

# RAPPORT DE STAGE

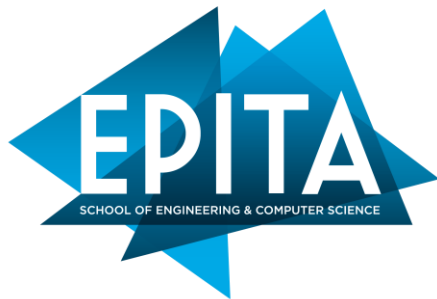
---

Nom : Ducrocq

Prénom : Tanguy

Promotion : EPITA - Promo 2026

Sujet de stage : Création et implémentation d'un agent intelligent pour la rédaction automatique de documentation technique



**DILI TRUST**

Nom du maître de stage : Vial

Prénom du maître de stage : Romain

Titre : Head of Machine Learning

Signature : \_\_\_\_\_



## Sommaire

1	Introduction .....	3
2	Présentation de l'entreprise.....	4
2.1	Le secteur d'activité .....	4
2.2	L'entreprise.....	4
2.3	Le service .....	5
2.4	Le positionnement du stage dans les travaux de l'entreprise .....	5
3	Travail effectué .....	6
3.1	Cahier des charges .....	6
3.1.1	But Général .....	6
3.1.2	Explication détaillée des résultats à obtenir .....	6
3.2	Compte-rendu d'activité.....	8
3.2.1	Axes d'étude et de recherche choisis.....	8
3.2.2	Déroulement concret des études, expérimentations, mises au point... ..	8
3.3	Interprétation et critique des résultats .....	11
3.3.1	Résultats obtenus.....	11
3.3.2	Limites et axes d'amélioration .....	11
4	Conclusion générale.....	13
5	Bibliographie & Glossaire.....	14
6	Annexes.....	16

## 1 Introduction

Ce stage a été obtenu grâce à une candidature spontanée motivée par l'intérêt que je porte à cette entreprise, particulièrement reconnue pour son expertise dans les domaines informatiques et, plus spécifiquement, en Intelligence Artificielle (IA). Cette spécialisation en IA rejoint mes intérêts personnels, et m'a encouragé à intégrer le service de Machine Learning (ML), une opportunité pour approfondir mes connaissances dans ce domaine.

Le sujet de mon stage consiste en la mise en place d'un agent de documentation pour assister dans la rédaction de la documentation pour le Crédit d'Impôt Recherche (CIR) à destination des projets internes de l'entreprise. Ce projet se déroulera principalement en Python, et impliquera l'intégration de l'API de ChatGPT, afin de générer des contenus documentaires précis et adaptés aux besoins du pôle R&D.

Mes attentes pour ce stage concernent non seulement le développement technique autour des technologies IA et ML, mais aussi l'intégration dans la vie professionnelle en entreprise, en découvrant le fonctionnement des réunions, la gestion des projets, ainsi que les processus de suivi et de coordination.

Les grandes lignes de déroulement de ce stage incluent :

1. Présentation de l'équipe ML avec laquelle je vais collaborer durant toute la période du stage.
2. Intégration dans les bureaux : mise en place des outils informatiques nécessaires, configuration de l'environnement de travail, et familiarisation avec les outils de communication internes.
3. Présentation approfondie du sujet de stage pour cadrer les objectifs et les livrables attendus.
4. Participation à une réunion d'accueil avec le Directeur Général, permettant une introduction détaillée à l'entreprise, sa mission, et ses valeurs.
5. Début des travaux autour du sujet de stage, avec un suivi structuré pour garantir l'alignement avec les besoins de l'équipe et du pôle R&D.

Je suis enthousiaste à l'idée de contribuer à un projet innovant qui apportera un soutien concret à l'équipe R&D, tout en enrichissant mon expérience et mes compétences en IA et en Machine Learning.

## 2 Présentation de l'entreprise

### 2.1 Le secteur d'activité

DiliTrust évolue dans le secteur de la LegalTech, domaine qui associe technologie et droit pour optimiser les processus des services juridiques des entreprises. Ce secteur en pleine expansion répond à la demande croissante d'outils numériques permettant aux équipes juridiques d'améliorer leur productivité, leur réactivité et leur conformité. Les solutions LegalTech comme celles de DiliTrust facilitent la gestion des contrats, des litiges et de la gouvernance d'entreprise, tout en garantissant la sécurité des données sensibles. Face aux pressions pour la digitalisation des tâches juridiques, les solutions SaaS (Software as a Service) se révèlent essentielles pour centraliser les documents et automatiser les tâches répétitives, permettant aux équipes juridiques de se concentrer sur des missions à plus forte valeur ajoutée. Pour résumer en une phrase, le rôle de DiliTrust consiste à numériser les procédures juridiques des entreprises.

