

**Master 2 - Language et Programmation**

**RAPPORT DE STAGE**

**Maintenance et Développement de  
l'application web ESENDEO  
- la solution logistique pour les commerçants**

**Ke LYU**

Du 1 Avril au 30 Septembre 2020

Tuteurs d'entreprise :

Tao YAN

Weiquan LONG

Responsable de stage :

Ralf Treinen

# REMERCIEMENTS

Je tiens tout d'abord à remercier tout le personnel de FTL pour son accueil chaleureux, son soutien tout au long de mon stage et les diverses connaissances qu'ils ont partagé avec moi durant toute cette période.

Je tiens remercier Monsieur Weiquan LONG pour m'avoir permis de réaliser ce stage.

Je tiens ensuite à remercier tout particulièrement Monsieur Tao YAN, qui a su m'apporter son expérience pour me guider et me conseiller tout au long du projet tout en me laissant assez indépendant sur la manière de conduire celui-ci et qui m'a montré clairement la différence entre des cas d'écoles et des cas pratiques de l'entreprise, ainsi que son aide dans l'élaboration de ce rapport.

Enfin, je tiens également remercier le département Informatique de Université de Paris pour la formation offerte, et mon suiveur de stage Monsieur Ralf Treinen qui m'a donné des conseils et était toujours disponible en cas de besoin.

# TABLE DES MATIERES

REMERCIEMENTS.....	1
TABLE DES MATIERES .....	2
INTRODUCTION .....	3
1.ENVIRONNEMENT.....	4
1.1 L'ENTREPRISE .....	4
1.2 L'EQUIPE.....	4
2. APPLICATION .....	5
3. TECHNOLOGIE .....	10
3.1 RESTFUL .....	10
3.2 SEPARATION DE FRONT-END ET BACK-END.....	11
3.3 SPA .....	11
3.4 VUE.JS.....	12
3.5 FEATHERSJS VS NESTJS.....	13
4. MISE EN ŒUVRE.....	15
4.1 METHODOLOGIE AGILE.....	15
4.2 DEVOPS.....	17
5. MON TRAVAIL.....	19
5.1 MON ROLE.....	19
5.2 MES TRAVAIS REALISE'S.....	19
5.2.1 Nouveau Module - Douane Configuration .....	20
5.2.2 Refactoring Module - SAV.....	23
5.2.3 Migration.....	24
5.2.4 Migration - client/ user/ role.....	25
5.2.4 Migration et test -billing/invoice.....	28
6. DIFFICULTÉ'S RENCONTRE'S.....	31
7. CONCLUSION .....	32
BIBLIOTHÈQUE.....	33

# INTRODUCTION

Ce présent rapport constitue le rapport de mon stage de six mois au sein de l'unité d'informatique de l'entreprise FTL-EXPRESS. Ce stage était dans le cadre de validation de la deuxième année de Master professionnelle spécialité Language et Programmation à l'Université de Paris Diderot.

Créé en 2014, FTL-EXPRESS est une société en France dans la logistique transfrontalière sino-européenne. Il possède sa propre équipe de développement pour soutenir un concept de gestion avancé (portail numérique). Esendeo est une filiale de FTL-EXPRESS, il s'engage à fournir des solutions logistiques multi-canaux à bas prix de pour les entreprises locales en France. Sur leur site Web, les commerçants ou les entreprises peuvent facilement obtenir et comparer les informations sur les services et les coûts de plusieurs transporteurs tels que UPS, Colissimo, y compris les propres services d'Esendeo. Le site Esendeo intègre les informations de service de plusieurs transporteurs et l'organise et enfin les fournit aux utilisateurs clairement pour leur choix.

Ma mission au sein de l'équipe projet Esendeo est de maintenir et de développer de nouvelles fonctionnalités pour le site, qui est actuellement en version 0.3.3, doit encore ajouter de nombreuses nouvelles fonctionnalités. En même temps, je dois également écrire des tests pour garantir la robustesse du code que j'écris. De plus, dans la deuxième phase du stage, j'ai également entrepris une partie du travail de migration du framework de back-end de FeathersJS à NestJS.

Je commencerai ce rapport par une introduction de l'entreprise et de mon équipe. Ensuite, je présenterai le projet principal de mon stage, à savoir un web application b2b, y compris le contexte technique et le cadre de gestion. Après cela, j'expliquerai la mission principale que j'ai réalisée au sein de la société FTL-EXPRESS pendant le stage à travers quelques exemples. Et les difficultés que j'ai rencontrées.

Enfin je conclurai en dressant un bilan du stage (les nouvelles compétences, connaissances, méthodes j'ai acquises) et en donnant des perspectives pour la suite du développement.

# 1. Environnement

## 1.1 L'entreprise

Crée en 2014, FTL EXPRESS est une entreprise de logistique dont l'activité principale est la logistique transfrontalière entre la Chine et l'Europe. Dispose de 4 points de vente en Europe, 3 centres logistiques dans la région parisienne, FTL Express est aujourd'hui une entreprise reconnue dans son domaine et travaille avec les grands groupes comme la Poste, UPS, Amazon etc. Elle se concentre également sur les services logistiques en B2B.

En 2020, FTL Express France compte environ 98 employés, répartis en 3 départements.

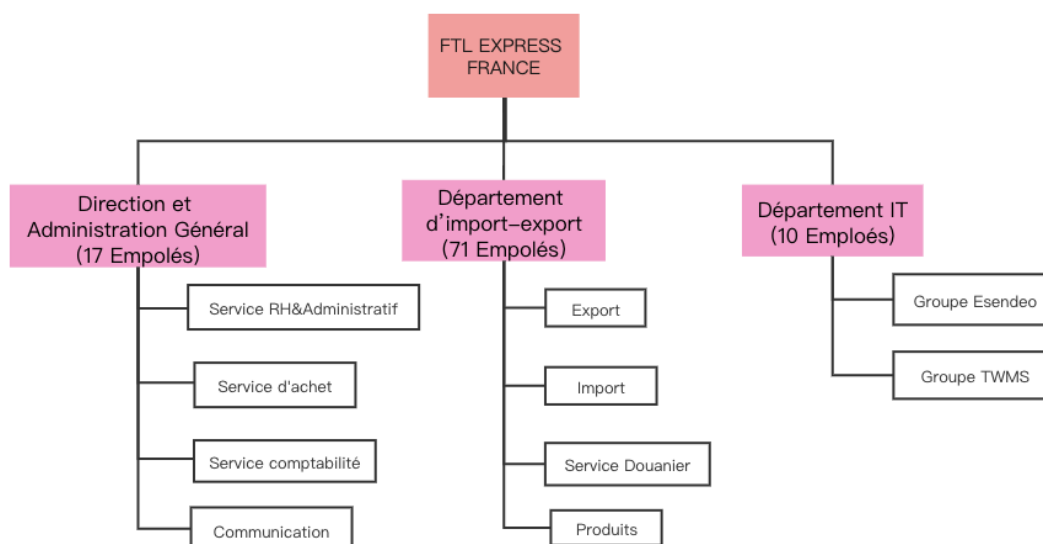


Figure 1– Organisation

## 1.2 L'équipe

Le département informatique de FTL Paris compte 10 personnes, divisé en deux groupes de projets. Groupe TWMS développe le système de gestion des importations de colis, et l'autre groupe Esendeo développe le système de commande et de gestion des colis pour les entreprises françaises, c'est le groupe auquel j'appartiens.

Il y a 4 personnes dans mon groupe. Le développeur senior Tao YAN qui dirige et contrôle l'avancement de l'ensemble du projet. C'est aussi mon tuteur. Le chef de produit Yaoyi LUO est responsable de la collecte des exigences et de la réalisation des tests commerciaux. Deux stagiaires full stack Sijie DONG et moi (Ke LYU) participer au développement de la nouvelle fonction quotidiennes et à la maintenance du site.

## 2. Application

[Esendeo](#) est une plateforme de services logistiques transfrontaliers pour petits et grands colis de 0 à 30 kg, il se concentre sur la conception de solutions logistiques et s'engage à fournir des solutions logistiques de haute qualité, pratiques et abordables pour les commerçants et les e-commerçants.

Similaire à la plate-forme de site Web tiers pour la réservation de billets d'avion, les clients peuvent remplir des informations telles que le poids et le volume du colis sur notre site Web, et obtenir le service logistique et les informations sur les prix de plusieurs transporteurs en un seul clic, ensuite, le client peut choisir l'un des services et passer une commande, puis imprimer la facture de transport et livrer le colis. Ils peuvent également consulter et gérer à tout moment les colis et les détails de la facture de service sur la plateforme. Après avoir profité du service, ils peuvent laisser un message sur Esendeo s'ils rencontrent des problèmes.

The screenshot displays the Esendeo application interface. At the top, there is a navigation bar with four tabs: 'Coordonnées' (checked), 'Colis', 'Service', and 'Notice Importante'. Below this, the main content area is divided into two sections. The left section, titled 'Service', contains a table listing various shipping services from different carriers. The right section, titled 'Vos Colis', provides details for the selected service, including the date and time of pickup, the total price (TTC), and a 'Confirmer' button.

Transporteur	Service	Délai	Tarif	Actions
GLS	GLS Multi-colis Pickup par Esendeo	24 - 48H	7.26€ 6.05€ HT	Choisir
GLS	GLS Pickup par Esendeo	24 - 48H	15.24€ 12.7€ HT	Choisir
colissimo	Colissimo sans signature Déposer à la Poste	2 Jours	16.14€ 13.45€ HT	Choisir
UPS	UPS Multi-colis Pickup par Transporteur	24 - 48 H	20.88€ 17.4€ HT	Choisir
dpd	DPD Pickup par Esendeo	24 - 48H	22.32€ 18.6€ HT	Choisir
chronopost	Chrono 13 Pickup par Esendeo	Avant 13H	24.84€ 20.7€ HT	Choisir

**Vos Colis**

**Coordonnées**

**Colis**

2 Colis, 6 Kg

**Offre**

GLS 2020-09-03 10:00~16:00

**15.24€ TTC**  
12.7€ HT + 2.54€ TVA

Confirmer

Après confirmation de votre affranchissement, nous considérons que vous avez pris connaissance des règles de

Figure 2 – Exemple des services de transporteurs

La plateforme Esendeo propose quatre modules principales: la gestion des commandes, la gestion des informations, la gestion des factures et la gestion des configurations. Parmi eux, la gestion de la configuration n'est visible que par l'administrateur.

