is set to 'JSON' and the 'JSON Property' is 'data'. The 'Return Value' is set to 'Text'. The 'OUTPUT' panel shows a table with two columns: 'actu' and 'date'. The 'actu' column contains news snippets, and the 'date' column contains timestamps.

actu	date
0 : 🚩 🚩 🚩 Le Fonds monétaire international (FMI) estime que l'Europe doit s'unir pour éviter une crise financière. (Euronews) [https://www.euronews.com/business/2024/10/09/imf-warns-about-european-complacency-and-urges-competitiveness#:~:text=As%20the%20world%20grippl...]	0 : 2024-10-10T18:58:30+00:00
1 : 🚩 🚩 🚩 La Russie appelle les pays partenaires des BRICS à établir une alternative au Fonds monétaire international (FMI) pour faire face à la pression politique exercée par les pays occidentaux. (Reuters) [https://www.reuters.com/world/russia-calls-brics-partners-create-alternative-imf-2024-10-10/]	1 : 2024-10-10T21:55:05+00:00
2 : 🚩 🚩 🚩 Elon Musk dévoile le robotaxi et le robotaxi autonome de Tesla. (Watcher Guru) [https://x.com/watcherguru/status/1844577549084487822?s=46&amp;t=JHn9CHBSalcRLq8zchXDww] / Tesla) [https://x.com/tesla/status/1843922599765590148?s=46&amp;t=JHn9CHBSalcRLq8zchXDww]	2 : null
3 : 🚩 🚩 🚩 Elon Musk dévoile le robotaxi et le robotaxi autonome de Tesla. (Watcher) [https://x.com/watcherguru/status/1844577549084487822?s=46&amp;t=JHn9CHBSalcRLq8zchXDww]	3 : null
	4 : 2024-10-11T03:32:40+00:00
	5 : 2024-10-11T14:13:11+00:00
	6 : 2024-10-13T20:39:21+00:00
	7 : 2024-10-16T15:51:42+00:00
	8 : 2024-10-20T21:37:12+00:00
	9 : 2024-10-23T13:28:11+00:00
	10 : 2024-11-04T21:52:07+00:00
	11 : null
	12 : 2024-11-08T18:01:33+00:00
	13 : 2024-11-12T16:18:08+00:00
	14 : 2024-11-18T15:59:39+00:00
	15 : 2024-11-19T00:51:28+00:00
	16 : 2024-12-02T01:35:48+00:00
	17 : 2024-12-03T18:18:24+00:00
	18 : 2024-12-04T19:36:30+00:00
	19 : 2024-12-04T20:12:07+00:00
	20 : 2024-12-13T12:28:37+00:00
	21 : 2025-02-13T17:51:07+00:00

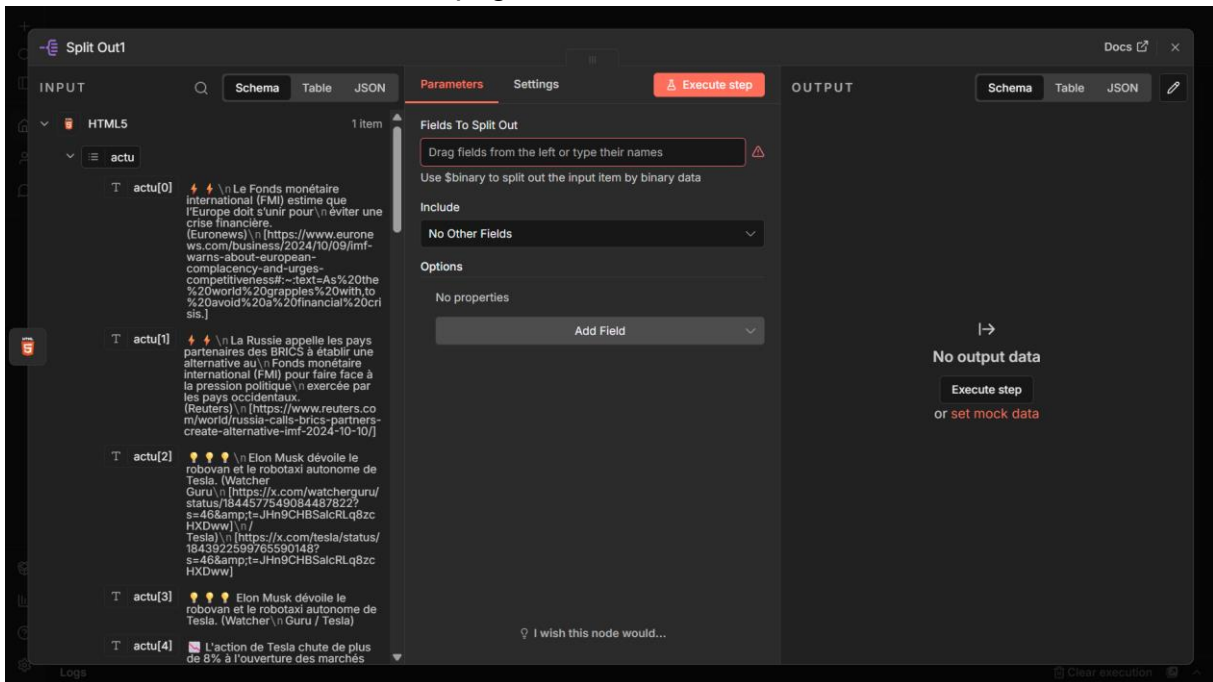
Le tableau de droite en sorti, c'est UN seul item. Là, si vous demandez au nœud Filter qu'on verra juste après, de filtrer les dates, il en sera bien incapable, car votre item contient trop de clés différentes. Pour que le nœud Filter puisse traiter au mieux les données, il faudrait qu'un item = une clé. De telle manière, il pourra regarder item par item, si la condition est respectée ou pas.

Et il se trouve qu'il existe un nœud exprès pour ça : le nœud Split Out.