### 2.2 L'entreprise

DiliTrust, entreprise française spécialisée en édition de logiciels SaaS, se distingue par une offre unifiée et sécurisée destinée aux directions juridiques. Sa plateforme couvre l'ensemble des besoins des services juridiques, incluant la gestion des conseils d'administration, le corporate (droit des sociétés), le contract lifecycle management (CLM) et la gestion des contentieux. Cette approche complète favorise la digitalisation des fonctions juridiques en intégrant des solutions d'intelligence artificielle pour automatiser les tâches répétitives et assurer la conformité.

Depuis plusieurs années, DiliTrust connaît une forte croissance avec une progression moyenne de 30 % par an. Elle compte actuellement plus de 2 000 clients répartis dans le monde entier (Europe, Amérique du Nord et du Sud, Afrique, Moyen-Orient), et prévoit de poursuivre son expansion internationale en ouvrant de nouvelles implantations. DiliTrust attache une grande importance à la souveraineté des données : ses clients sont informés de l'emplacement et des modalités d'accès aux données hébergées, ce qui leur garantit une sécurité maximale. En complément, l'entreprise propose un service client accessible 24h/24 et 7j/7 en plusieurs langues, assurant un accompagnement constant de haute qualité.

Grâce aux récentes acquisitions d'Aster et doeLEGAL, DiliTrust renforce sa suite logicielle pour offrir à ses clients un ensemble de solutions intégrées, couvrant aussi bien la gestion des comités opérationnels et stratégiques que des fonctionnalités avancées pour les départements juridiques, telles que la facturation électronique et le Matter Management.

## 2.3 Le service

La solution principale de DiliTrust est une plateforme SaaS qui centralise et sécurise les processus juridiques de ses clients. Cette plateforme offre divers modules, chacun répondant aux besoins spécifiques des services juridiques, notamment la digitalisation des conseils d'administration, la gestion des contrats (CLM), la gestion des litiges, et une solution de gestion documentaire. En rendant ces fonctions accessibles via une interface unique, DiliTrust simplifie la collaboration des équipes juridiques, réduit les tâches administratives et renforce la sécurité des informations sensibles. L'entreprise se différencie en intégrant des outils d'intelligence artificielle et d'automatisation, qui réduisent le temps consacré aux tâches répétitives, optimisant ainsi le travail des équipes et garantissant la confidentialité des données.

DiliTrust, en tant que créateur d'IA depuis 2014, garantit une technologie indépendante et un contrôle total des données. Son IA, spécialement formée pour le secteur juridique, permet d'analyser et de traiter les informations juridiques. Parmi les fonctionnalités proposées, l'IA de DiliTrust peut identifier automatiquement les écarts et risques dans les contrats, mettre en évidence les informations clés, résumer des documents pour les membres du conseil, et structurer les processus de relecture et d'approbation internes. Ces capacités offrent aux équipes juridiques un gain de temps et de précision, notamment en facilitant la comparaison des versions de documents et en assurant le suivi des clauses essentielles.

## 2.4 Le positionnement du stage dans les travaux de l'entreprise

Ce stage s'intègre dans la stratégie de DiliTrust pour optimiser l'innovation au sein de son département R&D. En effet, le projet consiste à développer un agent de documentation automatisé pour faciliter la préparation des dossiers liés au Crédit d'Impôt Recherche (CIR). Cette tâche nécessite une documentation précise des projets de recherche, ce qui représente un processus long et complexe. Mon rôle est de créer, en utilisant l'API de ChatGPT en Python, un outil d'automatisation qui génère un contenu structuré conforme aux exigences du CIR.

Ce projet apportera un gain de temps significatif pour l'équipe R&D, en lui permettant de se concentrer sur des projets à plus forte valeur ajoutée tout en assurant une qualité de documentation optimale. Cette démarche s'aligne parfaitement avec les priorités de DiliTrust en matière d'innovation, car elle renforce la compétitivité de l'entreprise et sa capacité à répondre efficacement aux besoins de ses clients en matière de digitalisation des processus juridiques.

## 3 Travail effectué

### 3.1 Cahier des charges

#### 3.1.1 But Général

Le projet qui m'a été attribué lors de mon stage avait pour objectif de concevoir et d'implémenter un agent intelligent capable de générer/rédiger une documentation technique conforme aux exigences du Crédit d'Impôt Recherche (CIR) grâce à ChatGPT. Ce projet s'inscrit dans le cadre de l'amélioration des processus internes de documentation au sein du département Recherche et Développement (R&D). ChatGPT est un outil très puissant en matière de synthétisation et de rédaction de documents, en exploitant ces capacités de la bonne manière et en utilisant les différentes options qu'il propose, l'objectif est de simplifier et d'accélérer la création de ces documents.

Le crédit d'impôt recherche est une incitation fiscale à la R&D qui exige une documentation rigoureuse, rédigée selon des critères précis. Les entreprises doivent prouver l'originalité, les défis scientifiques ou techniques, ainsi que les avancées réalisées pour chaque projet. La production de ces documents étant chronophage, il s'agit d'une tâche idéale à optimiser.

#### 3.1.2 Explication détaillée des résultats à obtenir

Pour cette section, partons du principe que nous souhaitons rédiger la documentation d'un projet R&D intitulé « Projet S7 ».