L'administrateur peut consulter toutes les informations utilisateur sur la plate-forme, et peut également configurer des listes de prix et diverses règles telles que la configuration de la déclaration en douane et la configuration de l'enlèvement, ainsi que fournir un service après-vente.

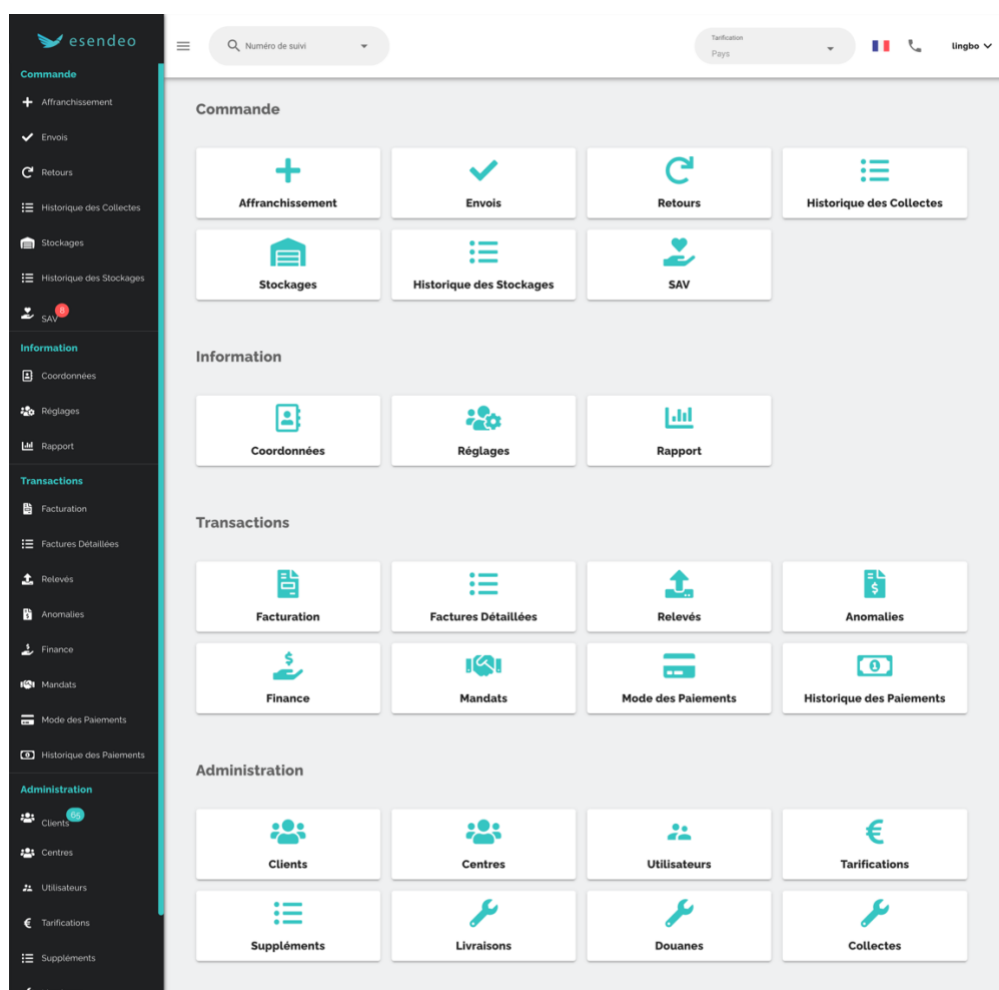


Figure 3 – Page d'accueil pour les admins

Dans le module de gestion des commandes, les clients peuvent prendre les commandes, qui est la fonction principale du site.

Les clients doivent renseigner les adresses de l'expéditeur et du destinataire comme indiqué. Ils peuvent définir les adresses à l'avance dans les paramètres personnels ou importer les adresses par lots avec des fichiers .csv et .xls, afin de facilement sélectionner lors de la commande. Ils doivent également indiquer le poids du colis et doivent parfois renseigner le contenu, le volume et le prix du colis pour faciliter le dédouanement lors du transport transfrontalier.

Une fois que le client a rempli les informations requises, le site Web résumera les informations de service et de prix de plusieurs transporteurs (tels que Colissimo, Dpd, etc...) et les renverra au client par ordre croissant de prix.

Après avoir sélectionné l'un des services et terminé la commande. Ils peuvent télécharger l'étiquette du package sur cette page, ou consulter l'historique des commandes et télécharger le package sur la page des détails du package.

Ensuite, le client peut coller directement l'étiquette sur le colis et l'envoyer au point de réception du transporteur sélectionné pour terminer l'envoi. Pendant le processus d'envoi, l'utilisateur peut voir les informations de suivi du colis sur le site Web.

Ou ils peuvent choisir d'inclure un service de ramassage à domicile, et le transporteur

récupérera le colis directement au domicile du client et l'enverra au point de réception. Ce qui concerne un autre module important: l'événement de collecte et sa configuration.

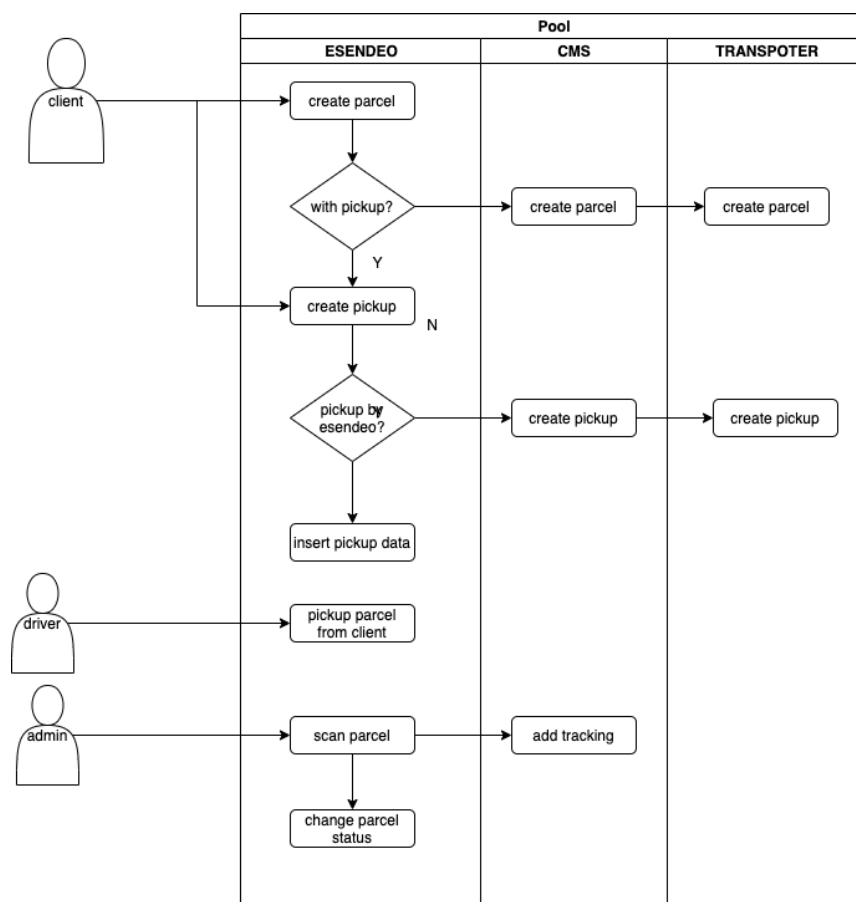


Figure 4 - Processus de ramassage(pickup)

Lorsque les clients créent une commande, nous trouverons tous les services actuellement disponibles pour eux, y compris ceux qui nécessitent une livraison au point de réception en libre-service et ceux qui peuvent être récupérés par transporteurs.

Nous devons d'abord déterminer si le fournisseur de services dispose d'un service de ramassage dans la zone en fonction de l'adresse de l'utilisateur et de la configuration de la zone de ramassage. Si tel est le cas et que l'utilisateur choisit l'option qui inclut le service de ramassage à domicile, nous laisserons le client prendre rendez-vous pour un ramassage en même temps lors de la commande. Ou ils peuvent choisir d'ajouter un rendez-vous à l'événement de ramassage à partir de la commande historique après que la commande est passée.

Nous devons limiter l'heure de rendez-vous disponible, car les jours fériés et les jours de repos ne sont pas sélectionnables et les clients ne peuvent prendre rendez-vous que le lendemain après la dernière heure de prise en charge de chaque jour.

Si le client choisit le service de ramassage fourni par Esendeo, nous devons vérifier si le client est un client qui a besoin d'un ramassage quotidien, car pour les clients avec de gros flux de fret, nous allons récupérer les colis tous les jours. Dans ce cas, il n'est pas nécessaire de prendre rendez-vous pour le ramassage.



Toutes les règles des événements de ramassage, à savoir la zone, le chauffeur, le jour de repos, la configuration quotidienne de ramassage, peuvent être modifiées ou ajoutées de manière flexible par l'administrateur.

Si le colis n'a pas été expédié, le client peut annuler la commande. Ou après avoir reçu la marchandise, les gens peuvent retourner le colis. À tout moment, les utilisateurs peuvent demander le service après-vente pour une certaine commande, simplement en envoyant un message dans le site et en téléchargeant les documents pertinents s'il y en a besoin.

Il existe également une page de statistiques de données, qui utilise divers graphiques pour résumer et analyser les conditions quotidiennes, hebdomadaires, mensuelles et annuelles des colis.

