

# Rapport de stage - Data Engineer

Polytech Nice Sophia

Lheureux Dylan - MAM4

Sous l'encadrement de Pierre Landoïn



# Table des matières

0.1	Remerciements	3
0.2	introduction	3
0.2.1	Présentation de l'entreprise & service	3
0.2.2	Contexte général du stage	3
0.3	Travail proposé	3
0.3.1	Benchmark des emails finders	3
0.3.2	Automatisation de traduction et reformulation d'articles	4
0.3.3	Tests, rédaction d'un tuto et réalisation d'une vidéo pour l'utilisation du module Icypeas sur Zapier, Make et n8n	5
0.3.4	Intégration de Zapier sur Icypeas & publication de templates sur n8n	5
0.3.5	Création d'une liste des entreprises utilisant Salesforce	5
0.3.6	Création d'un dashboard pour accueillir des podcasts sur Appsmith grâce à une table Elasticsearch	5
0.4	Travail réalisé	5
0.4.1	Benchmark des emails finder	5
0.4.2	Automatisation de traduction et reformulation d'articles	8
0.4.3	Tests, rédaction d'un tuto et réalisation d'une vidéo pour l'utilisation du module Icypeas sur Zapier, Make et n8n	10
0.4.4	Intégration de Zapier sur Icypeas & publication de templates sur n8n	11
0.4.5	Création d'une liste des entreprises utilisant Salesforce	13
0.4.6	Création d'un dashboard pour accueillir des podcasts sur Appsmith grâce à une table Elasticsearch	14
0.5	Conclusion	15
0.6	Bibliographie	15
0.7	Annexe	16
0.7.1	Podseeker	16
0.7.2	Tutoriels Zapier, Make et n8n	16
0.7.3	Templates n8n	17
0.7.4	Code permettant de faire des recherches DNS (SalesForce)	17

## 0.1 Remerciements

Je tiens à exprimer ma gratitude auprès des personnes qui ont contribué au bon déroulement de mon stage, par leurs conseils et esprit d'équipe.

Je souhaite d'abord remercier Mr. Pierre Landoïn, mon tuteur de stage pour la confiance qu'il m'a donné en me confiant ce stage, mais également pour son suivi assidû, qui m'a permis d'échanger avec lui et d'obtenir des conseils adaptés. Je le remercie également pour son attention et l'autonomie qu'il m'a laissé, cela m'a permis de gérer des problèmes seul.

Je remercie également Mr. Corentin Ribeyre qui a pu me conseiller sur des tâches plus techniques grâce à son expertise. Ses connaissances approfondies m'ont permises d'aller plus loin dans mon apprentissage.

Je tiens plus généralement à remercier toute l'équipe d'Icypeas pour leur chaleureux accueil et esprit d'entraide qui est vraiment le mot d'ordre de l'entreprise.

Merci à tous, d'avoir rendu mon stage enrichissant et de m'avoir permis d'apporter de la valeur ajoutée à l'entreprise.

## 0.2 introduction

### 0.2.1 Présentation de l'entreprise & service

Icypeas est une petite start-up basée à Paris (moins de 10 employés ) et fondée en 2022, dans le secteur du développement de logiciels.

Icypeas est spécialisée dans la découverte de données de contacts. L'entreprise développe un SaaS qui permet de trouver et vérifier des adresses emails, afin d'aider les petites et moyennes entreprises à contacter leur prospects.

### 0.2.2 Contexte général du stage

Les locaux de l'entreprise étant situés à Paris, j'ai eu l'opportunité de réaliser ce stage en distanciel, depuis chez moi à Sophia-Antipolis. Cela n'a pas été un frein puisque l'entreprise se dirige actuellement vers un "full remote" puisque la majorité des tâches peuvent se faire en ligne.

J'ai donc pu rejoindre Icypeas en tant que Data engineer où des tâches techniques m'ont été confiées. J'ai pu réaliser plusieurs tâches d'analyse et de développeur lors de mon stage, la plupart du temps en autonomie mais parfois en collaboration.

La structure de l'entreprise est assez simple, elle est constituée du CEO et du CTO qui possèdent tout deux 50% des parts de l'entreprise. L'entreprise est axée sur un management très participatif. En effet, tous les matins, nous avons un "daily" : une visio avec tous les membres de l'entreprise, stagiaires compris où chacun disait ce qu'il avait fait la veille et ce qu'il allait faire dans la journée, ce qui permettait à chacun de s'exprimer ou de proposer des axes d'amélioration au CEO et CTO qui en prenaient notes et valorisaient ce côté pro-actif, même pour les stagiaires. Le CEO prenait les décisions financières mais elles pouvaient être discutées ouvertement.

## 0.3 Travail proposé

Plusieurs tâches m'ont été confiées durant ce stage, certaines plus longues que d'autres et plus ou moins techniques. En effet, l'entreprise étant en plein développement, elle avait la nécessité d'avancer sur plusieurs tâches afin de faire face à l'attente et la demande des clients sur de nouvelles fonctionnalités.

### 0.3.1 Benchmark des emails finders

Ayant pour habitude de réaliser un benchmark annuellement, on m'a confié la tâche de réaliser ce benchmark de la manière la plus objective qui soit.

Le but était d'abord de faire une étude du prix du benchmark, Icypeas ayant un budget de 600 euros pour réaliser ce benchmark. Il faut donc sélectionner les emails finder les plus pertinents et trouver une taille d'échantillon à la fois suffisante pour que l'étude soit pertinente mais aussi qui convient au budget de l'entreprise. Une fois cela fait, une étude comparative m'a été demandé. Afin de rendre le benchmark clair auprès du public, on donnera les résultats du benchmark sous 3 indicateurs : Le taux brut de découverte d'emails, le taux net de découverte d'emails et le taux de bounce. Le taux brut de découverte d'emails étant le nombre d'emails que peut trouver un email finder

par rapport à un échantillon total de personnes. Le taux de bounce étant le taux d'emails qui ont bounce parmi ceux trouvés par un email finder, un bounce étant un message automatisé d'un système de messagerie, informant l'expéditeur d'un message précédent que le message n'a pas été livré. Moins formellement, il s'agit d'un email qui a été trouvé par l'email finder mais qui se révèle en réalité invalide. Le taux net de découverte d'emails est le taux d'emails trouvés et valides. L'objectif de ce benchmark est d'apparaître sur la page internet d'Icypeas et d'avoir un comparatif objectif des différents emails finders.

#### PERFORMANCE

## Pick The Most Efficient Email Finder Out There

We are proud to demonstrate the highest discovery rate of the entire Sales tech industry, AND the lowest bounce rate. Double whammy!

This result was obtained from a file of 300 rows containing the variables "first name", "last name", and "company name".

[Read the full report](#)

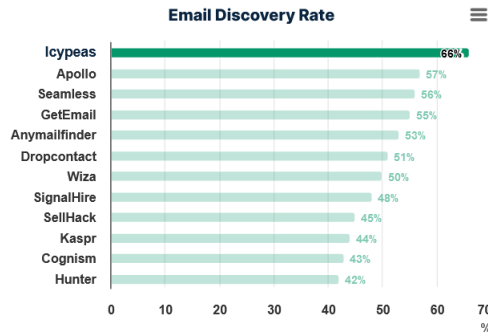


FIGURE 1 – Exemple de benchmark d'une année précédente

### 0.3.2 Automatisation de traduction et reformulation d'articles

L'entreprise souhaite tester un moyen de mettre en avant son email finder en faisant jouer le SEO (Search Engine Optimization). L'idée est la suivante : Créer un blog (sans liens avec Icypeas) constitué d'articles en lien avec les Sales, Email finder, technique marketing B2B etc... et y glisser des liens vers Icypeas afin de faire remonter Icypeas dans les recherches Google. Cependant, cela peut-être un processus très long de créer des articles pertinents que les utilisateurs liront, surtout sur de grosses quantités. On cherche donc à récupérer des articles existants sur différents blogs comme SalesDorado (en français) et de les traduire en anglais en reformulant les phrases, afin qu'elle ne soient pas détectés par une IA comme étant écrit par une IA ou du plagiat. Pour réaliser cela, le plus simple est de passer par des outils tels que n8n ou Zapier, qui permettent d'exécuter des workflow de manière automatisée. Pour réaliser cela, 4 choix s'offrent à nous : ActivePieces(nouveau sur le marché), n8n (plus technique mais plus complet), Make (moyennement technique) ou Zapier (peu technique).