Nous allons passer en revue chaque étape clé de l'agent intelligent pour qu'il parvienne à nous rédiger une documentation pour le crédit impôt recherche.

Dans le cadre de la réalisation de notre Projet S7, nous avons créé des notes tout au long de son avancement, incluant les solutions trouvées, les échecs rencontrés, les sources utiles, les dates clés, l'attribution des tâches... Ces informations sont documentées sur différentes plateformes, notamment : Jira, Confluence, des fichiers PDF ou, plus généralement, des pages web.

A partir de ces informations, voici les étapes clé de l'agent intelligent :

#### 1. Récupération des données pertinentes

L'agent intelligent doit extraire automatiquement les informations nécessaires à la documentation du projet en se basant sur les notes prises tout au long de celui-ci, comme mentionné précédemment. Ces outils permettent de centraliser les ressources pour chaque projet de R&D. Pour ce faire, l'utilisateur fournira les URL des différentes pages contenant les ressources. La rédaction du projet sera intégralement fondée sur le contenu issu de ces URL.

## 2. Conception d'un plan structurant la documentation

Afin de répondre aux attentes spécifiques définies par le CIR, l'agent doit élaborer un plan de rédaction conforme aux normes requises. En s'appuyant sur ces standards, l'agent génère une structure détaillée pour le document, précisant les sections attendues et fournissant une estimation du nombre de mots nécessaire pour chaque partie en fonction des sources dont il dispose pour rédiger chaque paragraphe.

## 3. Génération de contenu rédigé

À partir du plan fourni à l'étape précédente et des données collectées, l'agent produit un contenu rédigé sous forme de paragraphes clairs et cohérents, mettant en valeur les informations essentielles et en citant chaque source utilisée pour rédiger chaque paragraphe. Ce processus inclut une analyse approfondie des sources pour éviter les répétitions et les omissions, tout en assurant une présentation fluide et logique. Le challenge est de garder une cohérence entre les différentes sections. A cette étape notre documentation CIR pour notre projet S7 est terminée !

## 4. Facilité d'intégration et d'utilisation

La dernière étape du projet consiste à créer une interface utilisateur simple et intuitive pour faciliter l'utilisation de l'agent. Pour cela, l'agent a été directement intégré à Slack, un outil de messagerie professionnelle largement utilisé pour la communication interne au sein de l'entreprise. Cette intégration permet non seulement de simplifier l'accès à notre nouvel outil, mais aussi d'assurer une adoption rapide par les équipes grâce à un environnement qu'elles connaissent déjà.

## 5. Tests

Étant donné que le résultat attendu de ce projet est une documentation, il n'existe pas de "bonne réponse" définitive. Cependant, il est possible de valider la qualité du contenu en vérifiant les sources utilisées pour chaque paragraphe, ainsi qu'en relisant les sections rédigées pour s'assurer qu'aucune information non issue des ressources fournies n'a été ajoutée.

Lors des premières étapes il faut également prendre en compte le coût des appels à l'API de ChatGPT pour avoir conscience des coûts demandés par la génération de la documentation, que cela soit en nombre d'appel ou en quantité de mots par appels. (*Workflow global du projet Cf. annexe p16*)

## 3.2 Compte-rendu d'activité

### 3.2.1 Axes d'étude et de recherche choisis

Tout d'abord, la gestion des ressources a constitué un axe central du projet. Cette étape impliquait la transformation de fichiers non structurés, tels que des PDF, des pages Confluence, Jira ou encore des pages web, en fichiers JSON à partir de leurs URL. Afin d'exploiter efficacement ces JSON, il a été nécessaire d'ajouter des métadonnées pertinentes, garantissant ainsi une utilisation optimale des ressources.

Ensuite, un travail d'optimisation et de structuration des prompts a été réalisé afin de tirer parti des capacités de ChatGPT. Cela s'est appuyé sur une exploitation rigoureuse des ressources disponibles pour produire une documentation pertinente. Des fonctionnalités avancées du modèle ont été utilisées, comme l'utilisation de prompts système et utilisateur pour établir un contexte clair, la possibilité d'inclure des images dans les prompts, ainsi que d'avoir des réponses structurées au format JSON.

Puis, l'automatisation des différentes étapes a été mise en place grâce à des scripts Python. Ces scripts ont été conçus pour s'intégrer avec les API de ChatGPT, Jira et Confluence. L'objectif principal était d'éliminer toute intervention manuelle de l'utilisateur une fois le script lancé. Par exemple, pour récupérer automatiquement toutes les informations d'une page, un scraper a été implémenté, évitant ainsi les opérations fastidieuses et répétitives de copier-coller.

Enfin, une solution simple et intuitive pour mettre l'outil à disposition de tout le monde.

### 3.2.2 Déroulement concret des études, expérimentations, mises au point...

Au début du projet, un petit script Python existait déjà, permettant de créer un plan et de rédiger quelques éléments. Cependant, ce script était très basique, limité dans ses fonctionnalités et ne répondait pas pleinement aux exigences du projet. Il servait principalement à donner une idée générale de l'approche à adopter. Malgré ses limites, il m'a aidé à mieux comprendre les attentes et les besoins liés au développement de l'agent de documentation.