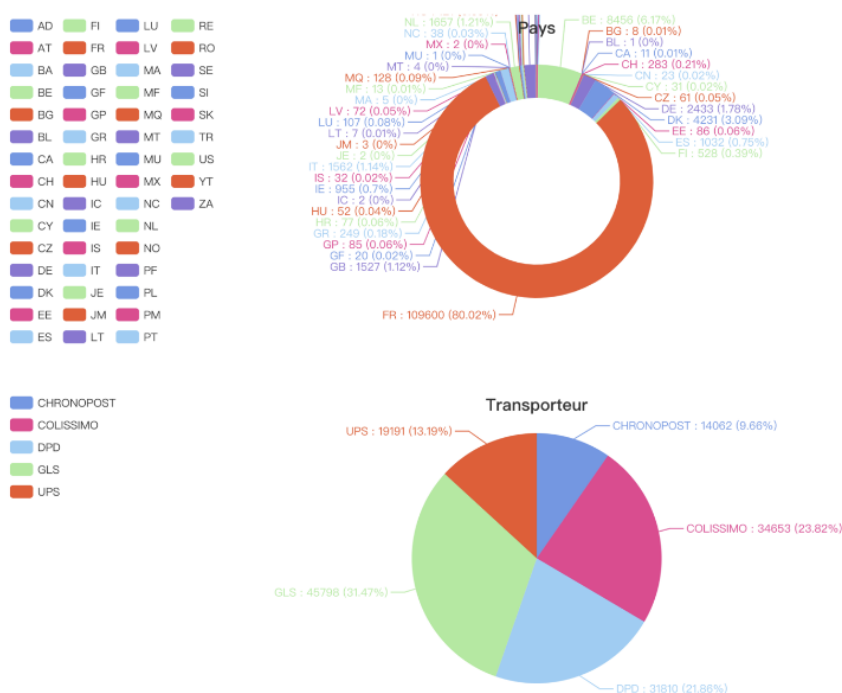


Figure 5 - Page de statistiques de données

Cela permettant à l'administrateur de voir visuellement:

1. Le résumé des frais de transport et le nombre de colis.
2. La répartition du pays de réception du colis.
3. La proportion de la transporteurs de livraison.

Un autre gros module est la gestion des factures. Cette partie est principalement liée au téléchargement, à l'analyse et à d'autres opérations de différents formats de fichiers. Les fichiers téléchargés sont tous stockés sur le serveur cloud Amazon S3.

Les clients doivent télécharger un mandat lors de l'inscription, ce qui nous permet de faire un prélèvement à partir du compte du client.

Le site internet analysera la facture générée par le transporteur, puis générera une facture pour chaque client. Les clients peuvent télécharger la facture mensuelle au format pdf ou xls sur

le site.

L'administrateur peut interroger des informations de facturation plus détaillées, telles que les dépenses mensuelles et les bénéfices mensuels, et peut également télécharger les factures de divers transporteurs dans des fichiers au format .xlsx, .xls et .csv pour faciliter le rapprochement individuel avec chaque transporteur.

## 3. Technologie

Dans le projet Esendeo, nous utilisons le framework Vue.js pour créer l'interface utilisateur sur le front-end, et nous utilisons vuetify comme le framework UI. Sur le back-end, nous utilisons JavaScript et TypeScript comme langage de développement, et nous utilisons FeathersJS et NestJS comme framework NodeJS pour la construction des application RESTful. Nous utilisons principalement PostgreSql comme base de données et nous stockons les fichiers dans le cloud amazon s3.

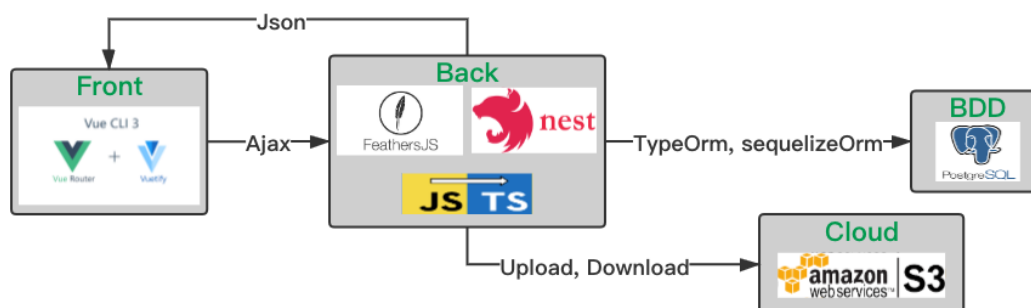


Figure 6 - Structure Technologie

### 3.1 Restful

Rest est un style de conception. L'url spécifie uniquement les ressources. Et les verbes de méthode HTTP sont utilisés pour différentes opérations

Exemple :

// API Non RESTful (Besoins de 3 API)

api/getfile.php - Obtenir des informations sur le fichier, télécharger le fichier.

api/uploadfile.php - Télécharger et créer des fichiers.

api/deletefile.php - Supprimer les fichiers.

// API RESTful (Seulement besoin un API: api/users )

GET api/file

POST api/file

DELETE api/file

L'url de Restful est fortement lisible et auto-descriptive. Elle peut standardiser le processus de demande et renvoyer les résultats, rendre la description de la ressource et la vue faiblement couplée, fournir OpenAPI pour faciliter l'intégration de systèmes tiers, fournir des interfaces de service sans état , réduire la complexité.

## 3.2 Séparation de Front-end et Back-end

L'application Esendeo adopte une méthode de développement qui sépare le front-end et le back-end. Plus précisément, le front-end et le back-end sont chacun un projet indépendant et doivent être déployés séparément. L'idée principale est que la page html front-end appelle l'interface api back-end restful via ajax et utilise des données json pour l'interaction. Le front-end traite les couches View et Controller, tandis que le back-end traite la couche Model et le traitement du code logique.

La séparation front-end et back-end du flux de travail peut faire que le front-end se concentre uniquement sur le front-end, le back-end ne se soucie que des activités back-end, les deux développements peuvent être effectués en même temps. Lorsque le back-end ne fournit pas d'interface, le front-end peut d'abord écrire les données ou appeler le fichier json local. Il n'est plus nécessaire de modifier le back-end pour ajouter des pages et modifier les itinéraires, ce qui rend le développement plus flexible..

En configurant le routage front-end, nous pouvons charger la page à la demande sans charger toutes les ressources du site Web lorsque la page d'accueil est chargée. Et le serveur n'a plus besoin d'analyser la page frontale, ce qui peut améliorer l'interaction de la page et l'expérience utilisateur.

## 3.3 SPA

Le site Esendeo est un SPA(Single Page Application).

SPA est une application avec une seule page principale, et les ressources associées telles que js, css et ainsi de suite ne doivent être chargées qu'une seule fois au début. Tout le contenu est contenu dans la page principale et chaque module fonctionnel est en composants.

Le routage et le saut de page d'une SPA consistent à changer les composants associés et à n'actualiser que des ressources partielles.

En revanche, MPA(Multi Page Application) fait référence à une application avec plusieurs pages indépendantes, et chaque page doit charger à plusieurs reprises des ressources js et css. Le routage des applications multi-pages nécessite l'actualisation de la page entière.

L'avantage du SPA est que le changement de page est rapide et qu'il réduit dans une certaine mesure la pression sur le serveur principal ( Le serveur ne gère pas la logique et le rendu de la page). Le programme back-end n'a besoin que de fournir des API, que le client soit une interface Web ou un téléphone mobile.

Cependant, la vitesse d'ouverture du premier écran de SPA est lente, car l'utilisateur doit télécharger le code du framework SPA et de l'application pour la première fois, puis rendre la page. De plus, SPA n'est pas fort en SEO(Search Engine Optimization) car certains web

spider ne peuvent pas exécuter le code JS ou obtenir des données via AJAX.

Les frameworks SPA actuellement populaires sont: Angular, Vue.js, React.

### 3.4 Vue.js

Entre Angular, React et Vue, pourquoi avons-nous choisi Vue?

Angular est pris en charge par Google, react est maintenu par Facebook et vue est créé par Evan You, un ancien employé de Google. Vue.js est le plus jeune d'entre eux. Bien qu'il n'y ait pas de soutien de la part des grandes entreprises, ce framework se développe très rapidement et devient de plus en plus populaire. Vue.js a maintenant plus d'étoiles sur Github que React.

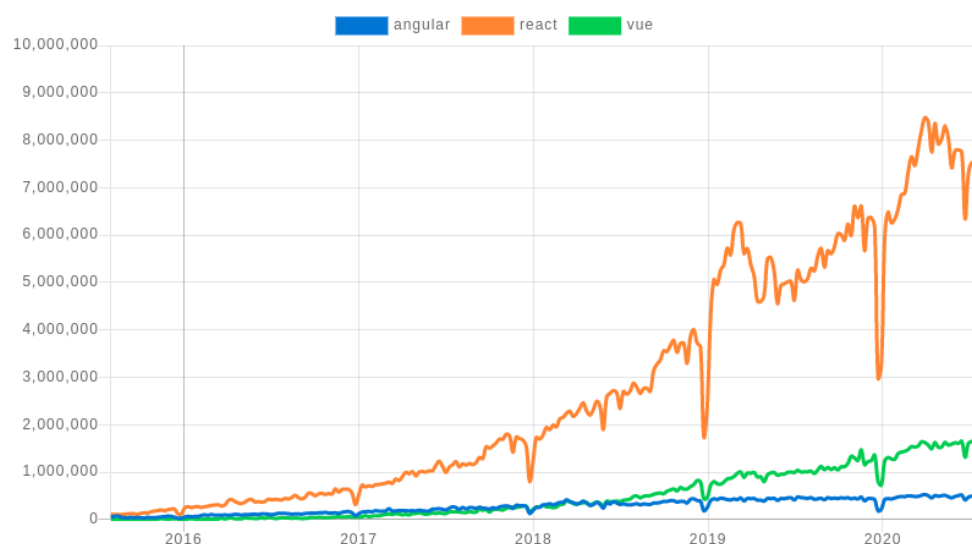


Figure 7 - Npm Téléchargements

Pour utiliser Angular 2+, nous devons apprendre le langage TS, et pour utiliser React, nous devons comprendre JSX et ES2015. Vue est relativement facile à démarrer. Nous n'avons besoin que d'une base HTML et Javascript pour investir rapidement dans le développement en lisant le site officiel. Il s'agit d'un framework JavaScript très simple, direct et facile à utiliser conçu pour simplifier le développement Web. C'est l'une des raisons pour lesquelles il est très populaire parmi les petites et moyennes entreprises.