PROCESS COMMERCIAUX

GUIDE

## L'ART DE LA PERSUASION DANS LA VENTE B2B

Publié le 18 décembre 2023, mis à jour le 18 décembre 2023

🕒 9 min

### SOMMAIRE

**Les 3 modes de persuasion d'Aristote**

Comprendre les motivations de son prospect avec la méthode SONCAS

Utiliser les différents principes

Dans cet article, on plonge dans les méthodes de persuasion les plus efficaces dans la vente B2B. À commencer par les modes de persuasion d'Aristote. On étudie aussi en profondeur la méthode SONCAS, même s'il s'agit en vérité surtout d'un outil de découverte, et sur les principes d'influence de l'excellent bouquin Robert Cialdini: Influence et Manipulation

Si vous voulez être plus convaincant(e), vous êtes au bon endroit.

**Axel LAVERGNE**  
Co-fondateur & rédacteur en chef

Axel est un des co-fondateurs de Salesdorado. Il est aussi le fondateur de reviewflowz, un logiciel de gestion des avis clients.

FIGURE 2 – Exemple d'article à traduire

### 0.3.3 Tests, rédaction d'un tuto et réalisation d'une vidéo pour l'utilisation du module Iypeas sur Zapier, Make et n8n

Cette tâche réponds directement aux besoins des utilisateurs. En effet, certains utilisateurs d'Iypeas souhaitaient automatiser la recherche d'emails, la vérification d'emails ou le scan de domaine.

Mr. Corentin Ribeyre ayant déjà réalisé un module Iypeas sur Zapier, Make et n8n, il faut désormais tester ce module afin de s'assurer que tout fonctionne bien et qu'on obtient bien les résultats attendus ( les mêmes que sur le site Iypeas ). L'objectif étant ensuite que je m'occupe de la soumission de ce module à une review permettant de rendre le module publique et utilisable par tous.

Dans un second temps, et après publication du module sur Zapier, Make et n8n il faut rédiger un tutoriel écrit permettant aux utilisateurs d'installer et de comprendre comment ce module fonctionne, puis de réaliser ses propres workflow. Dans un objectif de clarté totale, il faut aussi réaliser une vidéo tuto montrant comment utiliser les différentes fonctionnalités du module ( chercher un email, vérifier un email, scanner un domaine et "écouter" les résultats obtenus en temps réel sur le site d'Iypeas ).

### 0.3.4 Intégration de Zapier sur Iypeas & publication de templates sur n8n

Suite à la publication du module Iypeas sur Zapier, mon tuteur de stage m'a fait part d'une intégration possible de Zapier directement sur le site Iypeas. Cela permet aux utilisateurs de Zapier d'éviter d'avoir plusieurs onglets et de tout gérer directement depuis Iypeas. L'objectif est d'intégrer cette fonctionnalité sur Iypeas.

La deuxième tâche elle, est plus une nécessité pratique. Il s'agit de créer des templates ( workflow tout faits ) sur les fonctionnalités du module Iypeas. En effet, sur n8n il est plus complexe d'avoir son module Iypeas directement : nous devons faire appel à une requête HTTP et à un module de code JavaScript ce qui peut vite décourager les utilisateurs. Ces templates pourront directement être importés dans le workflow d'un utilisateur.

### 0.3.5 Création d'une liste des entreprises utilisant Salesforce

Dans le but d'obtenir de nouveaux clients, il est utile de multiplier l'enrichissement des prospects quand on utilise l'email finder. C'est à dire que si on décide de chercher l'email de 10 prospects, nous allons recevoir un fichier de sortie contenant l'email des prospects, mais il est toujours intéressant d'obtenir d'autres informations, comme le compte LinkedIn ou le numéro de téléphone par exemple. Ici, l'objectif est similaire, des entreprises sont intéressées de connaître quelles entreprises utilisent Salesforce car elles seront ensuite plus à même de les démarcher pour mettre en lien leurs démarches avec ce qu'utilisent leurs prospects.

Ainsi, il faut réussir à avoir une liste des entreprises qui utilisent Salesforce, à partir d'une astuce : Si on tape "nom de l'entreprise".my.salesforce.com et que la recherche aboutit, alors l'entreprise utilise Salesforce, sinon non. Ici, le challenge est d'automatiser cette recherche sur un grand nombre d'entreprises, ce qui est un défi de scalability ( plus de 80 millions ).

### 0.3.6 Création d'un dashboard pour accueillir des podcasts sur Appsmith grâce à une table Elasticsearch

Afin de poursuivre le référencement d'Iypeas, nous avons décidé de faire un site accueillant des millions de podcasts, qui serait une référence pour beaucoup d'utilisateurs, et d'y glisser des liens qui redirige vers Iypeas.

Il m'a donc été demandé de créer une table Elasticsearch contenant les données de millions de podcasts. Puis de linker cette table à Appsmith, un outil permettant de créer des dashboards/sites internet et qui a l'avantage de pouvoir se link à une table Elasticsearch, d'intégrer des widget répondants à des codes JavaScript.

Ainsi, l'objectif est de réussir à linker cette table à Appsmith et d'afficher les données des podcasts, de pouvoir filtrer les résultats, interagir avec les pages etc "à la manière de podseeker" (cf annexes).

## 0.4 Travail réalisé

### 0.4.1 Benchmark des emails finder

La première étape de ce benchmark est de faire une étude du pricing des différents emails finder afin déterminer la taille de l'échantillon que l'on va passer dans les emails finders. D'après les conseils de Mr. Pierre Landoïn qui bénéficie d'une certaine expertise sur les emails finder, plusieurs tailles d'échantillons m'ont été laissées à l'analyse :

200, 600 et 1000 prospects. J'ai donc pu chercher parmi les principaux concurrents d'Icypeas, les plus pertinents à analyser. J'ai dressé un Google Sheet répertoriant tous les prix des emails finders pour chaque échantillon puis nous avons pu choisir ensemble les emails finders les plus pertinents. Notre choix final s'est porté sur un échantillon de 600 personnes et sur ces email finders : Findemails, Aeroleads, Anymailfinder, Apollo, Wiza, Sellhack, Dropcontact, Icypeas, Getemail, Hunter, Enrow, Findymail, Prospeo.

Ainsi, j'ai pu commencer l'étude. Tout d'abord, maintenant que nous connaissons la taille de l'échantillon que nous voulons, il m'a été important de panacher les prospects le plus possible. Je me suis donc rendu sur LinkedIn (Sales Nav) et j'ai cherché des filtres pertinents. Mon choix s'est porté sur les pays et l'effectif de l'entreprise à laquelle appartient le prospect. En effet, les emails finder cherchent pour la grande majorité l'email pro d'un prospect et certains sont plus performants si les prospects proviennent de petites ou grandes entreprises. D'autres n'effectuent la recherche que si le pays du prospect est renseigné. D'autre part, cela permet d'avoir un grand éventail et représente un cas commun pour les clients des emails finders.

Ensuite, j'ai extrait : le prénom, le nom et l'entreprise du prospect car c'est ce qui est toujours demandé pour réaliser une recherche d'adresse email.