#### 1. Génération du plan

Premièrement je dois déterminer comment fournir les ressources nécessaires à ChatGPT pour qu'il puisse générer un plan de rédaction. J'ai d'abord décidé de mettre toutes les informations dans un seul prompt, car cette étape visait à obtenir les grandes lignes du plan sans entrer dans les détails. Les sources étaient données dans le prompt avec une certaine structure (*cf. annexe p16*) pour faciliter ChatGPT à les identifier.



Une fois le plan généré, l'étape suivante consiste à citer les sources nécessaires pour la rédaction. J'ai d'abord choisi de faire un appel à ChatGPT pour chaque paragraphe du plan, en fournissant toutes les sources disponibles, afin qu'il identifie celles qui étaient pertinentes. Cependant, cette méthode nécessitait un appel supplémentaire par paragraphe, ce qui alourdissait le processus. Par la suite, j'ai découvert qu'il était possible de demander à ChatGPT de renvoyer une réponse structurée sous forme de JSON, en définissant sa structure à l'aide du module Pydantic. Nous avons donc décidé de combiner la génération du plan et la citation des sources en une seule étape, optimisant ainsi le workflow. (Cf. annexe p17)

## 2. Rédaction des paragraphes

L'étape suivante consiste à rédiger les paragraphes. Grâce à l'étape précédente, nous savons précisément quoi écrire dans chaque paragraphe et quelles sources utiliser pour chaque section. Cependant, un aspect n'avait pas encore été pris en compte : les images. Une grande partie des informations liées aux projets repose sur des schémas, dessins, photos, et autres supports visuels. Il est donc essentiel d'intégrer ces éléments dans la rédaction. Ces images n'avaient pas été prises en compte lors de la génération du plan, car il n'était pas possible d'y associer des métadonnées permettant de les référencer, et le volume d'informations à traiter dans le prompt aurait été trop élevé.

Lors de la rédaction, ChatGPT cite systématiquement les sources utilisées, ce qui permet de vérifier sur quelles informations il s'est appuyé et de s'assurer que le contenu est cohérent avec ces sources, cela structure d'avantage le document.

## 3. Scraper

Jusqu'à présent, je testais mon script en utilisant un dataset issu d'un projet de R&D. Pour cela, j'avais manuellement copié-collé toutes les ressources, une par une, depuis chaque URL, récupéré les images, puis organisé le tout dans des fichiers JSON pour que ces données soient facilement exploitables dans le programme. Cependant, cette méthode est particulièrement fastidieuse et chronophage. Pour remédier à cela, je dois implémenter un scraper qui automatisera ce processus, rendant le travail beaucoup plus rapide, précis et efficace.

Les quatre types de contenus à scraper sont les suivants : PDF, pages Confluence, pages Jira et pages web. Chaque type de source nécessite une méthode spécifique de scraping. L'agent doit donc prendre en entrée toutes les URLs nécessaires à la documentation ainsi que leur type de page.

Pour les pages Confluence et Jira, une API dédiée est disponible, facilitant le scraping grâce à l'utilisation de l'ID présent dans l'URL. Pour les PDF, j'ai utilisé des bibliothèques Python spécialisées dans leur manipulation. Enfin, pour les pages web, il a fallu adopter une approche plus générale en utilisant la bibliothèque BeautifulSoup. Étant donné que les pages web sont construites de manière variée, il était essentiel de trouver une méthode qui regroupe la majorité des informations utiles tout en évitant les éléments inutiles présents sur une page web.

Voici le format des données générées par le scraper pour chaque source, quel que soit le type d'URL :

```
{
  "id": "<unique identifieur>",
  "name": "Title of the Source",
  "URL": "https://example.com/source-page",
  "content": "Text source content.",
  "images": [
    "path/to/image1",
    "path/to/image2",
  ]
}
```

Les images sont toutes enregistrées dans un dossier temporaire avec des chemins uniques, ce qui permet de les identifier facilement.

#### 4. Interface Utilisateur

Le flux global du projet est désormais implémenté. Il reste à créer un bot Slack pour rendre l'utilisation de l'agent simple et intuitive. Slack propose un "Builder" permettant de concevoir des interfaces de manière relativement facile, ce que j'ai utilisé pour développer l'interface du bot. Grâce à cela, il n'est pas nécessaire de gérer le développement d'un front-end personnalisé.

Dans cette interface, j'utilise des boutons d'action et des champs d'entrée pour interagir avec l'utilisateur. Cela nécessite l'implémentation de l'API afin de gérer les différentes fonctionnalités que je souhaite offrir, notamment :

- **Ajouter une nouvelle source** : un bouton permettant de fournir une nouvelle URL.
- **Supprimer une source** : un bouton pour retirer une URL précédemment ajoutée.
- **Choisir le type d'URL** : un bouton ou un menu déroulant pour sélectionner le type de ressource (PDF, page web, Confluence, Jira, etc.).
- **Valider et lancer la génération de la documentation** : un bouton pour démarrer le processus une fois toutes les informations nécessaires fournies.