Bien sûr, Vue n'est pas la balle en argent. En utilisant React et Angular, nous pouvons faire du développement multiplateforme. Nous pouvons utiliser React Native pour développer des applications natives pour iOS et Android. Le NativeScript d'Angular a été adopté par des applications natives, en particulier le framework Ionic a été largement utilisé dans la création d'applications hybrides. Mais Vue n'est pas parfait à cet égard et ne peut être utilisé que pour le développement Web.

Pour notre projet, il s'agit encore d'un petit projet et il n'y a pas d'exigence multiplateforme.

De plus, vue est facile à apprendre, nous avons donc choisi Vue.js comme cadre de développement front-end.

### 3.5 FeathersJs vs NestJs

Express et Koa sont les frameworks back-end les plus basiques pour Node.js. Mais parce que c'est trop basique, la création d'une application nécessite toujours d'écrire beaucoup de code d'échafaudage, tant d'autres frameworks sont apparus au-dessus d'eux pour réduire l'écriture de ce type de code. Les graphiques suivants comparent le nombre d'étoiles ajoutées sur Github de différent Node.js framework au cours des 12 derniers mois.

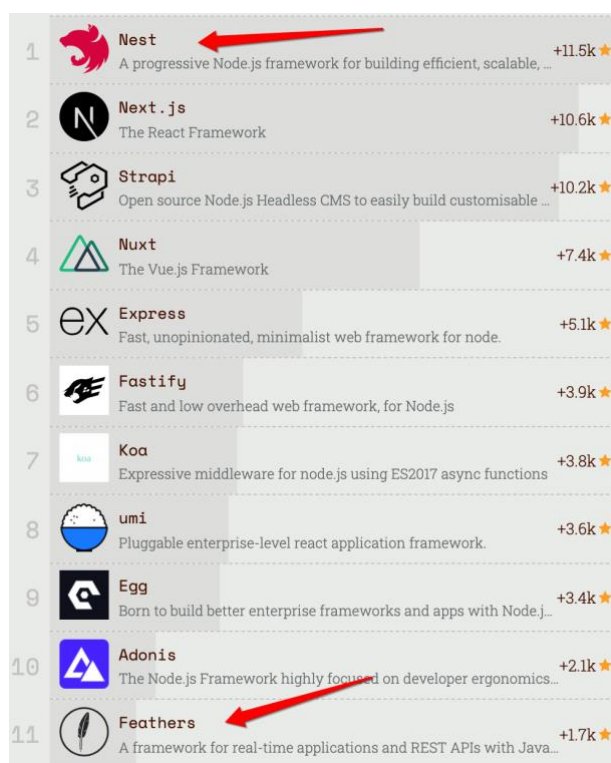


Figure 8 – Node.js framework

Feathers est un framework Web léger pour créer des applications en temps réel et des API REST base sur Express4.0.

La plupart des architectures Web en temps réel n'atteignent des performances en temps réel que lors de la transmission de données au client. Interagissez avec le service via l'interface REST, puis recevez des événements via websocket. Et Feathers vous permet d'envoyer et de recevoir des données via websocket, et même d'abandonner REST, et d'utiliser simplement websocket pour communiquer. Il place le temps réel dans une position importante, rendant notre application élégante et légère. Il s'agit d'une solution en temps réel idéale pour les appareils IoT.

La base de l'application Feathers est l'objet app, créé avec 'const app = feathers ()'

. Il existe plusieurs méthodes dans l'objet app, et la méthode 'app.use()' est utilisée pour inscrire les services. Les services sont au noyau de l'application Feathers, et gèrent également l'accès aux données. Feathers est une solution très flexible pour créer une architecture «orientée services» (SOA), c'est un bon choix pour créer des microservices node.js.

La principale caractéristique de NestJs est qu'il supporte complètement TypeScript. JavaScript est un langage faiblement typé. La conversion de type peut être effectuée librement, et la vérification de type n'est pas rigoureuse, ce qui rend l'application difficile à maintenir. Bien que ES6, ES7 et ES8 soient apparus ces dernières années, ce qui peut aider les ingénieurs à améliorer la qualité des programmes, il existe toujours un écart avec les langages fortement typés. Ainsi, TypeScript est apparu. Nous pouvons utiliser TypeScript pour écrire du code fortement typé, puis utiliser Babel Convertir vers la version spécifiée de JavaScript (version ECMAScript)

Nestjs crée un cadre Web similaire à java spring MVC (Model View Controller), qui utilise une méthode de séparation de la logique métier, des données et de l'affichage de l'interface pour organiser le code afin de réduire le couplage et d'augmenter la réutilisabilité.

Il fait également référence au modèle de conception de l'injection de dépendances et à l'idée de modularité dans Angular. L'injection de dépendances consiste à passer des variables d'instance dans un objet, de sorte que la classe appelante ne dépende que de l'interface et non de la classe d'implémentation spécifique, qui peut réaliser l'inversion de contrôle et réduire le couplage.

Son style de syntaxe est similaire à Angular et Java Spring. Beaucoup de syntaxes de décorateurs sont utilisées. Avec les décorateurs, les classes, les méthodes et les attributs peuvent être décorés de manière pratique, ce qui est plus flexible que l'héritage. De plus, grâce à la méthode d'annotation, des métadonnées peuvent être introduites dans Typescript pour réaliser une métaprogrammation.

Nest fournit également un bon support pour la programmation AOP (Aspect-Oriented Programming), qui résout le problème de la réutilisation du code de l'horizontale, en séparant principalement certains codes non commerciaux, tels que les statistiques de journal, l'analyse des performances, la gestion des exceptions, le contrôle de sécurité, etc. Le code peut être découpé dynamiquement dans la méthode et l'emplacement spécifiés de la classe au moment de l'exécution. Plus précisément, sur la base du service, nestjs complète le middleware, les filtres d'exception, les tuyaux, les gardes et les intercepteurs en fonction du niveau de traitement. Des couches de traitement sont effectuées entre les fonctions de traitement réelles. L'ordre d'exécution est (request) ---> middleware ---> Guards ---> interceptors ---> pipes---> (controller) ---> interceptors ---> Exception filters->(response).

Pour les raisons ci-dessus, nous avons décidé de changer le programme back-end de js en ts. Et migré du framework featherJs vers nestJs, l'adaptateur de base de données utilisé est également passé de sequelize matching featherJs à typeorm matching nestJs

## 4. Mise en œuvre

### 4.1 Méthodologie Agile

Notre équipe utilise une combinaison de Kanban et Scrum pour un développement Agile.

Que signifie Agile , Kanban et Scrum ?

Agile est une méthode itérative structurée qui est principalement utilisée dans la gestion de projet et le développement de produits. Selon les caractéristiques de variabilité du développement de produits, il permet à l'équipe de réagir et de changer à tout moment sans s'écarter de la voie normale du projet.

Les équipes Scrum livrent généralement le produit final à intervalles fixes des Sprints. Leur approche consiste à créer des tâches récurrentes pour collecter et intégrer rapidement les commentaires des clients. L'équipe Scrum adopte des rôles spécifiques, crée des outils spéciaux et tient des réunions régulières pour maintenir le projet en cours.

Kanban nous permet de visualiser le travail en cours, de limiter la grande quantité de travail en cours et de maximiser l'efficacité (ou d'optimiser les processus). L'équipe peut réduire efficacement le temps entre le début du projet (ou de l'exigence) et la fin en utilisant Kanban et en améliorant continuellement son flux de travail. Kanban est très utile pour les équipes avec des tâches de projet complexes et des priorités claires.

Scrum nécessite un haut degré de contrôle sur le contenu global du travail, tandis que Kanban est plus flexible



Figure 9 - Méthodologie Agile

Nous faisons des sprints d'une semaine. Mais nous ne publions pas une version après le sprint comme la méthode traditionnelle Scrum. Nous spécifions à l'avance toutes les exigences à développer pour une certaine version dans la liste des tâches, puis la publions une fois le développement terminé. Il n'y a pas de période fixe, mais cela prend généralement 1 à 2 mois



Nous utilisons Jira comme outil de gestion de projet.

Il y a deux tableaux dans Project Esendeo :

- Tableau Demand est utilisé pour collecter les besoins commerciaux des clients et est principalement utilisé par les clients. Leur kanban contient 7 colonnes : TO REFINE, DEMANDS IN PROGRESS, DELIVERED TO DEV, DEVELOPPING, QA TESTING, BUSINESS VERIFYING, DONE.
- Tableau Dev est utilisé pour suivre et gérer les tâches entrant dans le processus de développement, principalement utilisé par les développeurs et les testeurs. Leur kanban contient 6 colonnes: TO DO, DOING, REVIEW, QA, BUSINESS VERIFYING, DONE

Les cartes de différents status sont placées dans différentes colonnes. Lorsque le statut de ces cartes change, l'utilisateur peut faire glisser les cartes vers la colonne correspondante.

Dans le tableau Demande, nous pouvons voir toutes les exigences, mais généralement le tableau du développeur affiche uniquement la liste des tâches du sprint actuel, ce qui peut rendre le développement plus ordonné. Toute l'équipe peut partager le tableau.