	A	B	C	D	E
1	First Name	Last Name	company		
2	Subbiah	Palaniappan	Colourful Interiors		
3	Atul	Rawat	Fusionpact Technologies Inc		
4	Rupesh	Shah	Marwiz Tech Pvt. Ltd.		
5	Varun	Kodnani	Flowace.ai - Boost Workforce Productivity by 31%		
6	Anand	Kulkarni	Videosys Software Pvt Ltd		
7	Atul	Gupta	CloudVandana Solutions		
8	Nitin	N.	ITI Long Short Equity Fund		
9	Arjoon	Mehra	Magique Creations & Services		
10	Husain	Ghadially	Stealth Startup		
11	Rajesh	Sahasrabudhe	Rezoomex		
12	Amiit	Naik	ancillarie		
13	Pradeep	Jeyaraj	Recode Solutions		
14	Darshana	Jain	InfoBeans		
15	Nishchay	Nath	Tap Invest (Formerly Leaf Round)		
16	Kammal	KKalra	VegNonVeg		
17	Vinay	Trivedi	TerraPay		
18	Deepak	K.	Xenlogic Technologies Pvt Ltd		
19	Christopher	Wakare	IntelliConnect Technologies		
20	Tushar	Garimalla	Aspire UPI Pay Later		
21	Sudeep	Sagar	Sharpe Labs		
22	Srikanth	Appana	Bajaj Auto Ltd		
23	Mathan	Jincilin R.	Harvard Business Review		
24	Shubhodeep	Das	Stealth Startup		
25	Amit	A.	Stealth Startup		
26	Madhumidha	M	ScientiaMobile		
27	Balaji	G	Indie Spirit Technologies		
	<	>	LIX_518261_2ce93c22-0815-4a6a-b	+	

FIGURE 3 – Extrait du dataset des 600 prospects

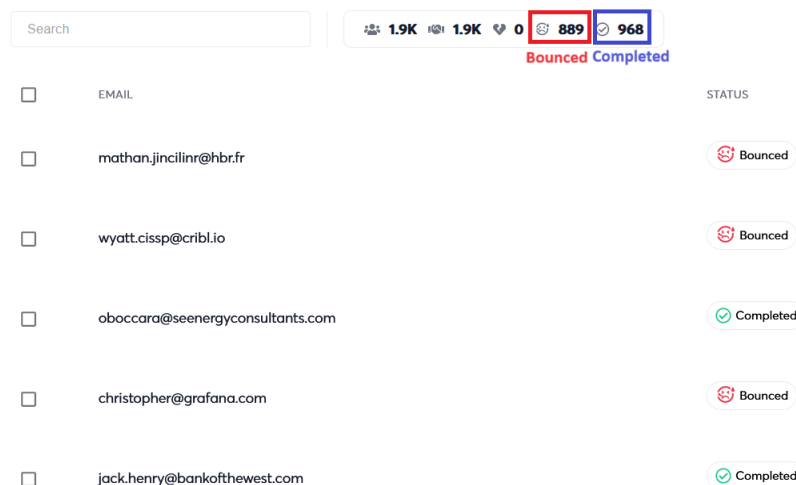
J'ai stocké le résultat de chaque email finder dans un Google Sheet, puis j'ai calculé le taux brut de découverte en prenant le nombre d'emails trouvés, divisés par le nombre total de prospects.

Afin de calculer le taux de bounce et donc le taux de découverte, il a fallu tester chacun des emails. Cela représente plus de 2000 emails pour ce benchmark ( car entre deux emails finders ils peuvent trouver un email différent pour un prospect ). D'où la nécessité d'avoir un moyen efficace de tester si les emails bounce sans avoir à le faire manuellement. J'ai donc utilisé Instantly, un service payant qui permet de réaliser des campagnes et d'automatiser l'envoi d'emails. Ici encore, il y avait un soucis car il ne fallait pas que l'email émetteur soit mis dans les spams, ce qui aurait faussé les résultats de bounce.

J'ai donc créé plusieurs adresses emails professionnelles avec Google workspace et acheté un nom de domaine pour chacune d'entre elles. Il a ensuite fallu que je configure le SPF (Sender Policy Framework), le DKIM (Do-

mainKeys Identified Mail) et le DMARC (Domain-based Message Authentication Reporting) afin que les emails n'arrivent pas dans les spams. L'avantage d'avoir un email pro est de pouvoir envoyer plus d'emails quotidiennement sans passer dans les spams. Néanmoins la limite quotidienne raisonnable d'envoi d'emails pour une personne est d'environ 120 maximum et avec plus de 2000 emails à tester (j'ai joint tous les emails différents trouvés par les emails finders puis j'ai dédoublonné) j'ai donc décidé de créer et configurer 6 adresses mails, ce qui est un processus assez long mais avantageux. Pour configurer le SPF, DKIM et DMARC je me suis principalement aidé de tutoriels disponibles sur internet. Après avoir relié mes boîtes mails émetteuses à Instantly, j'ai configuré l'email à envoyer qui devait une nouvelle fois être crédible, pour ne pas être signalé par les receveurs. J'ai donc rédigé le mail comme un stagiaire qui cherchaient un stage dans l'entreprise du prospect et j'ai lancé la campagne.

Au moment de passer en entrée les quelques 2000 emails à Instantly, le logiciel m'indiquait qu'il y avait 12 emails invalides et j'ai donc cherché pourquoi ils étaient invalides et je suis parvenu à la conclusion suivante après analyse des données : Certains emails finder dont Icypeas trouvent des emails invalides du type exemple.@icypeas.com. J'ai donc pu transmettre à Mr. Corentin Ribeyre qui a pu noter la remarque et s'occuper du problème, car ce format d'email est impossible et ne devrait même pas être renvoyé dans le fichier de sortie. Une fois la campagne terminée, Instantly nous donne directement la liste des emails et nous dit s'ils ont bounce ou non.



EMAIL	STATUS
<input type="checkbox"/> mathan.jincilnr@hbr.fr	Bounced
<input type="checkbox"/> wyatt.cissp@cribl.io	Bounced
<input type="checkbox"/> oboccarra@seenergyconsultants.com	Completed
<input type="checkbox"/> christopher@grafana.com	Bounced
<input type="checkbox"/> jack.henry@bankofthewest.com	Completed

FIGURE 4 – Résultat de la campagne Instantly

Mr. Corentin Ribeyre m'a ensuite passé un script python issu des précédents benchmark qui prends le fichier résultat des emails finders en entrée, ainsi que le fichier résultat d'instantly avec les status de bounce. L'algorithme permet de faire le tri parmi le statut des quelques 2000 emails, et d'indiquer pour chaque email finder, parmi les emails qu'il a trouvé, lesquelles ont bounced ou non.

Ainsi, j'ai pu calculer le taux de bounce de chaque email finder, puis le taux net de découverte. J'ai donc réalisé 3 graphiques présentant ces taux et j'ai pu valider ces résultats avec mon tuteur de stage qui a été agréablement surpris de constater qu'Icypeas est 2e en taux net de découverte et possède le plus petit taux de bounce.

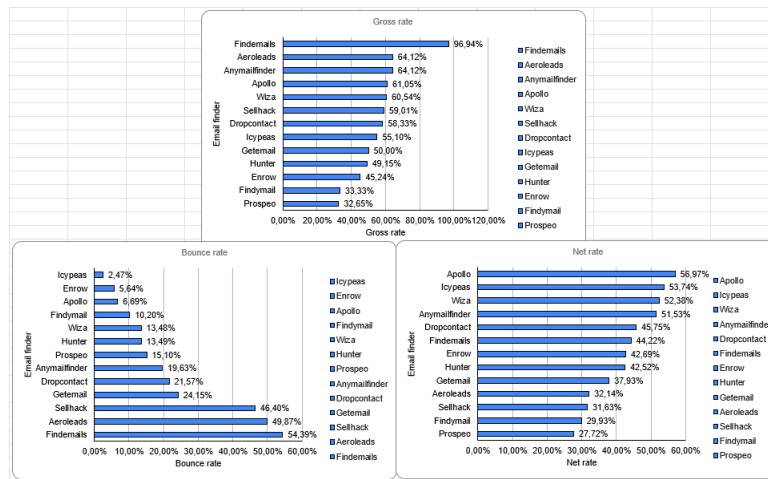


FIGURE 5 – Résultats du benchmark : histogramme

	aeroleads	anymailfinder	apollo	dropcontact	enrow	findemails	findymail	getemail	hunter	icypeas	prospero	selhack	witza
Count TRUE :	377	377	359	343	266	570	196	294	289	324	192	347	356
Taux bruts :	64.11564626	64.11564626	61.05442177	58.33333333	45.23809524	96.93877551	33.33333333	50	49.14965986	55.10204082	32.65306122	59.01360544	60.54421769
Count TRUE sachant bounced :	188	74	24	74	15	310	20	71	39	8	29	161	48
Taux bounces :	49.87%	19.63%	6.69%	21.57%	5.64%	54.39%	10.20%	24.15%	13.49%	2.47%	15.10%	46.40%	13.48%
Taux nets :	32.14285714	51.53061224	56.97278912	45.74829932	42.68707483	44.21768707	29.93197279	37.92517007	42.5170068	53.7414996	27.72108844	31.63265306	52.38095238
	10e	4e	1e	5e	7e	6e	12e	9e	8e	2e	13e	11e	3e