*(Cf. annexe p18)*

Pour générer la documentation, l'utilisateur devra fournir plusieurs informations essentielles : le nom du projet, toutes les URLs des sources pertinentes, ainsi que le type de chaque URL (par exemple, PDF, page web, Confluence, ou Jira). Il devra également indiquer l'URL d'une page Confluence où la documentation sera accessible. L'agent créera alors une page enfant sous celle spécifiée, y copiera l'ensemble de la documentation générée, et retournera l'URL de cette nouvelle page afin que l'utilisateur puisse y accéder facilement.

*(Cf. annexe p19)*

### 3.3 Interprétation et critique des résultats

#### 3.3.1 Résultats obtenus

Le projet a permis de générer une documentation conforme aux exigences du Crédit d'Impôt Recherche (CIR), avec une qualité de rédaction satisfaisante. Les sections produites sont bien structurées, argumentées et correctement sourcées, offrant une base solide pour présenter les projets de R&D de manière professionnelle.

L'agent offre un gain de temps considérable : ce qui nécessitait autrefois environ six heures de travail manuel peut désormais être réalisé en seulement 15 minutes grâce à cet outil de documentation.

L'agent fournit l'URL de la page contenant toute la documentation rédigée, conformément aux attentes. Le contenu est structuré en respectant le format Markdown généré par ChatGPT, ce qui garantit une présentation claire et professionnelle. De plus, une table des matières est automatiquement ajoutée en haut de la documentation, rendant la relecture et la navigation plus agréables et efficaces pour l'utilisateur.

#### 3.3.2 Limites et axes d'amélioration

##### 1. Limites

Bien que l'outil ne remplace pas entièrement l'intervention humaine, une relecture et des ajustements restent obligatoires pour garantir une documentation correcte. La documentation générée, qui fait en moyenne une douzaine de pages, contient souvent des erreurs ou des imprécisions dans les paragraphes. De plus, certaines formulations produites par ChatGPT ont tendance à être excessives, ce qui n'est pas adapté au style attendu dans ce type de documentation. Ces relectures permettent de corriger ces points, de simplifier les phrases lorsque nécessaire, et de s'assurer que le ton employé est en accord avec les exigences spécifiques du Crédit d'Impôt Recherche (CIR).

Il arrive parfois que certaines sections du schéma de rédaction manquent de ressources nécessaires pour produire un contenu solide. En l'absence de ces ressources, ChatGPT génère souvent un paragraphe générique qui n'apporte pas de réelle valeur. Pour résoudre ce problème, deux approches sont envisagées : soit ne pas rédiger le paragraphe concerné et passer directement à la section suivante, soit ajouter des sources complémentaires, si elles sont disponibles, afin d'enrichir la rédaction. Cette approche a fonctionné, mais en l'absence de ressources pour certaines sections, ChatGPT revenait systématiquement au prompt initial décrivant le plan, nécessitant l'ajout de plus de ressources.

L'agent étant complément basé sur ChatGPT :

- Les formulations générées peuvent parfois refléter des biais implicites présents dans les données d'entraînement, ce qui peut affecter la neutralité ou l'objectivité attendue dans une documentation technique.
- Le transfert de données vers ChatGPT pour la génération de contenu peut poser des problèmes de confidentialité et de sécurité

## 2. Améliorations

L'agent a été étendu pour aller au-delà de la simple rédaction de documentation conforme au Crédit d'Impôt Recherche (CIR). En effet, la documentation est importante en interne, notamment pour les équipes qui n'ont pas travaillé directement sur un projet donné. Ces membres peuvent ainsi consulter une documentation détaillée pour mieux comprendre le projet et identifier des éléments d'intérêt.

Une nouvelle fonctionnalité a donc été ajoutée à l'agent, lui permettant de rédiger une documentation interne. Celle-ci est créée à partir d'une structure prédéfinie spécifiquement pour les projets internes, offrant une ressource claire et accessible aux membres de l'équipe qui souhaitent approfondir leurs connaissances sur ces projets, que ce soit pour des projets annexe ou bien par curiosité.

Enfin, une dernière fonctionnalité a été implémentée pour étendre les possibilités de documentation : la personnalisation du plan de rédaction. Cette option permet à l'utilisateur de définir lui-même la structure souhaitée pour la documentation.

Concrètement, l'utilisateur peut activer cette option en cochant une case, ce qui fait apparaître une zone de texte dans laquelle il peut entrer son plan personnalisé. Par exemple :

- Section 1 : Introduction
- Section 2 : Chronologie
- Section 3 : ...

Cette flexibilité permet de répondre à des besoins spécifiques, en adaptant la documentation aux exigences précises de chaque projet ou utilisateur (*Cf. annexe p19*). La documentation générée pour le Crédit d'Impôt Recherche est rédigée en français, tandis que les documentations internes et personnalisées sont produites en anglais.