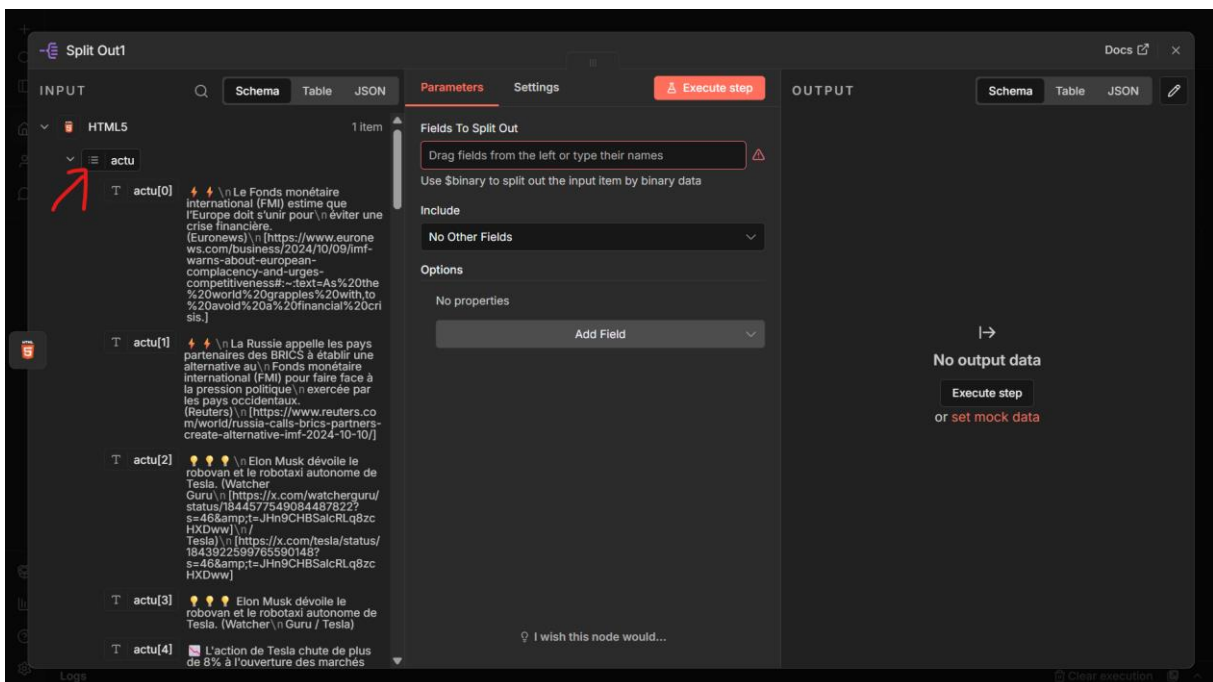
Ajoutez ce nœud à votre workflow, il ressemble à ça :



Comme son nom l'indique, ce nœud permet de spliter un tableau existant. Ouvrez ce nœud et vous tombez sur cette page :



Ici c'est très simple, vous avez une case à remplir, celle « Fields To Split Out », c'est là que vous indiquez le tableau que vous voulez spliter. Vous vous souvenez tout à l'heure dans le nœud HTML, on avait coché une case pour que les items sortent sous forme de tableau : voilà pourquoi. Maintenant on va spliter ces deux tableaux pour que la clé actu[0] et date[0] forment 1 item, actu[1] et date[1] un autre item, etc... . Pour ça très simple, on va juste drag and drop les noms dans deux tableaux dans la case à remplir. Les tableaux étant notés avec le petit logo que vous voyez à gauche du mot « actu » sur l'image :



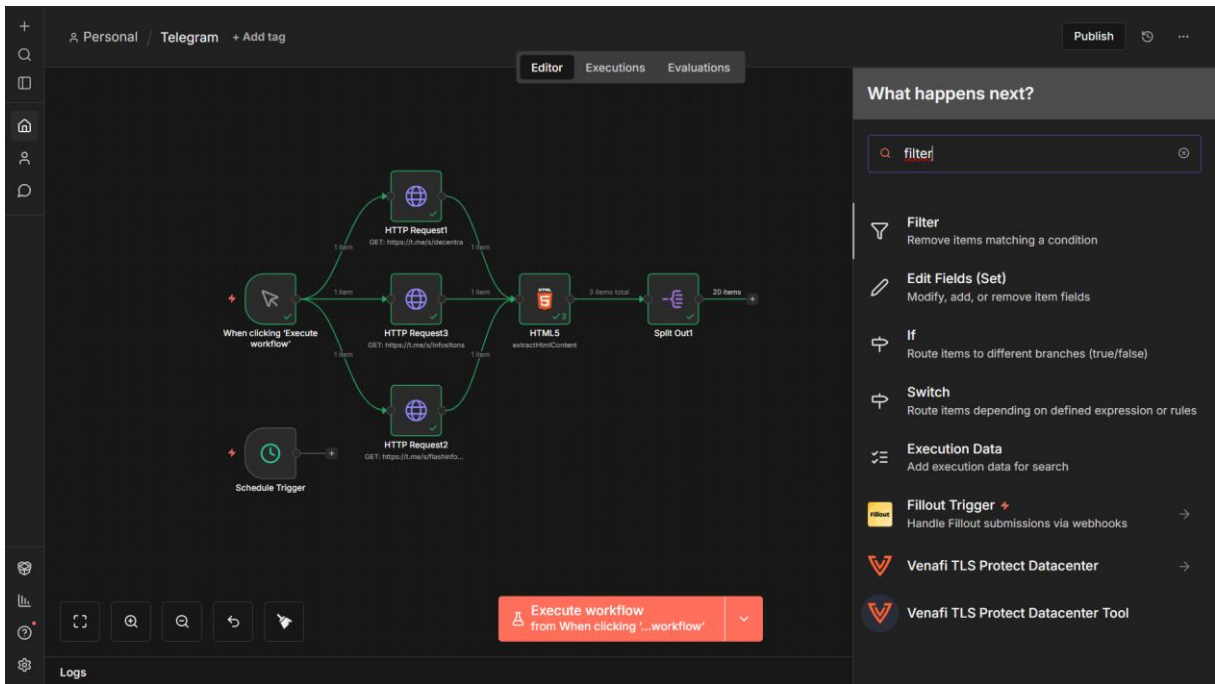
Contrairement aux chaînes de caractères qui sont notées avec un grand T. Vous pouvez maintenant cliquer sur Execute step pour exécuter le nœud, voici les items de sortie :

The screenshot shows the 'Split Out' node configuration. The 'INPUT' section displays a list of HTML5 items, each with a title 'actu' and a text snippet. The 'Parameters' section is configured with 'Fields To Split Out' set to 'actu, date', 'Include' set to 'No Other Fields', and 'Options' set to 'No properties'. The 'OUTPUT' section shows 20 items, each with a JSON object containing 'actu' and 'date' fields. The 'actu' field contains the text snippet from the input, and the 'date' field contains a timestamp.