FIGURE 6 – Résultats du benchmark

## 0.4.2 Automatisation de traduction et reformulation d'articles

Pour réaliser cette tâche, qui paraît simple mais est complexe à automatiser je suis resté prudent. Après avoir extrait avec un outil de scraping les URL des différents articles de SalesDorado. J'ai décidé d'analyser le format général des pages de blog et de traduire manuellement 10 articles avec l'aide de chatGPT afin d'identifier les difficultés rencontrées.

La première étant que je n'ai jamais fait de Javascript auparavant et que j'ai donc dû me renseigner sur le format des pages internet afin d'extraire au mieux les parties de la page internet qui nous intéressent. La seconde difficulté étant que le nombre de caractères dans un prompt ChatGPT est limité, alors que certains articles sont très longs. J'ai donc identifié un autre soucis : il faut diviser le texte des articles en plusieurs parties puis le réassembler. En relisant la traduction des articles, je me suis également aperçu qu'elle partait dans tous les sens sur les dernières lignes lorsque le prompt était trop long. J'ai donc identifié un autre problème : chaque prompt ne doit pas être trop long. Je suis donc parti sur des prompts de 1000 caractères.

Après avoir effectué cette analyse, j'ai du décider avec quelle plateforme automatiser cette tâche. Mon tuteur de stage a souhaité que j'essaie d'abord sur ActivePieces car il a pu bénéficier d'un "lifetime deal" car il s'agit d'une nouvelle plateforme. Néanmoins, je n'ai pas pu trouver de manière simple d'utiliser OpenAI (API de ChatGPT) ni de parser du html sur cette plateforme. Je me suis donc redirigé vers n8n, possédant un module "OpenAI" et étant très complet.

Ici encore, j'ai préféré rester prudent et commencer par réaliser un workflow qui me permet de traduire un seul article, sans faire de boucles, afin de déceler des difficultés et éviter de trop dépenser ( Open AI est payant, pay as you go c'est à dire que plus on lance de prompt, plus on paye ).

J'ai commencé par définir les étapes du workflow : Récupérer un URL d'un Sheet -> Faire une requête HTTP pour récupérer les bonnes parties de l'article -> récupérer l'URL des images de l'article et le séparer du reste de l'article mais en conservant l'ordre -> diviser le texte de l'article tous les 1000 caractères (plus ou moins pour ne pas casser de phrases ou de mots) -> créer un Array contenant un prompt disant de traduire l'article sans être détectable par une IA suivi du contenu de l'article (divisé ou non selon l'article) -> Passer les prompts sur OpenAI -> Récupérer les résultats pour chaque prompt dans un Array -> Réassembler le résultat de chaque prompt dans l'ordre, et ajouter l'URL de l'article tout en haut, suivi des URL des images de l'article dans un Google Doc.



J'ai ensuite pu passer à la réalisation de workflow sur n8n. Les principaux défis pour ce workflow pour moi ont été de séparer l'URL des images de l'article pour le réassembler à la fin. Ce problème résidant surtout dans la structure de n8n qui nécessite de créer des item lists pour fusionner le résultat de deux modules. Le deuxième défi pour moi a été de traiter le résultat de la requête HTTP, qui rendait principalement le contenu de l'article mais où il fallait diviser ce texte tous les 1000 caractères, ici le défi était de pouvoir s'arrêter un peu avant ou après ces 1000 caractères selon où on se trouve (au milieu d'un mot ou d'une phrase) et séparer l'URL du reste de l'article. Pour faire ceci, le seul moyen est d'utiliser des modules de code, en Javascript. J'ai donc passé un peu de temps pour cela car je n'ai pas de connaissances en JavaScript.

Néanmoins je suis parvenu à obtenir le résultat escompté assez rapidement et j'ai donc pu passer à la prochaine étape : Faire le même workflow mais cette fois-ci valable pour plusieurs articles (boucle).

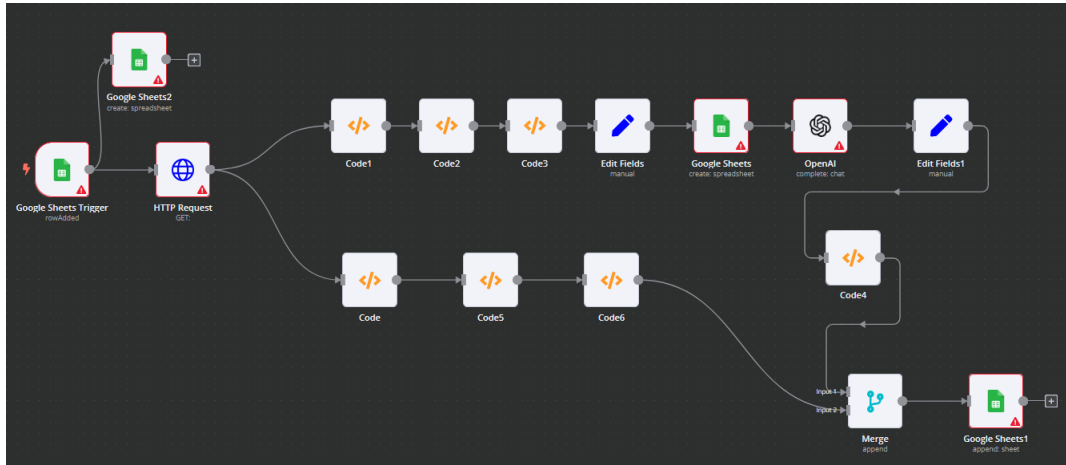


FIGURE 7 – Workflow pour traduire et reformuler un article

J'ai identifié le problème le plus important de la version avec boucle : Afin de donner les prompts à Open AI je créais un Sheet avec un nom donné dans mon drive. Cela ne posait pas de soucis pour un article mais ici, si on décide de traduire 2000 articles, on va vite se retrouver avec des problèmes de mémoire. Il fallait donc prévoir un module pour supprimer ce Sheet après l'exécution d'OpenAI. Il fallait aussi prévoir le temps d'exécution du workflow qui peut être assez long, principalement à cause du temps d'exécution de la node OpenAI ( le temps d'écrire le résultat ). Il vaut mieux être prudent et ne pas soumettre plus de 100 articles à la fois si un problème serveur survient sur OpenAI ( ce qui est assez fréquent ).

J'ai donc pu réaliser le workflow et le tester sur une centaine d'articles, avec succès.