Le scraper dispose également d'un potentiel d'amélioration, notamment en élargissant son champ d'action pour récupérer des informations depuis d'autres types de sources. Par exemple, il pourrait être adapté pour extraire des données à partir de fichiers Excel ou interagir directement avec des bases de données. Ces extensions permettraient d'enrichir encore davantage les ressources disponibles pour la génération de documentation.

## 4 Conclusion générale

Ce stage a été une opportunité pour développer un outil innovant répondant aux besoins de production de documentation technique conforme aux exigences du Crédit d'Impôt Recherche (CIR). À travers l'automatisation des processus de collecte, de structuration et de rédaction de données, l'agent intelligent conçu permet un gain de temps significatif, réduisant les tâches manuelles de plusieurs heures à seulement quelques minutes.

Le projet a mis en lumière des défis techniques variés, notamment la gestion des données non structurées, l'optimisation des prompts pour exploiter efficacement les capacités de ChatGPT, et l'intégration fluide de l'outil au sein des environnements professionnels comme Slack. L'approche adoptée s'est également attachée à garantir la conformité et la qualité des documents produits tout en explorant des fonctionnalités supplémentaires telles que la personnalisation des plans et la rédaction de documentations internes en anglais.

D'un point de vue personnel, ce stage m'a permis de renforcer mes compétences en développement Python, en manipulation d'API et en traitement de données. J'ai également développé une meilleure compréhension des contraintes liées aux outils d'intelligence artificielle, telles que la confidentialité des données et la nécessité d'une supervision humaine pour assurer la qualité des résultats.

Mais aussi de développer des compétences pour aborder la structuration d'un projet complexe. J'ai appris à séparer les concepts dans différentes classes, une approche qui favorise la modularité et la maintenabilité du code. En adoptant une architecture bien pensée, chaque composant de l'agent intelligent a été conçu de manière indépendante mais interconnectée, rendant le projet plus lisible et plus facile à faire évoluer.

À l'entreprise, ce projet apporte une solution performante et évolutive qui simplifie la création de documents stratégiques, optimise les ressources et améliore les processus internes. Il pose également les bases d'un outil modulable pouvant être étendu à d'autres types de documentations ou intégré à de nouveaux flux de travail.

En somme, ce projet illustre parfaitement la manière dont les technologies d'intelligence artificielle peuvent transformer les pratiques professionnelles en offrant à la fois gain de temps, efficacité et adaptabilité aux besoins de l'entreprise.

## 5 Bibliographie & Glossaire

### Bibliographie

#### 1. OpenAI Documentation

- Documentation officielle sur les modèles ChatGPT et leur utilisation via l'API.
- <https://platform.openai.com/docs>

#### 2. Documentation sur le Crédit d'Impôt Recherche (CIR)

- Guide pratique pour l'élaboration de dossiers CIR.
- Ministère de l'Enseignement Supérieur et de la Recherche.
- [https://www.impots.gouv.fr/sites/default/files/media/1\\_metier/2\\_professionnel/EV/4\\_difficultes/410\\_rescrit/modele\\_rescrit\\_credit\\_impot\\_recherche.pdf](https://www.impots.gouv.fr/sites/default/files/media/1_metier/2_professionnel/EV/4_difficultes/410_rescrit/modele_rescrit_credit_impot_recherche.pdf)

#### 3. Bibliothèques Python

- **BeautifulSoup** : Bibliothèque utilisée pour le web scraping.
- <https://www.crummy.com/software/BeautifulSoup>
- **Pydantic** : Gestion des modèles de données en Python.
- <https://pydantic-docs.helpmanual.io>

#### 4. Documentation Slack API

- Documentation officielle sur l'intégration d'applications dans Slack.
- <https://api.slack.com>

#### 5. Jira & Confluence APIs

- Guide pour l'interfaçage avec les plateformes Atlassian.
- <https://developer.atlassian.com>