Vous voyez qu'en haut à gauche de la fenêtre OUTPUT, j'ai cette fois 20 items et non plus trois. Normal car les tableaux ont été splité en plus petit item. Maintenant j'ai dans un seul item :

- L'actu extraite du canal Telegram
- La date associée juste en dessous

Cette configuration est parfaite pour un filtrage, on n'aura qu'à demander au prochain nœud de regarder la date pour chaque item et voir si elle respecte une certaine condition. Faisons ça, recherchez maintenant le nœud « Filter »

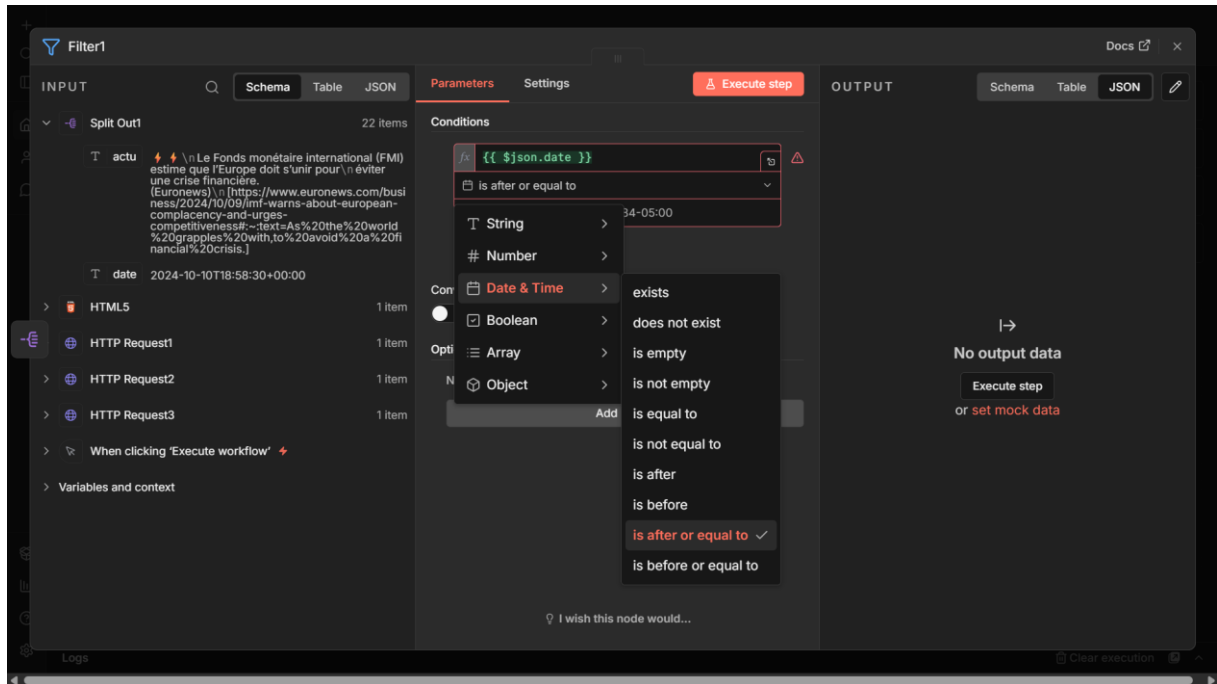


Voilà comment se présente ce nœud qui sert tout le temps pour le traitement de données :

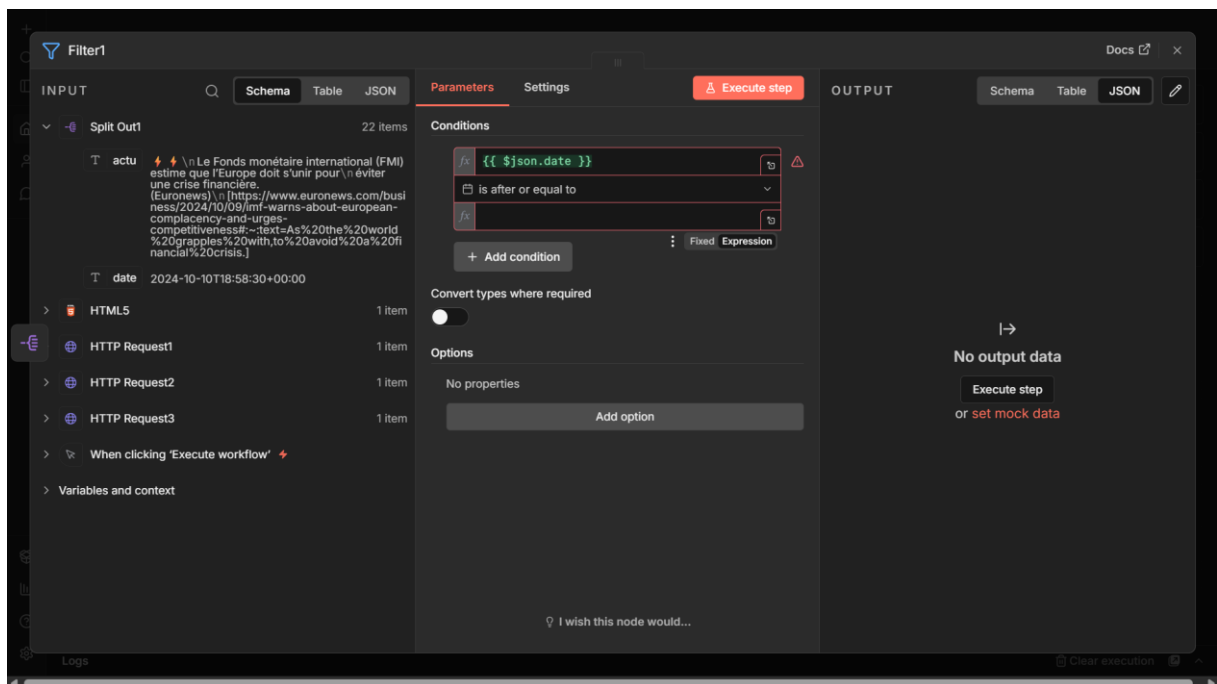
- Value 1 : La clé que vous voulez filtrer (dans notre cas la date)
- Is equal to & Value 2 : la condition pour passer le filtre (dépend de la nature de votre clé initiale)

The screenshot shows the configuration window for a 'Filter1' node. The 'Parameters' tab is selected, showing a condition where 'value1' is equal to 'value2'. The 'INPUT' section on the left shows a list of nodes: 'Split Out' (22 items), 'HTML5' (1 item), 'HTTP Request1' (1 item), 'HTTP Request2' (1 item), 'HTTP Request3' (1 item), and 'When clicking "Execute workflow"'. The 'OUTPUT' section on the right shows 'No output data' and buttons for 'Execute step' or 'set mock data'. At the bottom, there's a link 'I wish this node would...'.