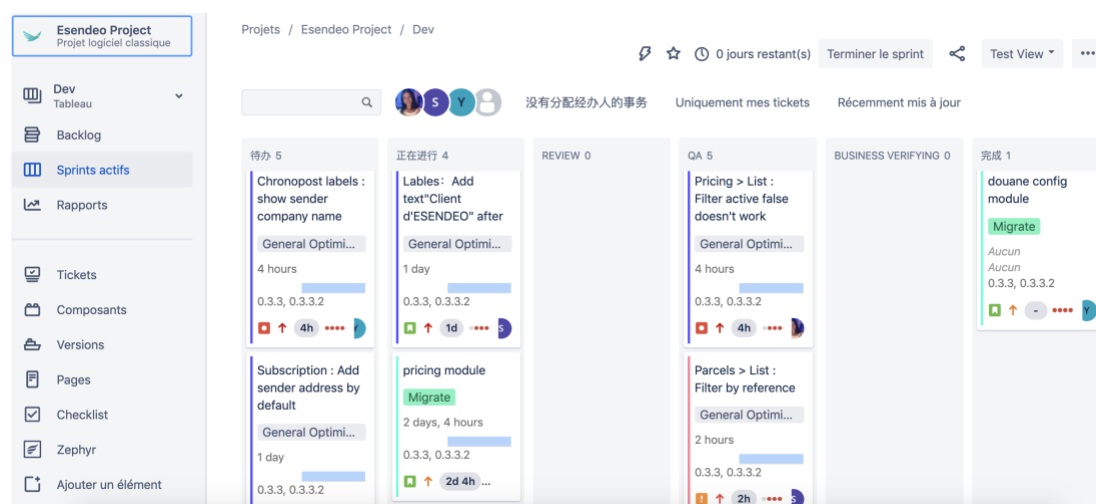


Figure 10 - Sprints Jira

Sur chaque carte, nous pouvons définir Story Point, qui est une estimation de l'investissement en temps nécessaire pour accomplir une certaine tâche, ou il peut être utilisé pour décrire la difficulté du développement. Après avoir terminé la tâche, nous pouvons indiquer le temps réel utilisé sur la carte et joindre un commentaire, qui peut être utilisé pour la révision et le résumé une fois le sprint terminé.

Nous pouvons également hiérarchiser les cartes afin que les cartes soient affichées dans différentes couleurs et ordre de priorité, ou créer ticket (y compris BUG, TASK, STORY, etc.), version, histoire longue (Perspective d'un utilisateur sur un besoin sur plusieurs cartes fonctionnelles), etc.

Dans la colonne Backlog, nous pouvons voir la progression du développement d'un certain composant ou d'une certaine version du projet.

## 4.2 DevOps

Le terme DevOps vient de la combinaison du développement et des opérations. DevOps met l'accent sur la manière d'organiser efficacement l'équipe pour compléter la gestion du cycle de vie du logiciel grâce à une collaboration et une communication automatisées avec les outils, afin de fournir des logiciels plus stables, plus rapidement et plus fréquemment.

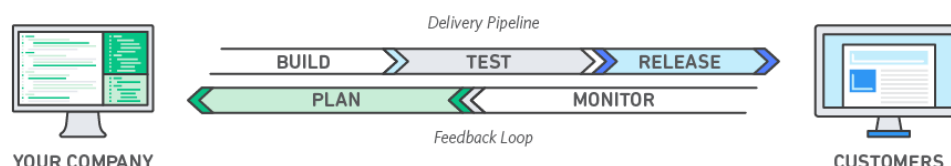


Figure 11 – Processus DevOps

Pendant le processus de développement, nous utilisons principalement les outils suivants pour collaborer.



Figure 12 – Outil DevOps qu'on utilise

Le processus de mise en œuvre de l'intégration continue est le suivant.

Le développeur clone le code dans local et ouvre une nouvelle branche pour l'édition. Après avoir terminé le codage, ils soumettent les modifications à git. Le serveur CI Jenkins surveille le référentiel de code et extrait les modifications lorsqu'elles se produisent. Ensuite, le serveur CI extrait ces modifications pour construire, exécuter des tests unitaires et des tests d'intégration. Le serveur CI informera immédiatement l'équipe si la construction est réussie ou non. Si la construction échoue, il enverra une alerte à l'équipe pour que l'équipe développement le résoudra.

Nous avons 4 branches à long terme dans Git :

Master: Publier des changements fonctionnels majeurs(ex : 0.1, 0.2), le plus stable

Prod: Version pour les tests en ligne

Deploy: Version en ligne, pour les clients finaux

Next: Publier des fonctions les plus nouvelle

## 5. Mon Travail

### 5.1 Mon rôle

En tant que stagiaire full-stack, mes tâches incluent: le développement de nouvelles fonctionnalités, l'écriture de tests, la maintenance du site Web, la migration de code.

Les trois développeurs du groupe n'avaient pas de division claire du travail pour les tests ou la maintenance. Le rôle de chacun est le même. Sans doute mon tuteur est plus expérimenté donc il fait souvent des tâches plus difficiles et il gère les autres projets aussi.

Par exemple, si je veux développer un nouveau module, je dois d'abord écrire du code de fonction, ensuite, j'écris un test e2e ou un test unitaire pour le code que j'ai écrit, puis je déploie le code dans la branche deploy de git et j'attends que le chef de produit examine et accepte ce module. S'il n'y a pas de problème, mon tuteur passera en revue le code et publiera cette partie du code en ligne, c'est-à-dire la branche prod. S'il y a un problème, ou un bug apparaît après la sortie, je continue à corriger ou optimiser cette partie.

La chronologie ci-dessous montre le travail principal de mon stage. Certaines corrections de bugs et modifications fonctionnelles mineures ne sont pas incluses.

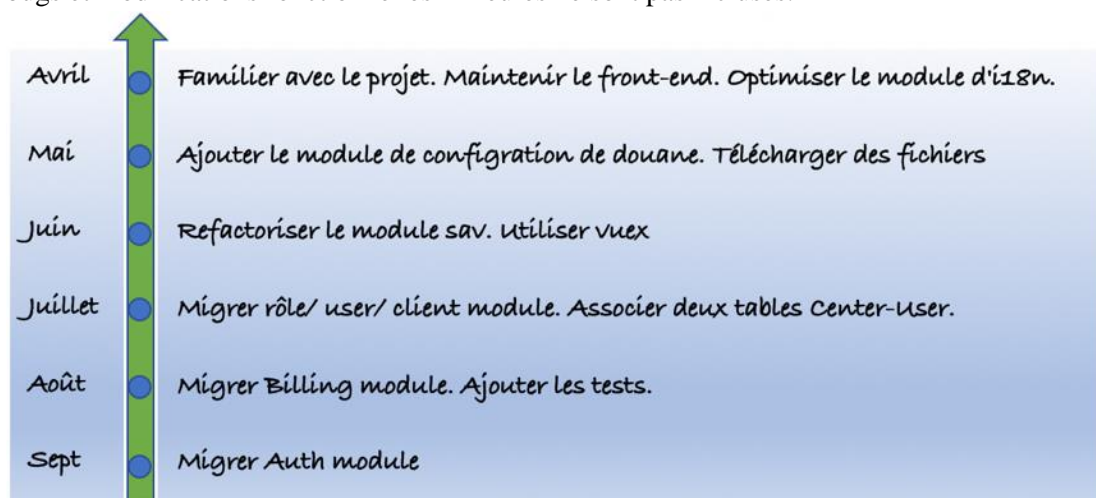


Figure 13 – Chronologie du stage

### 5.2 Mes travaux réalisés

Comme mentionné précédemment, nous utilisons principalement git pour le développement collaboratif. Nous utilisons la commande rebase au lieu de la commande merge pour fusionner les branches, ce qui peut rendre les branches plus claires et plus propres. Un nœud de commit correspond généralement à une carte de tâche sur Jira.

Le site esendeo est divisé en trois sous-projets :

Front: esendeo-web

Back: esendeo-api + esendeo-api-v2

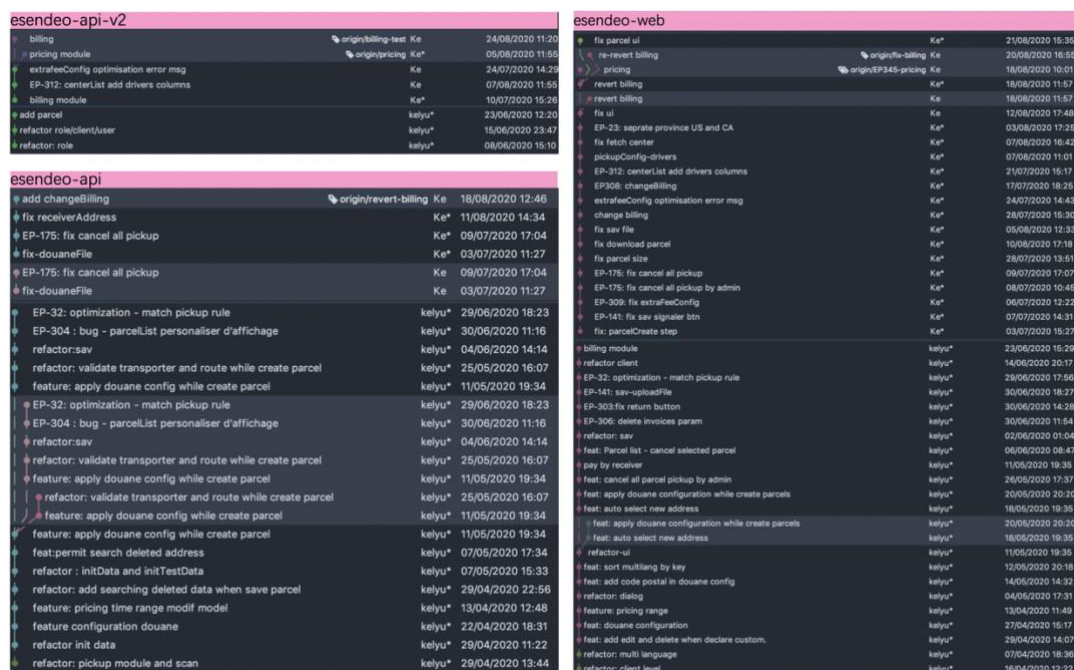


Figure 14 - Git Commit Graph personnel

Quand j'ai commencé le stage, j'avais d'abord besoin d'apprendre les technologies utilisées dans le projet, telles que le framework vue, postgresql et feathersJS, et de comprendre le projet. Mon tuteur m'a d'abord assigné quelques tâches simples de maintenance frontale.