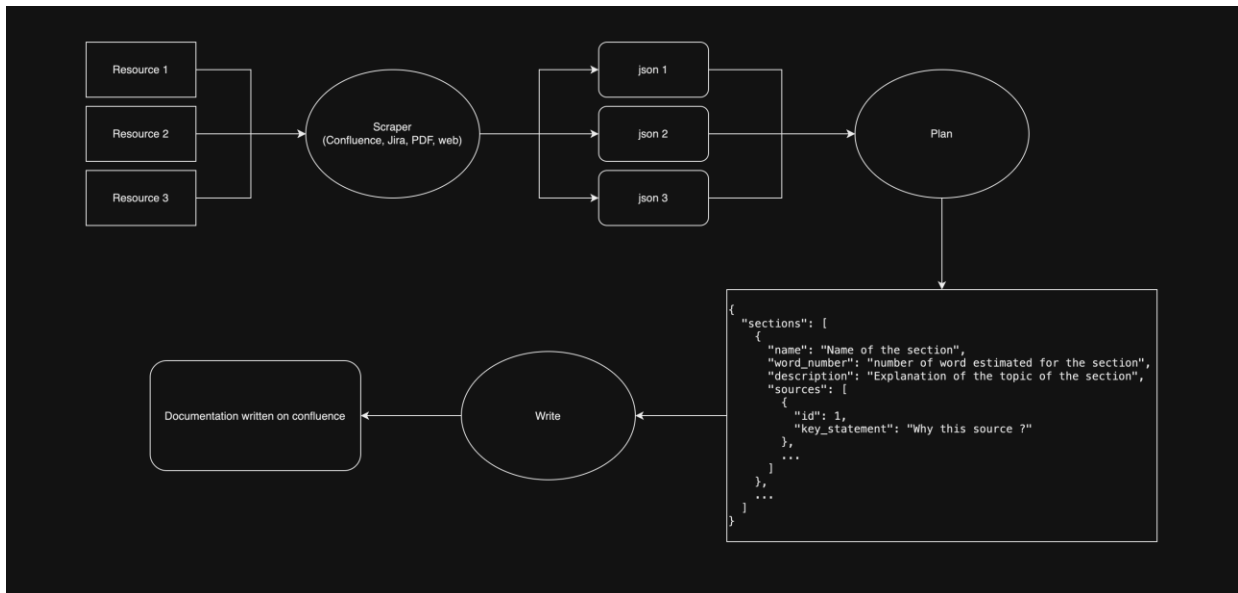
### Glossaire

- **Prompt** : Commande ou question donnée à un modèle d'intelligence artificielle pour qu'il génère une réponse.
- **JSON (JavaScript Object Notation)** : Format de fichier utilisé pour structurer et transmettre des données dans un format facilement lisible par les machines.
- **Scraping** : Technique utilisée pour extraire des informations depuis des pages web ou des fichiers.
- **Slack Bot** : Programme intégré à la plateforme Slack permettant d'automatiser des tâches ou d'offrir des services interactifs.
- **Métadonnées** : Informations supplémentaires sur des données, comme leur contexte, leur origine ou leur format.
- **BeautifulSoup** : Bibliothèque Python utilisée pour analyser et extraire des données de pages HTML et XML.
- **Mark down** : Syntaxe légère permettant de formater du texte de manière structurée, souvent utilisée dans la documentation.
- **Pydantic** : Outil Python permettant de valider les structures de données et de générer des modèles facilement exploitables.



## 6 Annexes

### Plan global du projet



### Structuration des sources dans le prompt

L'utilisateur donnera sa documentation technique sous cette forme :

---

---

Identifiant : {id de la source 1}

Name : {nom de la source 1}

Contenu : {contenu de la source 1}

---

---

---

---

Identifiant : {id de la source 2}

Name : {nom de la source 2}

Contenu : {contenu de la source 2}

---

---

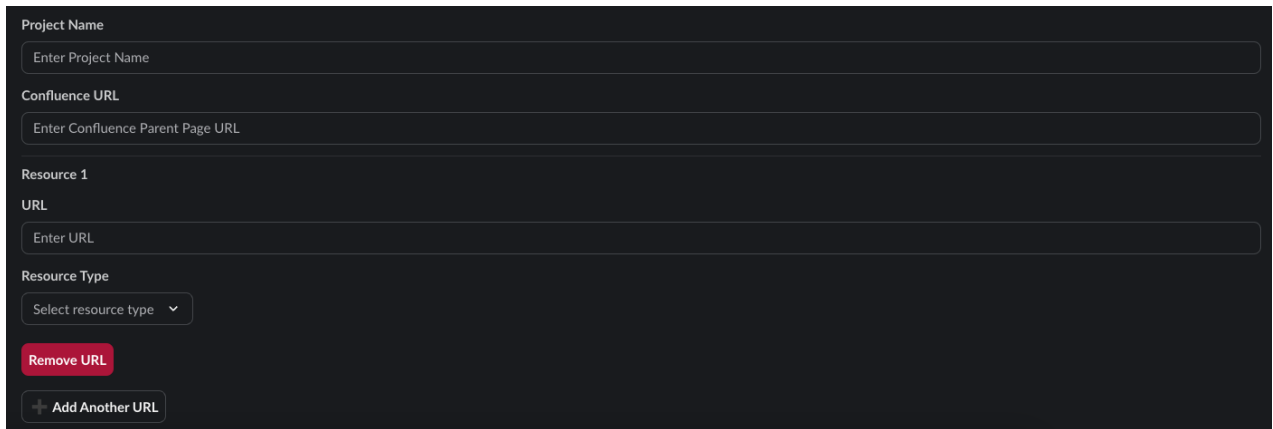
Sous cette forme, ChatGPT peut facilement s'orienter parmi les différentes sources, ce qui facilite la construction de son plan.