Là il y a plusieurs manières de faire, dans mon cas dans Value 1, j'ai mis la clé « date », puis choisi « is after or equal to » pour une date.



Pour la Value 2, comme je ne veux pas une date fixe mais une date qui s'adapte automatiquement, il va falloir manipuler un peu de code JSON. Pour ça cliquez sur « expression ».



C'est là qu'est tout l'avantage d'n8n, un mélange entre une interface no-code pour les actions les plus simples avec du drag and drop comme on a vu tout à l'heure, et code pour des expressions plus avancées.

Là encore, pas besoin d'être un expert en JSON, juste d'arriver à lire ce que sort Gemini, il m'a fait ça :

```
{{ $now.minus({ days: 1 }) }}
```

Vous allez voir c'est très simple, \$now, ça donne la date exacte qui se met à jour automatiquement. Par exemple il sait qu'aujourd'hui c'est le 25/02/26, mais souvenez-vous, je veux uniquement les data datant d'un jour maximum.

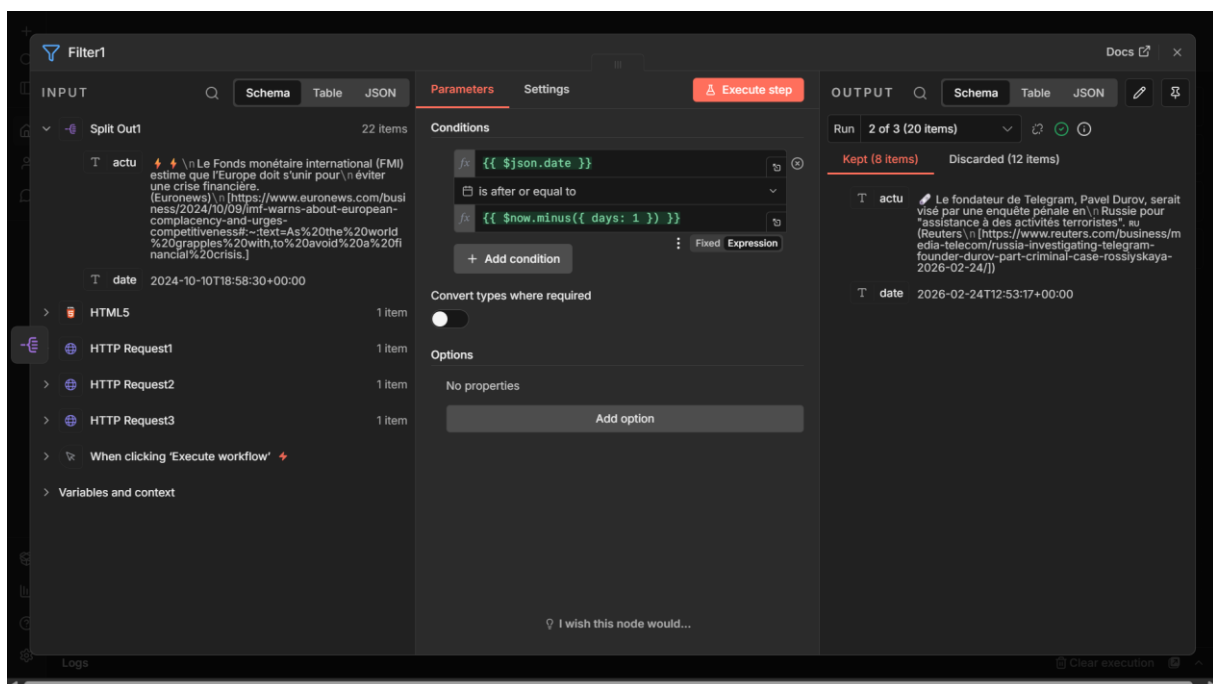
Pour ce faire, il faut reculer d'un jour par rapport au jour actuel, de tel sorte qu'en prenant les dates égales ou après celle-ci, je n'ai pas de dates plus anciennes qu'un jour à compter du jour d'aujourd'hui.

L'explication était peut-être bancale mdr mais regardez le JSON, c'est beaucoup plus clair, il dit :

Maintenant – 1 jour.

.minus ça permet de soustraire un intervalle de temps, donc .minus({ days: 1 }) ça veut dire : enlève un jour.

En prenant les dates supérieures ou égales à ça je suis sûr de respecter ma condition. On obtient :



The screenshot shows a workflow editor interface. On the left, the 'INPUT' section displays a list of items, including a news article snippet and a date field. The main area shows a 'Filter1' step with 'Parameters' and 'Settings' tabs. The 'Conditions' section is active, showing a condition: 'is after or equal to' with the expression '{{ \$now.minus({ days: 1 }) }}'. Below this, there are options for 'Convert types where required' and 'Options'. On the right, the 'OUTPUT' section shows the results of the filter: 'Kept (8 items)' and 'Discarded (12 items)'. The 'Kept' items include a news article snippet and a date field.

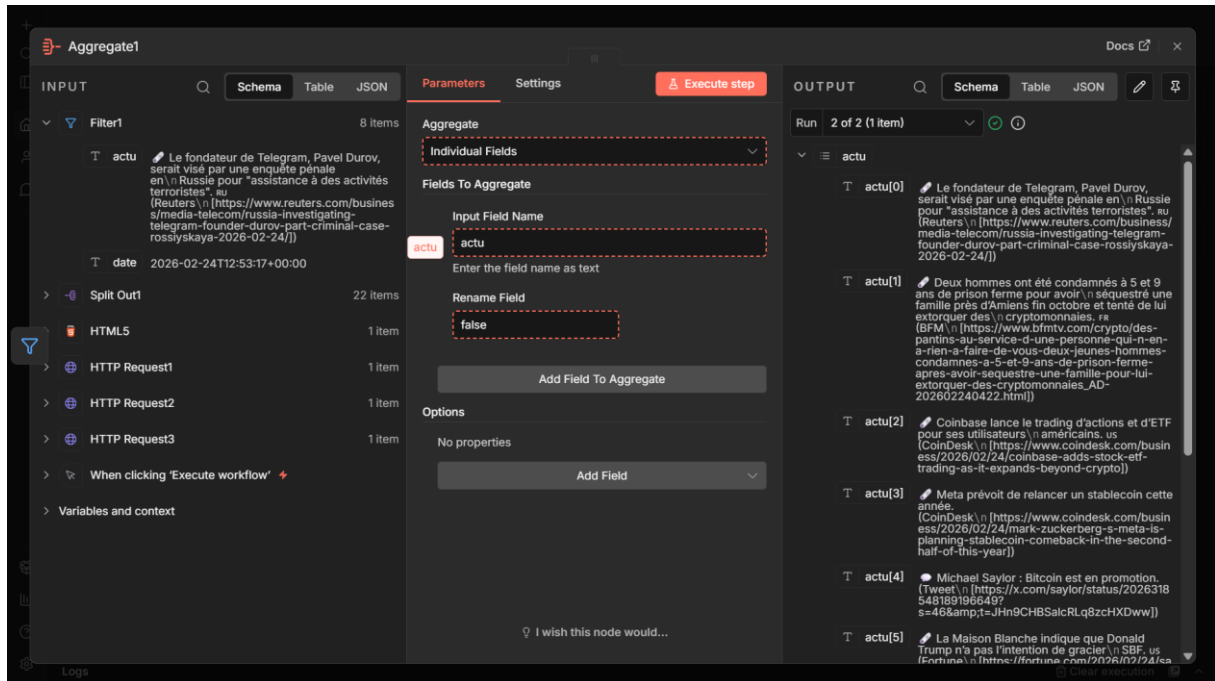
Super le filtre marche bien, vous voyez que j'ai 8 items qui sont passés et 12 qui se sont fait dégager.