Par exemple, j'ai besoin de refactoriser le module de langage. Notre projet prend en charge trois langues: chinois, anglais et français, donc ce que je dois faire est d'utiliser des modules linguistiques pour une gestion unifiée du texte statique utilisé dans le projet, de sorte que lors du changement de paramètres de langue, l'affichage du texte de l'ensemble du projet puisse être automatiquement commuté. Ici, nous utilisons le composant vue-i18n fourni par vue.

Et j'ai besoin d'unifier certains composants de l'interface utilisateur, comme le remplacement des boutons de suppression sur toutes les pages par les mêmes composants 'DeleteButton.vue'. La composanteisation de vue évite la duplication de code. Lorsque nous l'utilisons, nous pouvons utiliser la balise HTML <delete-button> pour introduire le composant dans la position correspondante.

Ensuite, je décrirai en détail le développement de certains modules plus importants.

### 5.2.1 Nouveau Module - Douane Configuration

Les paramètres douaniers appartiennent au module de configuration de l'administrateur.

Les paramètres douaniers du site Web d'origine sont codés en dur, c'est-à-dire que seuls les pays et les transporteurs spécifiques inscrits dans le code doivent être dédouanés. Nous devons modifier le code si nous voulons changer les paramètres, il est donc difficile d'étendre. Et sans page visualisée, nous ne pouvons apprendre qu'à partir du code ou de la documentation quels transporteurs ou pays ont besoin d'un dédouanement.

De plus, une nouvelle fonction doit être ajoutée. Dans certains cas, nous devons ajouter un document de dédouanement au format PDF au client. Nous espérons configurer ce fichier et fusionner ce fichier avec le fichier d'étiquette du colis. Une fois que l'utilisateur a passé la commande, il peut tous les télécharger dans un seul fichier.

Pour ces besoins, nous avons décidé de développer un nouveau module DouaneConfig.

Le chef de produit a déjà donné le style de la page frontale. Il contient principalement deux pages: DouaneConfigList permet d'afficher la liste d'informations, DouaneForm permet de créer ou de modifier, et un fichier de routage est nécessaire pour naviguer dans différentes pages.

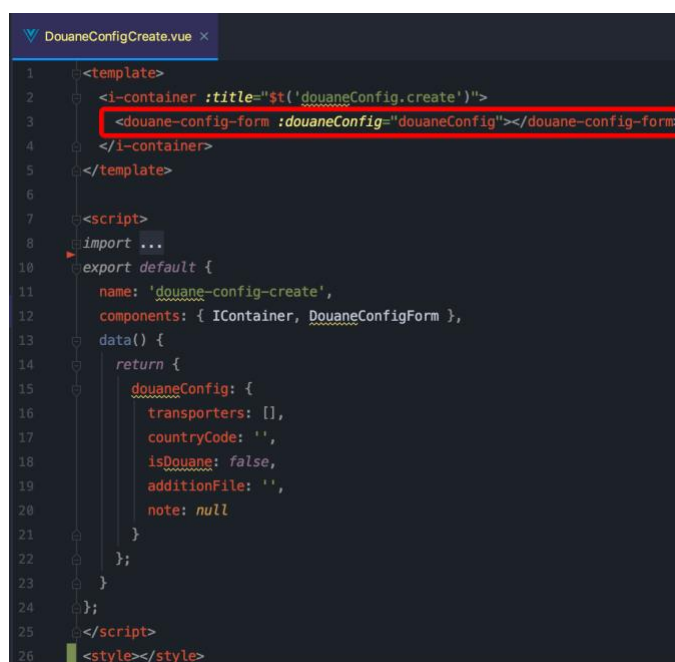
Transporteur	Country	Need Douane	Code Postal	Addition File	Message	Actions
COLISSIMO CHRONOPOST	YT	Actif				
COLISSIMO CHRONOPOST	GY	Actif				
COLISSIMO CHRONOPOST	RE	Actif				
COLISSIMO CHRONOPOST	MQ	Actif				
COLISSIMO CHRONOPOST	GP	Actif				
COLISSIMO	GB	Actif				
COLISSIMO DPD UPS GLS	CH	Actif		vosTimbres (2).pdf		
CHRONOPOST	GB	Actif				
COLISSIMO	GB	Actif				
UPS	GB	Inactif				
COLISSIMO DPD UPS GLS	RE	Actif	75* 75003 135* 280* 4*	Blender Cheatsheet v1.pdf		
CHRONOPOST						

Lignes par page: 10 21-30 de 32

Figure 15 – douaneConfigList

Étant donné que la page de création et la page d'édition sont dans des chemins différents, mais que leur contenu de page est presque le même, j'extrais les parties communes en un seul composant DouaneForm, qui peut être facilement référencé à plusieurs endroits .

Comme indiqué dans le code ci-dessous, Vue a personnalisé une sorte de fichier .vue, qui nous permet d'écrire HTML, JS, CSS (enveloppé avec des balises <template><script> <style> dans le fichier) dans un fichier pour réaliser le composant Empaqueté, un fichier .vue est un composant indépendant.



```

1 <template>
2   <i-container :title="$t('douaneConfig.create')">
3     <douane-config-form :douaneConfig="douaneConfig"></douane-config-form>
4   </i-container>
5 </template>
6
7 <script>
8   import ...
9   export default {
10     name: 'douane-config-create',
11     components: { IContainer, DouaneConfigForm },
12     data() {
13       return {
14         douaneConfig: {
15           transporters: [],
16           countryCode: '',
17           isDouane: false,
18           additionFile: '',
19           note: null
20         }
21       }
22     }
23   };
24 </script>
25
26 <style></style>

```

Figure 16 - Référence des composants

Tout d'abord, je dois concevoir la relation de base de données. Nous avons utilisé Postgresql, qui est une base de données relationnelle. Dans l'entité Douaneconfig, les champs transporteur, countryCode et additionFile sont obligatoires. J'ai également ajouté isActive, note (lorsque le dédouanement est requis, nous pouvons laisser un message au client pour demander), needShowDimension (lors de la commande, nous jugeons s'il faut laisser le client remplir le volume de colis en fonction de cela) pour une éventuelle extension future.

Puis est l'écriture du code logique back-end, c'est-à-dire l'écriture du fichier de controller et service.

Les «services» sont au noyau de chaque application Feathers. Les services sont des objets JavaScript (ou des instances de classes ES6) qui implémentent certaines méthodes. Ces méthodes sont des méthodes CRUD prédéfinies que notre objet de service peut implémenter (ou qui ont déjà été implémentées par l'un des adaptateurs de base de données). Chaque adaptateur de base de données est une implémentation de l'interface de service Feathers. Nous utilisons Sequelize comme adaptateur ici.

En d'autres termes, le framework a déjà écrit les opérations de base, et je n'ai besoin que d'ajouter des méthodes spéciales. Par exemple, je dois implémenter la fonction de téléchargement de fichier ici. Nous téléchargeons le fichier en tant qu'objet sur le serveur cloud amazon s3 (Amazon Simple Storage Service). Nous pouvons télécharger, supprimer et parcourir des fichiers via des programmes.

Maintenant que les fonctions de base du module sont terminées, il ne reste plus qu'à appliquer cette configuration lors de la passation de commande, c'est-à-dire à ajouter le jugement logique des affaires douanières dans le module Parcel. Plus précisément, une fois que

L'utilisateur a rempli l'adresse et sélectionné le service, nous envoyons d'abord une demande au backend pour demander s'il existe une configuration correspondante dans la base de données. Si tel est le cas, cela signifie que le dédouanement est nécessaire, un formulaire de douane sera affiché, laissant l'utilisateur remplir les informations douanières du colis. Si cette configuration douanière contient des fichiers supplémentaires, lors du téléchargement de l'étiquette, nous fusionnerons tous les fichiers et laisserons le client télécharger.

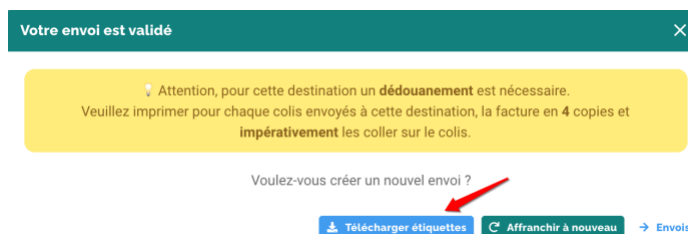


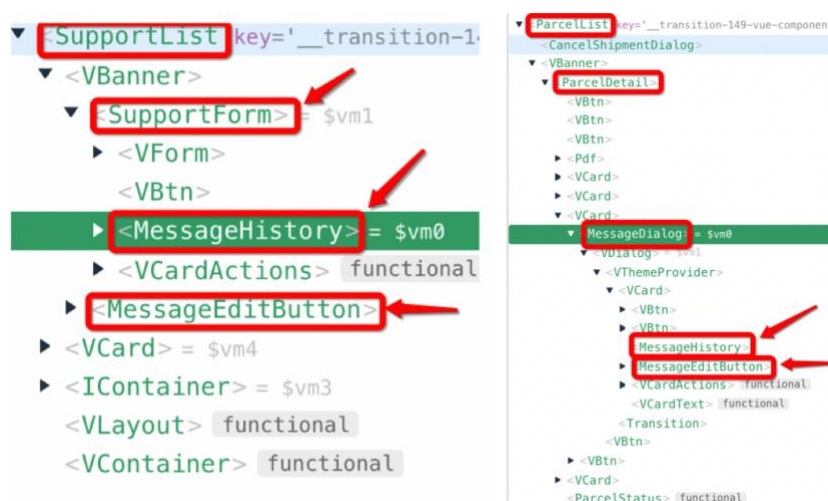
Figure 17 - Page de fin de commande

### 5.2.2 Refactoring Module - SAV

Ce changement ne concerne que le front-end. Puisque notre projet est séparé du front-end et du back-end, il suffit de changer l'affichage de l'interface, c'est-à-dire de modifier le projet esendeo-web.