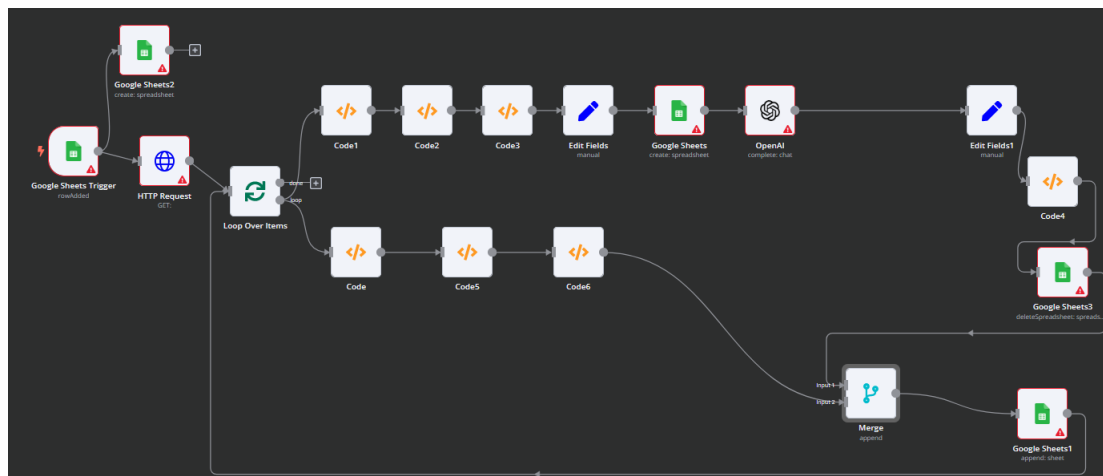
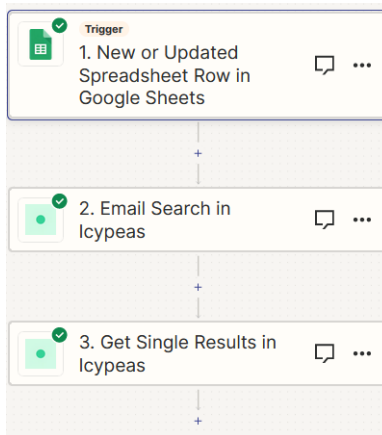


FIGURE 8 – Workflow pour traduire et reformuler plusieurs articles

### 0.4.3 Tests, rédaction d'un tuto et réalisation d'une vidéo pour l'utilisation du module Icyneas sur Zapier, Make et n8n

Tout d'abord, j'ai dû tester le module Icyneas sur Zapier, Make et n8n. J'ai donc réalisé des workflows très basiques, servant à tester chacune des fonctionnalités.



Je n'ai pas décelé de problème d'exécution lors de mes tests, tous mes workflow renvoyaient le résultat attendu. J'ai donc pu passer à une étape importante et très attendue, la publication du module au grand public.

J'ai pu discuter avec l'équipe Make qui a fait ma review et j'ai apporté les correctifs nécessaires. Le principal correctif qui a été demandé était l'absence d'interface qui "mappe" les résultats obtenus. Plus simplement, lorsqu'on faisait une recherche d'email sur Make par exemple, et qu'on voulait utiliser le résultat de cette recherche dans un autre module ( par exemple stocker le résultat dans un Sheet ) on était obligé d'exécuter au moins une fois le workflow, sans exécuter le noeud "Sheet" car on ne pouvait pas directement sélectionner "email". La création de cette interface se fait au sein même de Make, et se fait sous le format .json. J'ai donc dû une nouvelle fois faire face un petit soucis de connaissance puisque je ne connaissais pas le format des .json auparavant. J'ai finalement pu apporter les modifications nécessaires et publier le module avec succès.

```
</> jsonc
1 // Defines JSON object with "id", "email" and "name" parameters as expected API response body.
2 [
3   {
4     "name": "success",
5     "type": "boolean",
6     "label": "Success"
7   },
8   {
9     "name": "items",
10    "type": "array",
11    "spec": {
12      "type": "collection",
13      "spec": [
14        {
15          "name": "name",
16          "type": "text"
17        },
18        {
19          "name": "user",
20          "type": "text"
21        },
22        {
23          "name": "results".
```

FIGURE 9 – .json pour modifier l'interface

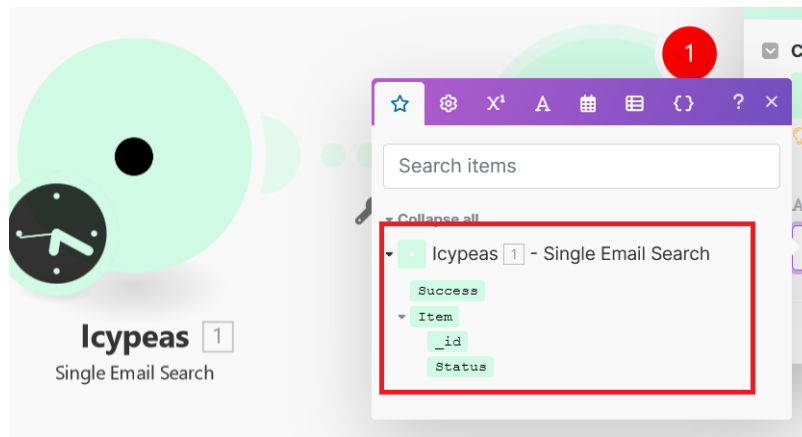


FIGURE 10 – Mapping fonctionnel (id disponible)

La publication du module sur Zapier et n8n n'a demandé que des changements mineurs comme la modification du format du logo, la mise en place d'un accès total à tous les accès payants d'Icypeas afin qu'un développeur vérifie le bon fonctionnement du module.

J'ai également dû réaliser des workflow de "test" utilisant toutes les fonctionnalités du module, afin de faciliter la vérification du module par les développeurs.

Une fois le module publié sur toutes les plateformes. J'ai dû réaliser un tutoriel écrit destiné à aider les utilisateurs à utiliser le module Icypeas. C'est un exercice qui m'a tenu à coeur car nous avons tous déjà regardé, lu un tutoriel qui était parfois ambigu ou qui ne répondait pas à notre question. J'avais donc la volonté de bien faire et d'être très précis et exhaustif en me mettant à la place d'un utilisateur.

De plus, j'ai réalisé une vidéo tuto sur les 3 plateformes en anglais, ce qui m'a permis de confirmer mon niveau d'anglais.

Vous trouverez en annexe des liens directs vers les tutoriels réalisés. J'ai pu obtenir, un retour lors de mon dernier jour de stage, sur l'utilisation du tutoriel. Mon tuteur de stage m'a informé que mon tutoriel avait été apprécié par une dizaine de personnes. C'est une tâche que j'ai particulièrement appréciée car elle mêlait différents aspects, JavaScript, anglais, relationnel etc.. et où j'ai pu voir l'impact direct de mes tutoriels.

#### 0.4.4 Intégration de Zapier sur Icypeas & publication de templates sur n8n

Pour cette tâche, je me suis d'abord informé sur la manière de réaliser cette intégration en regardant la documentation Zapier. Je me suis aperçu qu'ils avaient un générateur qui permettait d'intégrer cette outil. Le générateur fournit le header et le body. Il suffit de choisir les couleurs et les widgets que l'on souhaite ajouter et de générer le code. Après avoir fait cela, j'ai pu fournir le code à un développeur web de l'entreprise afin d'ajouter cela au site Icypeas.