Donc on en résumé : on a récupéré les données depuis Telegram et on les a filtrées pour garder que les plus récentes. La dernière étape consiste à utiliser l'IA pour nous faire une belle newsletter prête à l'emploi.

Mais attention, il a y un point à régler avant. Je vous rappelle qu'on a splité nos data au préalable avec le nœud Split Out. Si on envoie des data splitées à l'IA, elle va les recevoir une à une, et donc s'exécuter 8 fois, sans avoir la totalité des données.

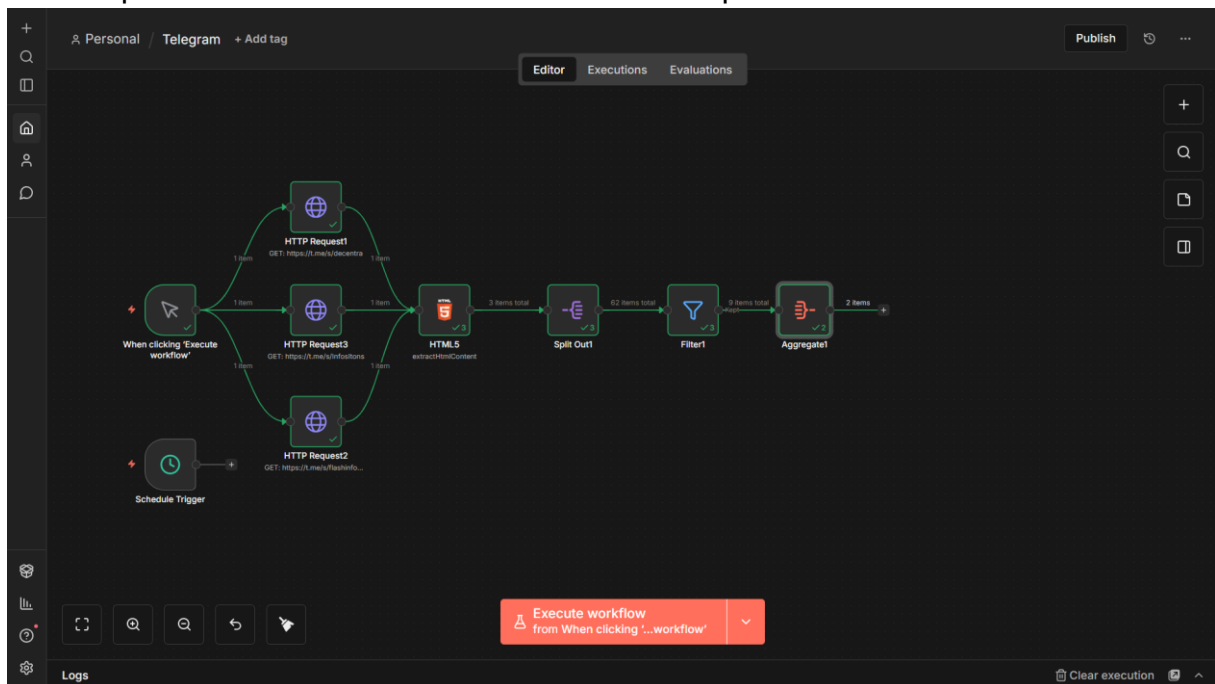
Vous perdrez donc de l'argent en plus de ne pas obtenir le résultat voulu. La solution est d'utiliser le nœud inverse du nœud Split Out : le nœud Aggregate.

Au lieu de casser un tableau, ce nœud va en créer un, c'est exactement l'inverse du Split Out.



Dans « Input Field To aggregate », veuillez bien à drag and drop “actu”, de tel sorte que toutes les données filtrées soient dans un seul et même item pour le traitement IA. Pas besoin de la date de publication ici, comme on a déjà filtré.

Voici à quoi ressemble le workflow total à cette étape :



On a maintenant tous les éléments pour le traitement IA, donc c'est parti.

Avant toute chose, sachez qu'il n'y a pas besoin d'abonnement quelconque pour utiliser l'IA avec n8n, comme on utilise des APIs. En fait, vous payez directement les tokens que vous consommez. Rassurez-vous, les tokens sont (pour l'instant du moins), vraiment pas cher si vous prenez les modèles les moins puissants (ce qui est largement suffisant pour ce type de tâche).