Sur la page de sauvegarde, nous pouvons envoyer un message à l'administrateur et afficher les informations détaillées de toutes les sauvegardes historiques, y compris le numéro de commande associé, la description du problème rencontré, les fichiers téléchargés et la liste des conversations avec l'administrateur, etc. Désormais, le client souhaite afficher de brèves informations sur le service après-vente directement sur la page des détails de la commande, y compris la liste de dialogue et les coordonnées du service après-vente.

Les pages supportList et parcelList contiennent plusieurs composants, et leur structure est la suivante.





*Figure 18 – Les Components Vue*

Bien que les composants de page puissent être imbriqués couche par couche très facilement, le transfert de données entre les composants n'est pas si facile. Habituellement, nous pouvons transmettre les données statiques ou dynamiques du composant parent au composant enfant unidirectionnel via l'attribut 'prop', et le composant enfant peut déclencher la méthode de surveillance des événements du composant parent pour modifier les données du composant parent via '\$emit (eventName, data)'.

Les données peuvent être transmises entre le composant grand-parent et le composant fils via l'attribut 'prop' imbriqué. Nous pouvons également lier '\$attrs' et '\$listeners' au composant fils lorsque le composant parent référence le composant fils, puis utiliser '\$emit' dans le composant fils pour déclencher le changement de valeur du composant grand-père.

La situation à laquelle je suis confrontés est :

- Dans le composant <parcel-list>, le composante <message-history> et le composante <message-edit-button> doivent se transférer les données du message, c'est-à-dire le transfert parent-enfant.
- Dans le composant <support-list>, les données de message doivent être transférées entre le <support-form> et <message-history> et <message-edit-button>, c'est-à-dire le transfert de données de composants entre parent et enfant et le transfert de données de composants entre frères.

Il est relativement facile de mettre en œuvre les deux séparément, mais il est très difficile de faire partager l'état de plusieurs composants si le flux de données des deux doit être pris en compte en même temps. Le passage de paramètres via 'prop' ne permet pas le transfert d'état entre les composants frères. Et référencer directe aux composants parents et enfants ou modifier et synchroniser plusieurs copies d'état via des événements sera très lourd. Ces modèles sont très fragiles et conduisent souvent à un code non maintenable. J'ai donc décidé à utiliser vuex. Autrement dit, l'état partagé des composants est extrait et géré en mode singleton global.

Le noyau de chaque application Vuex est le 'store'. 'Store' est essentiellement un conteneur, qui contient la plupart de l'état de notre application. Nous pouvons obtenir l'objet d'état via store.state et déclencher le changement d'état via la méthode store.commit.

Si la code javascript est complexe et que de nombreux composants doivent traiter certaines données communes en même temps, vuex peut être utilisé pour extraire les données communes et rendre l'application plus organisée , afin d'atteindre l'objectif de découplage.

### **5.2.3 Migration**

En juillet, nous avons commencé la migration de feathersJS vers nestJS

Bien qu'il s'agisse d'un framework nodejs back-end remplacé, le front-end a également besoin de quelques modifications correspondantes. Parce que nous avons changé l'adaptateur de base de données de sequelize correspondant à feathers à typeorm correspondant à nest. Cela conduit à des changements dans l'écriture des paramètres de requête passés du front-end au back-end.

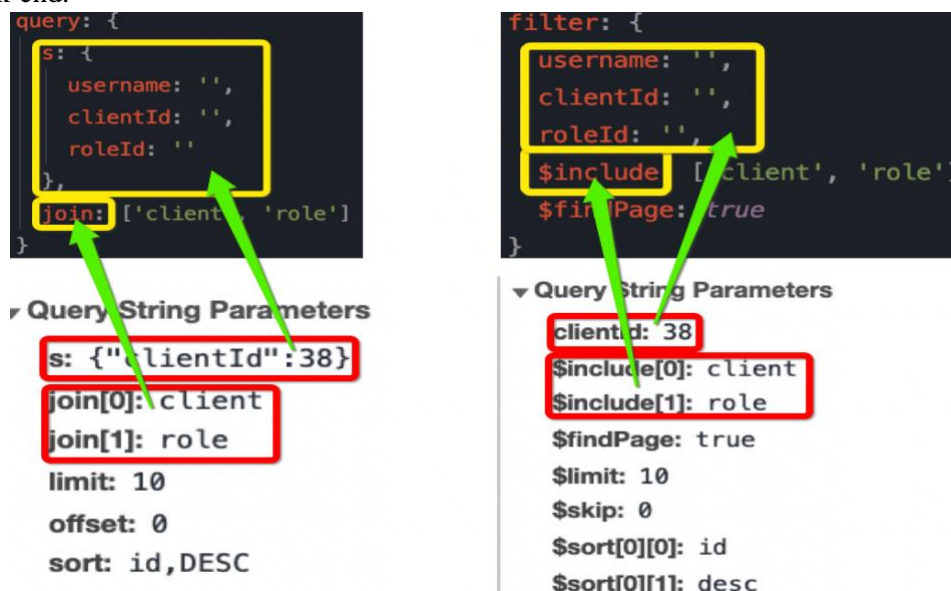


Figure 19 – Changement Front

Il y a relativement peu de changements dans le front-end et plus de changements dans le back-end. Par exemple, la structure de fichier du projet, la structure du module, l'écriture de l'entité, l'utilisation du décorateur, et la nécessité d'utiliser dto (objet de transfert de données) lors de la modification des données, l'utilisation des gardes, des filtres, des intercepteurs pour remplacer les fonctions de hook d'origine, etc.

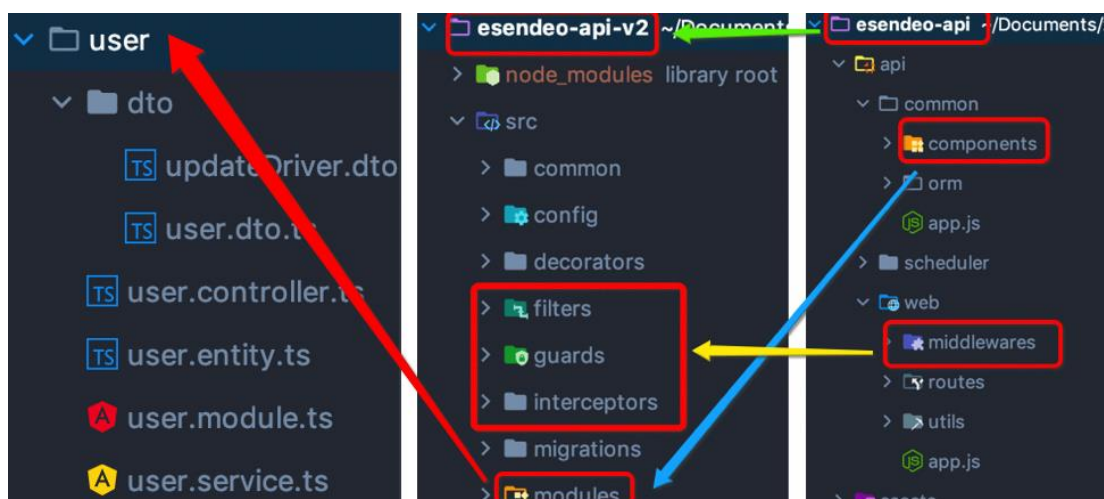


Figure 20 – Changement Back

#### 5.2.4 Migration - client/ user/ role

Les trois modules Client et User et Role sont étroitement liés, j'ai donc effectué la migration ensemble. Un client peut créer plusieurs utilisateurs et chaque utilisateur est associé à un rôle.

## ***RoleGuard***

Nous avons maintenant quatre rôles: Admin, Client, Driver, Public. L'ajout, la suppression, la modification et la vérification des rôles ont été implémentés par le framework et peuvent être appelés directement. Ce qu'il faut faire attention, c'est que nous devons juger les rôles des utilisateurs pour déterminer leurs droits d'accès à certains contenus.

La fonction hooks est utilisée comme middleware dans Feathers, qui peut enregistrer et traiter des données avant ou après le retour de la méthode de service. Dans Nest, nous utilisons la garde. La garde détermine si une demande donnée est traitée par le gestionnaire de routage en fonction de certaines conditions (telles que les autorisations, les rôles, les listes de contrôle d'accès, etc.) qui apparaissent au moment de l'exécution. C'est ce qu'on appelle généralement l'autorisation.

Dans les applications Express traditionnelles, l'autorisation est généralement gérée par un middleware, tel que le Feathers. Si la requête actuelle n'est pas autorisée, le middleware actuel n'appellera pas le middleware suivant pour atteindre l'objectif de blocage de la requête. Mais les responsabilités du middleware ne sont pas claires. Le middleware peut tout faire (vérification des données, conversion de format, compression du corps de réponse, etc.). En conséquence, le middleware ne peut être identifié que par son nom. Après une longue itération de projet, les coûts de maintenance seront relativement élevés.

Le garde du routeur est essentiellement une sorte de middleware. Habituellement, les gardes sont situés après le middleware, et avant que la requête ne soit traitée par la fonction de traitement. Selon le principe de responsabilité unique, le garde de routage ne peut renvoyer que vrai et faux pour décider de libérer ou bloquer la requête en cours et ne peut pas modifier l'objet de requête ou réponse, car une fois que le principe de responsabilité unique est rompu, le dépannage est plus gênant. Si nous devons modifier l'objet de requête, nous pouvons l'utiliser en conjonction avec un middleware

Je dois créer un système d'authentification basé sur les rôles.