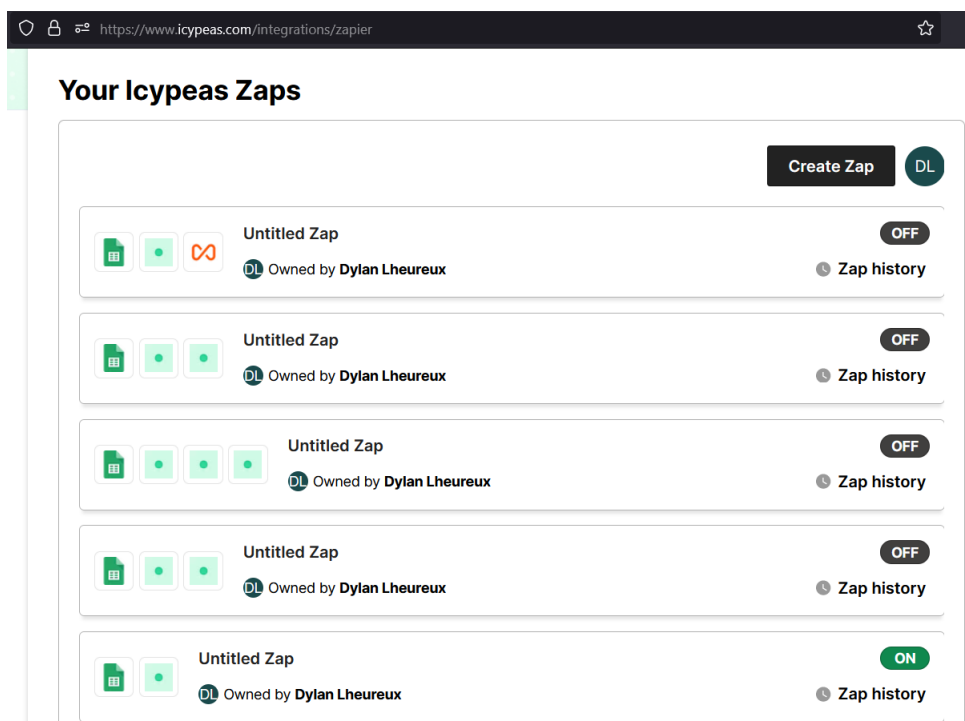


FIGURE 11 – Intégration Zapier au sein d'Icypeas

Par la suite, j'ai pu réaliser des "templates" sur n8n. C'est à dire des workflows tout faits et prêts à l'emploi. Après avoir lancé la review, j'ai dû changer quelques choses. Les principaux changements concernent l'universalité des templates, et plus précisément, rendre les templates utilisables par les utilisateurs qui travaillent sur le cloud et ceux qui sont en self-host. Pour cela j'ai dû changer les noeuds de code afin d'éviter d'utiliser la librairie URL, et j'ai également dû donner des instructions pour ajouter la librairie "crypto" en self-host. A la suite de cela, mes templates ont été approuvés et publiés (cf annexes).

```

Authenticates to your Icypeas account
Parameters Docs
1 const BASE_URL = "https://app.icypeas.com";
2 const PATH = "/api/domain-search";
3 const METHOD = "POST";
4
5 // Change here
6 const API_KEY = "PUT_API_KEY_HERE";
7 const API_SECRET = "PUT_API_SECRET_HERE";
8 const USER_ID = "PUT_USER_ID_HERE";
9 ///////////////
10
11 const genSignature = (
12   path,
13   method,
14   secret,
15   timestamp = new Date().toISOString()
16 ) => {
17   const Crypto = require('crypto');
18   const payload = `${method}${path}${timestamp}`.toLowerCase();
19   const sign = Crypto.createHmac("sha1", secret).update(payload).digest("hex");
20
21   return sign;
22 };
23
24 const fullPath = `${BASE_URL}${PATH}`;
25 $input.first().json.api = {
26   timestamp: new Date().toISOString(),
27   secret: API_SECRET,
28   key: API_KEY,
29   userId: USER_ID,
30   url: fullPath,
31 };
32 $input.first().json.api.signature = genSignature(PATH, METHOD, API_SECRET, $input.first().json.api.timestamp);
33 return $input.first();

```

FIGURE 12 – Remplacement de la librairie URL en rajoutant l'URL au PATH

### 0.4.5 Création d'une liste des entreprises utilisant Salesforce

Afin de réaliser cette tâche, j'ai tout d'abord effectué quelques tests par moi-même sur des entreprises afin de voir les résultats.

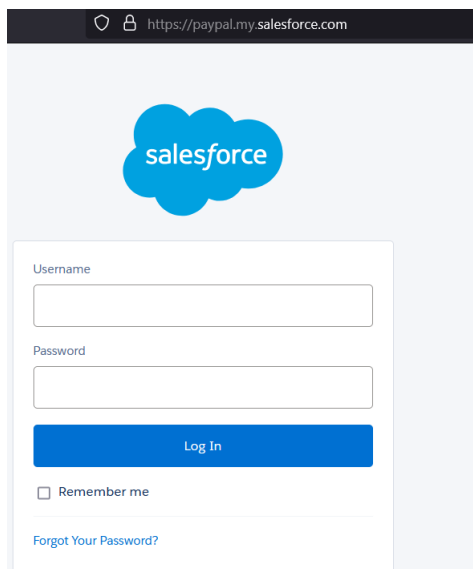


FIGURE 13 – Entreprise utilisant Salesforce : Paypal

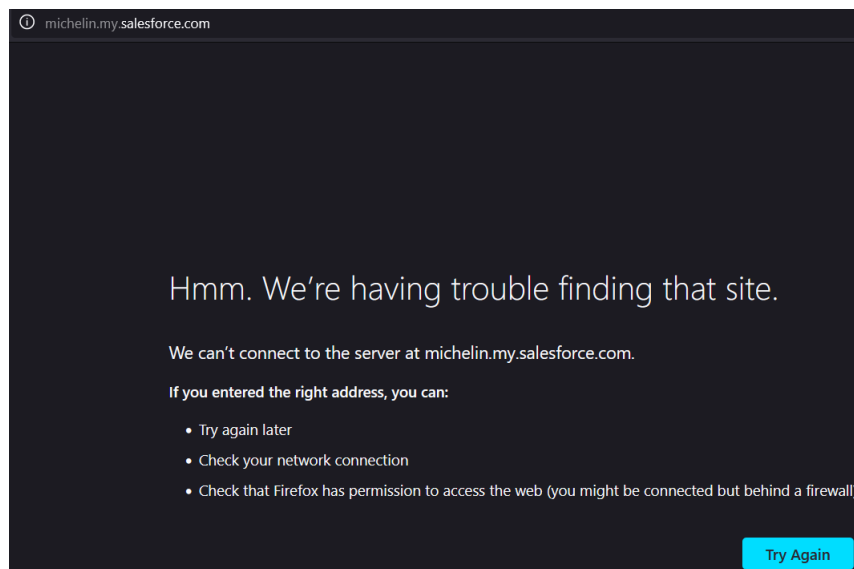


FIGURE 14 – Entreprise n'utilisant pas Salesforce : Michelin

Afin de pouvoir automatiser le même schéma sur plusieurs millions d'entreprises, il fallait regarder la réponse DNS lorsqu'on effectuait la recherche. J'ai donc décidé de faire cela en python. J'ai passé un moment à essayer de faire fonctionner les recherches DNS sur Python mais je n'obtenais pas de résultats. J'ai donc décidé de faire un code sur VBA dans mon Google Sheet et ce dernier effectuait bien la recherche mais j'avais un souci : la scalability. Non seulement les recherches étaient longues mais en plus, afficher uniquement 1 millions d'entreprises et faire la recherche faisait tout planter. J'en ai donc parlé au CTO de l'entreprise qui m'a conseillé de faire cela en Python, et m'a conseillé d'utiliser une autre librairie que celle que j'utilisais pour faire les recherches DNS. J'ai donc pu continuer sur Python car les recherches DNS fonctionnaient. J'ai également fait d'autres petits algorithmes pratiques, pour lire une ligne d'un csv, tronquer un csv, remplacer les espaces par " ou '-' afin de tester plusieurs combinaisons possibles. En effet, un URL ne peut pas contenir d'espaces. Or, parmi ma liste d'entreprises, certaines

était composées de plusieurs mots. J'ai donc testé les deux combinaisons les plus probables : Tout coller : "Alpes Habitat" -> "AlpesHabitat" et ajouter un tiret : "Alpes Habitat" -> "Alpes-Habitat".

Cependant, j'avais là aussi un problème de scalability. Il fallait 20 jours d'exécution pour traiter 1 million d'entreprises car les recherches DNS se faisaient les unes à la suite des autres. J'ai donc essayé d'appliquer le parallel computing afin de faire plusieurs recherches DNS simultanément. En effet, ayant vu cela lors de mon semestre d'échange en Belgique en Scientific Computing, j'ai vu que cela était très puissant pour résoudre des problèmes comportant un très grand nombre de données. Cela étant fait, mon exécution ne mettait plus que 48h ce qui est très acceptable au vu du temps d'une recherche DNS et du PC que j'ai utilisé. J'ai donc pu lancer cet algorithme et transmettre les algorithmes à un autre stagiaire afin qu'il puisse poursuivre l'exécution de ce code sur les millions d'entreprises restantes. Le code permettant de faire les recherches DNS et de créer un CSV avec les résultats est disponible en annexe.