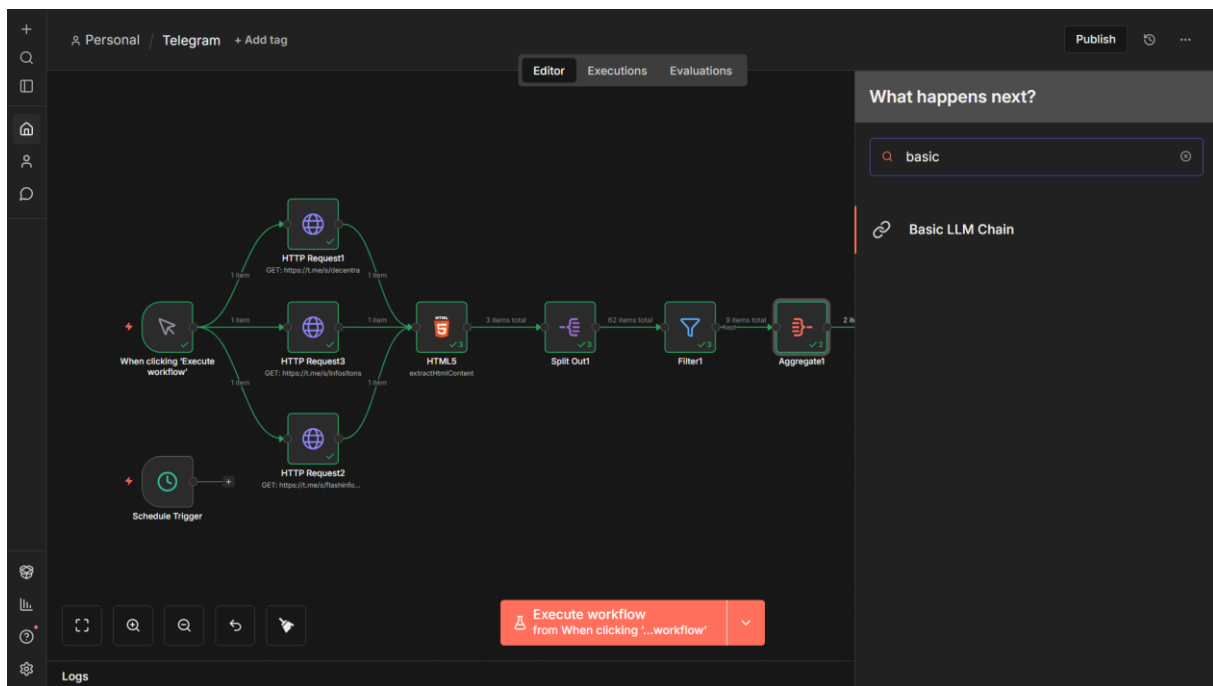
Je vais pas vous faire un tuto sur comment acheter des tokens, mais je vous conseille de commencer en mettant un toute petite somme (2-3€), car croyez-moi, avec ça vous pouvez déjà faire pas mal de chose, sachant qu'une génération comme celle qu'on va faire coute environ 1 centime (en prenant bien le modèle le moins puissant), et encore, je suis sur Claude, un modèle de base cher.

Si vous voulez acheter des crédits chez OpenAI : <https://platform.openai.com/usage>

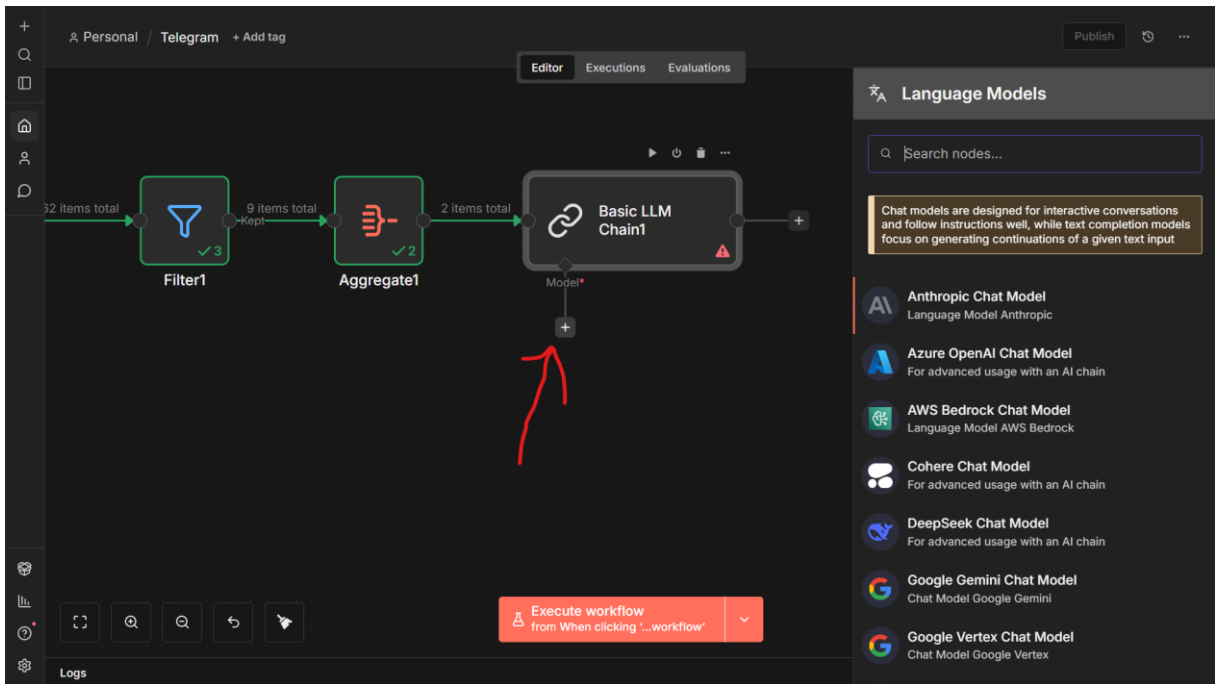
Sinon chez Anthropic : <https://platform.claude.com/settings/billing>

Ou chez qui vous voulez. Si vous ne voulez vraiment pas payer, vous pouvez utiliser un LLM en local bien sûr.