### Exemple de plan JSON rendu par ChatGPT

```
{
  "sections": [
    {
      "name": "Introduction",
      "word_number": 150,
      "description": "Présentation du sujet et contexte général.",
      "sources": [
        {
          "id": 1,
          "key_statement": "Le changement climatique est une des plus grandes menaces actuelles."
        },
        {
          "id": 2,
          "key_statement": "La prise de conscience mondiale a augmenté ces dernières décennies."
        }
      ]
    },
    {
      "name": "État des lieux",
      "word_number": 300,
      "description": "Analyse de la situation actuelle en matière de changement climatique.",
      "sources": [
        {
          "id": 3,
          "key_statement": "Les températures globales ont augmenté de 1,2 °C"
        },
        {
          "id": 4,
          "key_statement": "Les émissions de CO2 continuent de croître malgré les engagements."
        }
      ]
    },
    {
      "name": "Solutions proposées",
      "word_number": 250,
      "description": "Présentation des solutions possibles pour limiter le réchauffement climatique.",
      "sources": [
        {
          "id": 5,
          "key_statement": "La transition énergétique vers des sources renouvelables est essentielle."
        },
        {
          "id": 6,
          "key_statement": "Les politiques de reforestation peuvent absorber une partie des émissions."
        }
      ]
    },
    {
      "name": "Conclusion",
      "word_number": 100,
      "description": "Synthèse et message final pour sensibiliser à l'action collective.",
      "sources": [
        {
          "id": 7,
          "key_statement": "Un effort global est nécessaire pour éviter des conséquences catastrophiques."
        }
      ]
    }
  ]
}
```

## Interface utilisateur Slack bot



The screenshot shows a dark-themed Slack interface. At the top, there's a 'Project Name' section with a text input field labeled 'Enter Project Name'. Below that is a 'Confluence URL' section with a text input field labeled 'Enter Confluence Parent Page URL'. The 'Resource 1' section contains a 'URL' text input field labeled 'Enter URL', a 'Resource Type' dropdown menu with the text 'Select resource type', a red 'Remove URL' button, and a green '+ Add Another URL' button.

### Project Name

- **Champ de texte** : saisir le nom du projet pour lequel la documentation doit être générée.

### Confluence URL

- **Champ de texte** : L'utilisateur saisit l'URL de la page parente Confluence sur laquelle la documentation générée sera sauvegardée. Une nouvelle page enfant sera créée automatiquement sous cette page, contenant toute la documentation.

### Resource 1

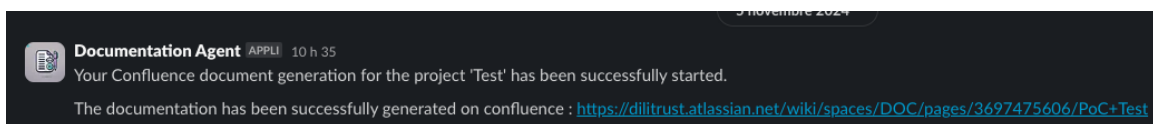
- **Champ de texte (URL)** : L'utilisateur renseigne l'URL d'une ressource pertinente à exploiter pour la documentation
- **Menu déroulant (Resource Type)** : L'utilisateur sélectionne le type de ressource à partir d'une liste ("Page web", "PDF", "Page Confluence" ou "Page Jira"). Cela aide l'agent à choisir le bon mode de scraping.

### Add Another URL

- Permet d'ajouter une nouvelle ressource. Une nouvelle section pour saisir une URL et choisir son type est générée dynamiquement en cliquant sur ce bouton.


### Remove URL

- Permet de supprimer une ressource déjà ajoutée. Si l'utilisateur s'est trompé ou souhaite retirer une URL, ce bouton supprime la ligne correspondante.



On reçoit ce message de la part du bot lorsque la documentation est prête.

## Améliorations du Slack Bot

 **Document Agent CIR**

The Document Agent is a bot designed to assist in creating comprehensive documentation for your project. It offers various types of documentation templates tailored to meet specific needs.

**Features**

Document Agent can create 3 types of documentation:

- \* **CIR Documentation:** Documentation formatted to meet the standards required by the French CIR (Crédit Impôt Recherche).
- \* **Internal Documentation:** Documentation created based on a predefined structure for in-house projects. Ideal for team members who want to learn more about these projects.
- \* **Custom Documentation:** Documentation tailored to a specific structure or outline that you provide, adapting to unique project requirements.

Please chose the type of documentation you need

- ☐ CIR documentation
- ☐ Internal documentation
- ☐ Customized documentation

Here's how to use the documentation agent:

1. Specify the Confluence Page Parent URL where you'd like the documentation to appear. It will be created as a child of the given parent page.
2. Enter all resources necessary for the documentation.
3. For every resource, provide its url and select its type (Confluence, Jira, PDF, or Web).

⚠ Data will be sent to OpenAI, so be careful about the resources you chose ⚠

**Project Name**

**Confluence URL**

**Resource 1**

**URL**

**Resource Type**

Select resource type ▾

Please chose the type of documentation you need

- ☐ CIR documentation
- ☐ Internal documentation
- ☒ Customized documentation

Enter the plan of your desired documentation

section 1: intro ...

section 2: first part ...

...

L'interface finale de l'agent de documentation présente un bouton permettant de sélectionner le type de documentation. Lorsqu'on clique sur « Customized documentation », une zone de texte apparaît pour saisir le plan souhaité