#### 0.4.6 Création d'un dashboard pour accueillir des podcasts sur Appsmith grâce à une table ElasticSearch

Travail réalisé

Cette tâche est très technique et m'a été donnée à la fin de mon stage, je n'ai donc pas pu la terminer. Néanmoins j'ai pu créer ma table sur ElasticSearch en important les données.

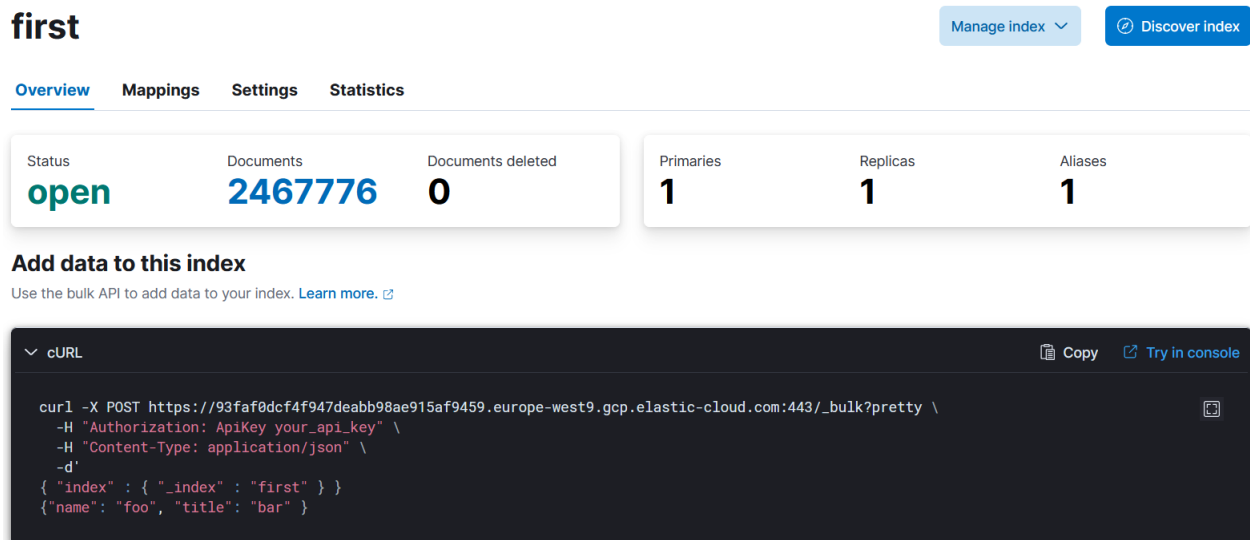


FIGURE 15 – Données de podcasts intégrés sur ElasticSearch : plus de 2m de données

J'ai également pu linker ma table à ElasticSearch en utilisant l'URL de ma table et ma clef API. Ensuite, j'ai essayé d'afficher ma table sur Appsmith à l'aide d'une widget table et en effectuant une requête POST directement. Je me suis aperçu que ma requête aboutissait mais qu'elle n'avait pas le bon format pour être affichée (Array<Object>). J'ai donc décidé de coupler ma requête à un code JavaScript afin de modifier le format du résultat. J'ai donc pu afficher les données dans ma widget table. Néanmoins je me suis aperçu que je ne pouvais pas afficher plus de 10 000 résultats, et que la norme est plutôt d'une trentaine de résultats par page, et d'ensuite laisser à l'utilisateur le choix de scroller ou changer de page afin d'afficher plus de résultats. Je me suis donc renseigné sur ces deux possibilités et j'ai choisi la méthode "search\_after" qui consiste à charger plus de résultats lorsque l'utilisateur clique sur un bouton. Néanmoins après plusieurs essais infructueux je ne suis pas parvenu à charger de nouveaux résultats en appuyant sur un bouton. Le lendemain et pour mon dernier jour de stage j'ai donc décidé de changer d'approche et d'utiliser le "scroll API" qui consiste à scroller pour afficher les prochains résultats. Mais ici aussi je n'ai pas réussi à obtenir le résultat escompté. J'ai donc consigné ce que j'avais testé dans un document afin que mon travail puisse être repris facilement.

image	podcastid	listeners	rating	description	active
https://is4-ssl.mzst...	1	Inactive	N/A	obrolan dan curha...	
https://is4-ssl.mzst...	2	Inactive	5.0	A	
https://is2-ssl.mzst...	3	Inactive	5.0	Listen to interview...	
https://is1-ssl.mzst...	4	Inactive	N/A	In this podcast Gu...	
https://is2-ssl.mzst...	5	Inactive	N/A	A group of friends ...	
https://is1-ssl.mzst...	6	Inactive	N/A	A-TI is a podcast t...	
https://is5-ssl.mzst...	7	Inactive	N/A	Cuento "Reflejo d...	
https://is4-ssl.mzst...	8	Inactive	N/A	B BlendIT Мы выг...	
https://is4-ssl.mzst...	9	Inactive	5.0	Mantasha	
https://is2-ssl.mzst...	10	Inactive	N/A	Aac 3 de ingles	
https://is2-ssl.mzst...	11	Inactive	N/A	Hello everyone! W...	
https://is3-ssl.mzst...	12	Inactive	3.67	A SOLO school g...	

FIGURE 16 – Résultat d’une recherche ”all”

## 0.5 Conclusion

En définitive, ce stage a été ma première expérience professionnelle dans les sciences des données. Il m’a permis de mettre en application mes connaissances en codage, anglais et relationnel. Il m’a aussi permis de découvrir des outils d’automatisation très puissants comme n8n, Make et Zapier, largement utilisés dans les entreprises pour automatiser des tâches. Il m’a également apporté quelques challenges puisqu’étant en distanciel, j’ai été plus livré à moi-même et ai dû résoudre la majorité de mes problèmes seuls, permettant de développer mon autonomie.

D’autre part, ce stage m’a permis de faire connaissance avec une équipe jeune et dynamique, travaillant dans la bonne humeur et la coopération, des valeurs qui sont pour moi essentielles et que j’aimerais retrouver dans le futur dans mes prochaines expériences professionnelles.

## 0.6 Bibliographie

Icypeas : <https://www.icypeas.com/>  
 Appsmith : <https://www.appsmith.com/>  
 Elasticsearch : <https://www.elastic.co/fr/elasticsearch>  
 n8n : [https://n8n.io/?utm\\_source=paid\\_google&utm\\_medium=cpc\\_google&utm\\_campaign=20722323063&adgroup=153737670126&matchtype=e&network=g&keyword=n8n&gclid=Cj0KCQiAhc-sBhCEARIsA0VwHuTbpE-55F7-27e1LJa\\_ot1stYDGXeF0awdD75W1zuNK\\_Dar3xfTXoaAvOHEALw\\_wcB](https://n8n.io/?utm_source=paid_google&utm_medium=cpc_google&utm_campaign=20722323063&adgroup=153737670126&matchtype=e&network=g&keyword=n8n&gclid=Cj0KCQiAhc-sBhCEARIsA0VwHuTbpE-55F7-27e1LJa_ot1stYDGXeF0awdD75W1zuNK_Dar3xfTXoaAvOHEALw_wcB)  
 Make : <https://www.make.com/en>  
 Zapier : <https://zapier.com/app/dashboard>  
 SalesDorado : <https://salesdorado.com/>