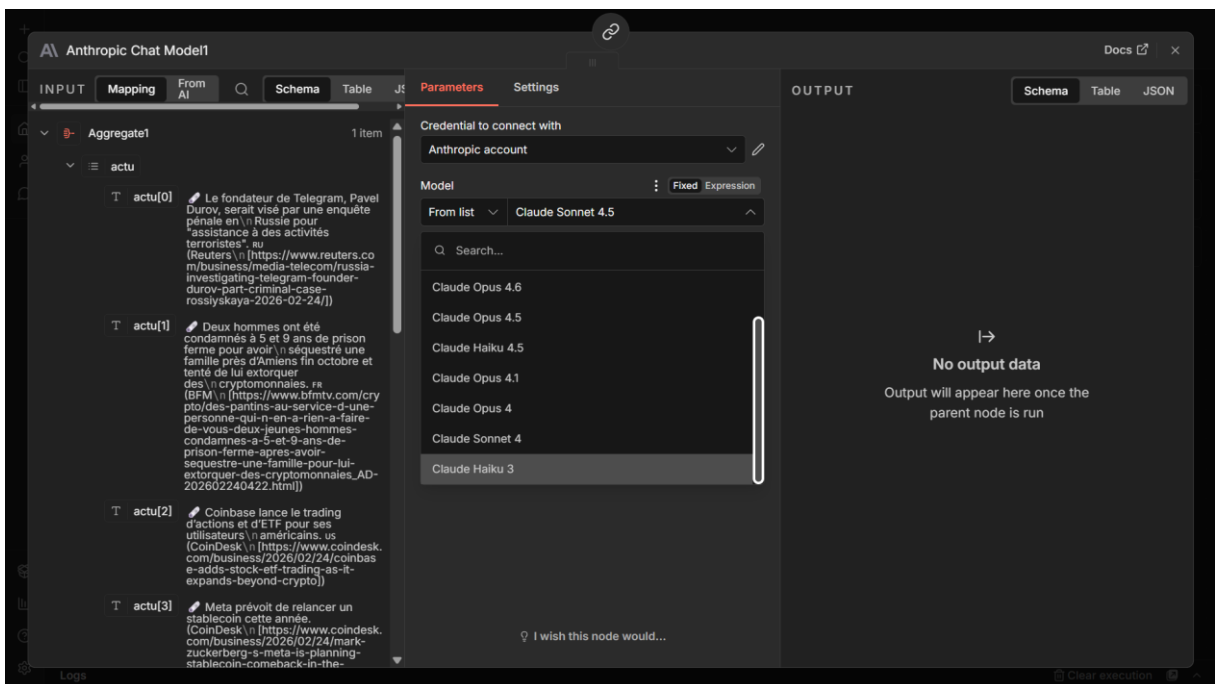
De retour sur n8n, recherchez le nœud Basic LLM Chain :



Quand vous cliquez sur Model, vous pouvez choisir le modèle chez qui vous avez acheté des crédits (vous voyez que la liste est longue et qu'il y a aussi de l'open source)



Dans le nœud de votre modèle, veuillez bien à choisir le moins puissant, perso je vais prendre Claude Haiku 3 ici :



N'oubliez pas de connecter votre compte au préalable via une clé API générée sur votre compte OpenAI/Anthropic/tout ce que vous voulez. Après quoi retournez sur le nœud LLM Chain, et sélectionnez define below sur Source of Prompt, pour définir le prompt manuellement. J'ai mis un prompt random :

Prompt :

Tu es un analyste expert en Web3 et Macro-économie avec 15 ans d'expérience. Ta mission est de synthétiser les actualités Telegram suivantes pour un investisseur pressé mais exigeant.

Voici les news brutes des dernières 24h :

```
{{ $json.actu }}
```

---

### STRUCTURE DU RAPPORT :

1. 🚀 **\*\*Le Top 3 Stratégique\*\*** : Identifie les 3 news les plus importantes. Pour chacune, explique en une phrase *\*pourquoi\** c'est crucial (l'impact potentiel sur le marché).
2. 📊 **\*\*Analyse Thématique\*\*** : Regroupe le reste des informations par catégories (ex: Régulation/Politique, Adoption Institutionnelle, Écosystème NFT/Crypto).
3. 💡 **\*\*Le "Sentiment du Jour"\*\*\*** : Donne une tendance globale (Bullish, Bearish ou Neutre) basée sur ces informations.

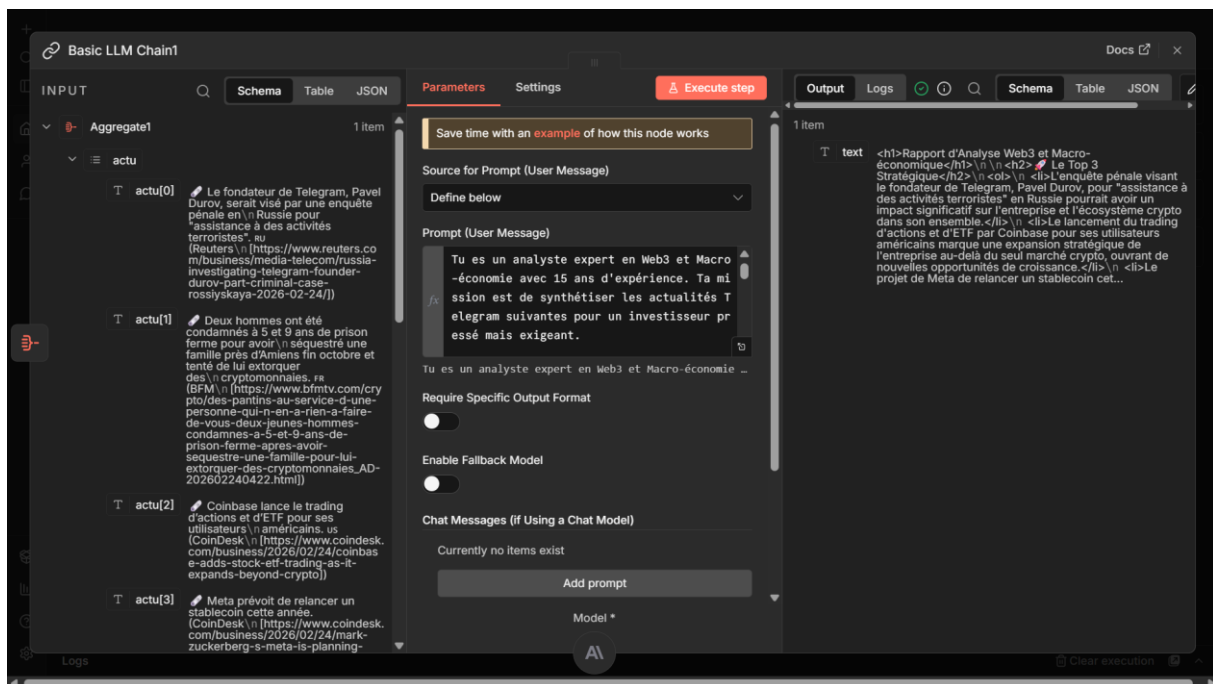
### TON ET STYLE :

- Professionnel, concis et analytique.
- Utilise des émojis pertinents pour la lisibilité.
- Ne répète pas les liens URL, concentre-toi sur la synthèse du contenu.
- Si une news concerne Trump ou la Fed, accorde-lui une attention particulière.

Réponds directement en HTML.

Y'a deux trucs importants que vous devez mettre dans votre prompt :

- Format de sortie HTML (pour formater le mail)
- La clé « actu » via un drag and drop du nœud précédent (pour qu'il ait accès aux données)



Vous obtenez votre code HTML prêt à l'emploi, bien sûr vous pouvez adapter le prompt comme vous voulez, selon la paire que vous tradez si vous tradez le forex, etc.