## 0.7 Annexe

### 0.7.1 Podseeker

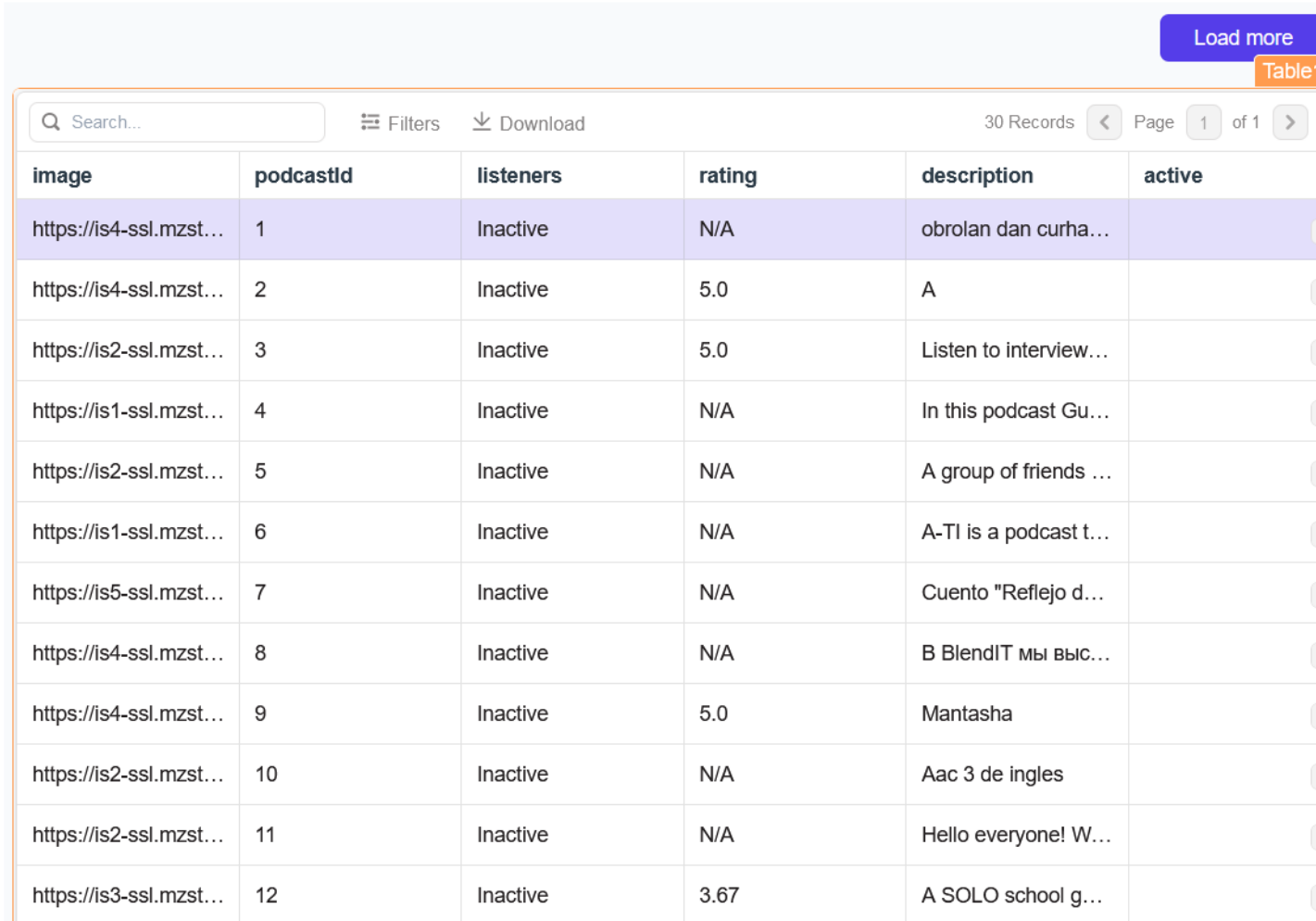


image	podcastid	listeners	rating	description	active
https://is4-ssl.mzst...	1	Inactive	N/A	obrolan dan curha...	
https://is4-ssl.mzst...	2	Inactive	5.0	A	
https://is2-ssl.mzst...	3	Inactive	5.0	Listen to interview...	
https://is1-ssl.mzst...	4	Inactive	N/A	In this podcast Gu...	
https://is2-ssl.mzst...	5	Inactive	N/A	A group of friends ...	
https://is1-ssl.mzst...	6	Inactive	N/A	A-TI is a podcast t...	
https://is5-ssl.mzst...	7	Inactive	N/A	Cuento "Reflejo d...	
https://is4-ssl.mzst...	8	Inactive	N/A	B BlendIT мы выс...	
https://is4-ssl.mzst...	9	Inactive	5.0	Mantasha	
https://is2-ssl.mzst...	10	Inactive	N/A	Aac 3 de ingles	
https://is2-ssl.mzst...	11	Inactive	N/A	Hello everyone! W...	
https://is3-ssl.mzst...	12	Inactive	3.67	A SOLO school g...	

FIGURE 17 – Interface de Podseeker

### 0.7.2 Tutoriels Zapier, Make et n8n

Vous trouverez ci-joint les liens redirigeant vers les tutoriels Zapier, Make et n8n réalisés ainsi qu'aux vidéos ( la vidéo n8n n'a pas encore été publiée car elle n'a pas été montée à la date où je fais ce rapport ).

Zapier : <https://www.icypeas.com/knowledge-base-article/how-to-use-icypeas-on-zapier>

Make : <https://www.icypeas.com/knowledge-base-article/how-to-use-icypeas-on-make>

n8n : <https://www.icypeas.com/knowledge-base-article/how-to-install-icypeas-node-on-n8n>



### 0.7.3 Templates n8n

Vous trouverez ci-joint deux templates n8n que j'ai réalisé et qui sont publics :

<https://n8n.io/workflows/2013-perform-an-email-search-with-icypeas-single/>

<https://n8n.io/workflows/2012-perform-a-domain-search-with-icypeas-single/>

### 0.7.4 Code permettant de faire des recherches DNS (SalesForce)

```
1  import csv
2  import dns.resolver
3  from dns.exception import DNSException
4  import string
5  from concurrent.futures import ThreadPoolExecutor
6  import threading
7
8  # Function to clean the company name and perform the DNS test
9  def perform_dns_test(company):
10     # Remove non-ASCII characters
11     clean_company = ''.join(char for char in company if char in string.printable)
12
13     full_domain = f"{clean_company}.my.salesforce.com"
14
15     try:
16         dns.resolver.query(full_domain, 'A')
17         return True
18     except dns.resolver.NXDOMAIN:
19         return False
20     except DNSException:
21         return False
22
23 # Input and output CSV file names
24 input_csv_file = 'C:\\Users\\hp\\Desktop\\lem\\pretest2.csv'
25 output_csv_file = 'C:\\Users\\hp\\Desktop\\lem\\pretest3.csv'
26
27 # Open the input CSV file in read mode with UTF-8 encoding
28 with open(input_csv_file, 'r', encoding='utf-8') as csv_input_file:
29     # Read the CSV file
30     reader = csv.reader(csv_input_file)
```

FIGURE 18 – Interface de Podseeker

```

32     # Load the companies from the first column
33     companies = [row[0] for row in reader]
34
35     # Open the output CSV file in write mode with UTF-8 encoding
36     with open(output_csv_file, 'w', newline='', encoding='utf-8') as csv_output_file:
37         # Create a CSV writer object
38         writer = csv.writer(csv_output_file)
39
40         # Write the header to the output file
41         writer.writerow(['Company', 'Result'])
42
43         # Function to perform DNS test and write the result
44         def process_company(idx, company):
45             result = 'TRUE' if perform_dns_test(company) else 'FALSE'
46             print(f"Processed line {idx}")
47             # Use a lock to synchronize access to the shared writer
48             with lock:
49                 # Write to CSV only if the result is TRUE
50                 if result == 'TRUE':
51                     writer.writerow([company, result])
52
53
54         # Use ThreadPoolExecutor to parallelize DNS queries
55         with ThreadPoolExecutor() as executor:
56             # Enumerate through companies and submit tasks to the executor
57             lock = threading.Lock()
58             futures = [executor.submit(process_company, idx, company) for idx, company in enumerate(companies, start=1)]
59
60         # Wait for all tasks to complete

```

FIGURE 19 – Interface de Podseeker

```

61         for future in futures:
62             future.result()
63
64     print(f"The results have been saved in '{output_csv_file}'.")

```

FIGURE 20 – Interface de Podseeker