Dernière étape, l'envoi de l'e-mail. Pour ça je vais utiliser le nœud d'action natif gmail. Bien sûr, vous pouvez utiliser outlook ou d'autres boîtes mails si vous voulez.

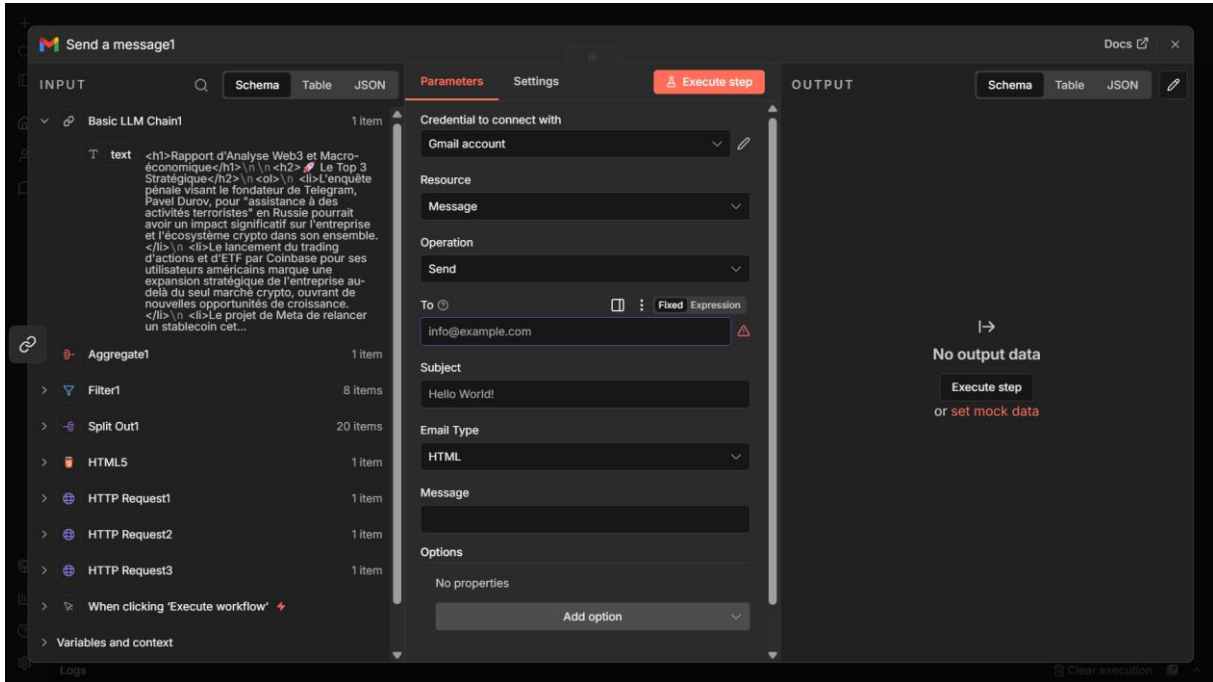
Par contre comme pour le nœud Model (celui attaché à la LLM Chain), il faut connecter les credentials. Je vais pas vous le cacher, c'est assez chiant à faire pour l'écosystème Google, mais une fois que c'est fait c'est pour toujours.

Donc je vais pas vous faire un tuto ça serait pas très pertinent par contre je peux vous recommander cette vidéo : <https://www.youtube.com/watch?v=iAqRoC-xvXI>

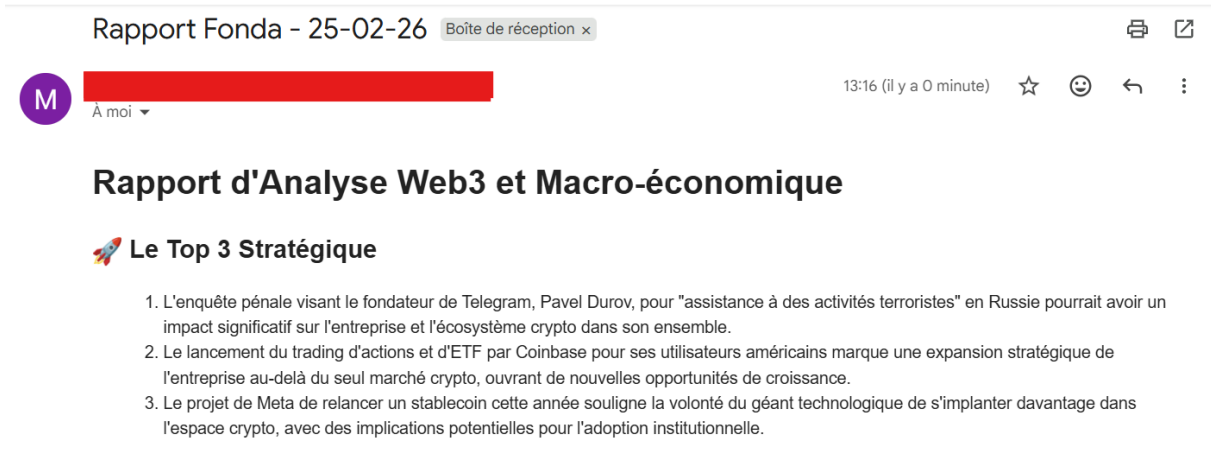
Cette vidéo est vraiment pas mal car elle montre aussi comment connecter google sheet et google calendar, ce qui est très utile.

Une fois que vous avez connecté les credentials, vous pouvez chercher le nœud gmail, et choisir « send a message ».

Voici la tête de ce nouveau nœud :



Il est assez intuitif donc je vais tout détailler, veuillez bien à drag and drop la clé « text » dans Message, après remplissez juste le destinataire et l'objet du mail et c'est bon. Toujours tester le nœud avec Execute step.



Et voilà ça marche parfaitement. Vous pouvez ajuster votre prompt si nécessaire. Enfin il ne reste plus qu'à remplacer le nœud trigger manuel par le nœud schedule trigger et publier le workflow.

Bravo vous avez votre premier workflow fonctionnel. On a vu pas mal de trucs, désolé pour la longueur du rapport d'aujourd'hui, mais maintenant que vous avez les bases, on pourra aller beaucoup plus vite pour les prochains projets et faire des trucs

plus complexes. Donc si vous avez des idées comme d'habitude, mes DM sont ouverts. De même si vous avez des questions, n'hésitez pas à passer ㄥㄹㄴ