Je crée d'abord une classe RolesGuard. Dans la fonction de traitement Guards, nous pouvons utiliser le mécanisme de réflexion pour obtenir les métadonnées sur le contrôleur :

```
const roles = this.reflector.get <string []> ('roles', context.getHandler ()).
```

Après avoir lu le rôle d'utilisateur actuel, utilisons-le pour faire correspondre le rôle avec l'autorité. Si la correspondance échoue, la requête sera bloquée, sinon la requête sera libérée.

Nous utilisons '@UseGuards (RolesGuard)' importé de package '@ nestjs / common' pour bind gardes dans les positions correspondantes, les gardes peuvent être à portée de contrôleur, à portée de méthode ou à portée global. Il suffit de décorer la position correspondante avec

des décorateurs.

Nous devons également utiliser le décorateur '@SetMetadata ('roles', ['admin'])' pour marquer les rôles autorisés à accéder. Afin de rendre le code plus lisible, nous pouvons également personnaliser le décorateur au lieu de SetMetadata:

```
export const Roles = (... roles: string []) => SetMetadata ('roles', roles).
```

De cette façon, une manière plus concise d'écrire '@Roles ('admin')' peut être utilisée pour déclarer la relation d'autorité de rôle.

Ensuite, nous devons utiliser '@Module ({.. providers: [RolesGuard] ..})' pour introduire RolesGuard sur les modules qui utilisent des gardes.

### *JwtGuard*

HTTP est sans état. Toutes les requêtes sont sans état. Cependant, il existe des situations dans lesquelles nous voudrions que nos états soient rappelés. Par exemple, dans une boutique en ligne, après avoir mis des bananes dans un panier, nous ne voulons pas que nos bananes disparaissent lorsque nous allons sur une autre page pour acheter des pommes. c'est à dire. nous voulons que notre état d'achat soit mémorisé pendant que nous naviguons dans la boutique en ligne.

Pour surmonter la nature sans état des requêtes HTTP, nous pourrions utiliser une session ou un jeton.

Dans l'authentification basée sur la session, le serveur créera une session pour l'utilisateur une fois que l'utilisateur se connectera. L'identifiant de session est ensuite stocké dans un cookie sur le navigateur de l'utilisateur. Pendant que l'utilisateur reste connecté, le cookie est envoyé avec chaque requête ultérieure. Le serveur peut alors comparer l'identifiant de session stocké sur le cookie avec les informations de session stockées dans la mémoire pour vérifier l'identité de l'utilisateur et renvoie une réponse avec l'état correspondant.

De nombreuses applications Web utilisent JSON Web Token (JWT) au lieu de sessions pour l'authentification. Dans l'application basée sur les jetons, le serveur crée le JWT avec un secret et envoie le JWT au client. Le client stocke le JWT (généralement dans le stockage local) et inclut le JWT dans l'en-tête avec chaque requête. Le serveur validerait alors le JWT avec chaque requête du client et renverrait une réponse.

La plus grande différence ici est que l'état de l'utilisateur n'est pas stocké sur le serveur, car l'état est stocké à l'intérieur du jeton côté client à la place. La plupart des applications Web modernes utilisent JWT pour l'authentification pour des raisons telles que l'évolutivité et l'authentification des appareils mobiles.

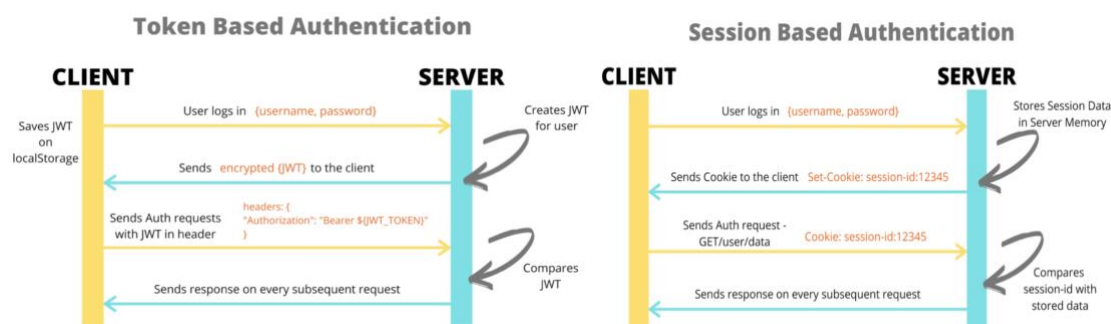


Figure 21 - Authentications

J'utilise le Passport (une bibliothèque d'authentification nodejs) pour effectuer l'authentification Jwt pour les clients. Lors de la première connexion, comme le client ne dispose pas de Jwt, nous vérifions le nom d'utilisateur et le mot de passe. Après la connexion, chaque demande doit envoyer Jwt et nous devons vérifier le Jwt envoyé par l'utilisateur, donc deux types de vérification sont requis ici. Habituellement, nous appelons ces différentes méthodes de vérification comme des stratégies, nous devons donc ici définir la 'LocalStrategy' pour vérifier le nom d'utilisateur et le mot de passe, et la 'JwtStrategy' pour vérifier le jeton. Mais j'ai utilisé une méthode d'écriture différente: j'utilise l'authentification Jwt pour toutes les requêtes, mais définisse en plus un décorateur '@IgnoreToken' pour gérer les requêtes sans jetons. Ce décorateur peut également être utilisé pour permettre aux utilisateurs qui ne sont pas connectés d'accéder aux ressources publiques qui ne nécessitent pas d'authentification.

Lors de la vérification du nom d'utilisateur et du mot de passe du client, il est important de noter que notre mot de passe n'est pas stocké en le texte d'origine. Pour la sécurité, nous utilisons la bibliothèque bcrypt avec un algorithme de hachage unidirectionnel crypté pour hacher le mot de passe transmis par l'utilisateur, puis le stocker dans la base de données. Lors de la vérification de la connexion de l'utilisateur, nous devons également crypter la valeur transmise par l'utilisateur avant de la comparer avec le mot de passe dans la base de données.

Dans le nouveau cadre, je dois utiliser la méthode de cryptage d'origine pour crypter le mot de passe transmis par l'utilisateur de la même manière, sinon, les clients de l'ancien système ne pourront plus se connecter à leurs comptes. Avant, nous utilisions directement la méthode hook hashPassword() dans le package @feathersjs/authentication-local, mais je ne peux évidemment pas continuer à l'utiliser dans le nouveau framework. Conformément à son code source, j'ai finalement écrit un nouveau méthode hashPassword() qui a la même fonction que l'ancien, en utilisant les décorateurs '@BeforeInsert' et 'BeforeUpdate' de Typeorm pour crypter le mot de passe avant de le stocker dans la base de données.

### 5.2.4 Migration et test -billing/invoice

Le module Invoice et le module Billing appartiennent au système de facturation. La Invoice est une facture pour un seul colis et la billing est une facture mensuelle, comprenant toutes les Invoices de ce mois.

Ces deux modules traitent principalement des activités de calcul et de génération de factures mensuelles, de parse des fichiers de factures et de calcul de prix ou de frais supplémentaires pour différents services.

### ***Transaction***

Lors de la génération de le Billing, nous devons récapituler les Invoices du mois en cours. Une fois le Billing générée, le billingId doit être ajouté à ces Invoices. Une telle opération impliquera des modifications dans plusieurs entités. Afin d'éviter une erreur dans le processus de mise à jour de l'entité et conduisant à une erreur de facturation, je dois utiliser transaction.

Une transaction peut être définie comme un groupe de tâches. Une seule tâche est l'unité de traitement minimale qui ne peut pas être divisée davantage. La transaction de base de données peut garantir que toutes les opérations dans le cadre de la transaction peuvent toutes réussir ou échouer toutes. Si la transaction échoue, l'effet est le même que si ces SQL ne sont pas exécutés et aucune modification ne sera apportée aux données de la base de données.

Une transaction dans un système de base de données doit conserver l'atomicité, la cohérence, l'isolement et la durabilité - communément appelées propriétés ACID - afin de garantir l'exactitude, l'exhaustivité et l'intégrité des données.

Pour deux transactions exécutées simultanément, s'il s'agit d'opérer sur le même enregistrement, des problèmes peuvent survenir. Parce que les opérations simultanées entraîneront une incohérence des données. À ce stade, il est nécessaire d'introduire un mécanisme d'isolation des données pour s'assurer qu'une seule transaction peut modifier les données à la fois et que l'une peut modifier les données après que l'autre est modifiée. Cela nécessite un verrouillage mutex sur les données. Le système de base de données fournit plusieurs niveaux d'isolation différents pour nous permettre de sélectionner les différent niveau d'isolation des transactions.

Dans nest, nous utilisons :

```
queryRunner.startTransaction (isolationLevel?: IsolationLevel): Promise <void>
```

pour démarrer les transactions et définir le niveau d'isolement, et :

```
queryRunner.rollbackTransaction ()
```

pour revenir en arrière lorsque les choses échouent.

### ***Test***

Les tests automatiques sont une partie importante d'un produit logiciel entièrement fonctionnel. Il est essentiel de couvrir au moins la partie la plus sensible du système.

À partir de ce module, nous avons commencé à ajouter progressivement des tests, y compris des tests unitaires et des tests e2e.

En ce qui concerne le test, nous avons utilisé la base de données sqlite en version1, qui peut être démarrée directement en mémoire. L'avantage est la commodité, mais l'inconvénient est que certaines opérations de base de données postgresql ne peuvent pas être utilisées dans sqlite.

Nest intègre le framework de test Jest par défaut et utilise la façon Mock pour tester. Donc, dans la v2, nous avons d'abord décidé d'utiliser mock. Mais nous nous sommes vite rendu compte que cette méthode est un peu lourde, car nous devons créer de nouvelles fausses données à tester à chaque fois, mais parfois le programme implique plusieurs modules, alors nous devons créer beaucoup de fausses données.

Nous sommes donc passés à docker. Il me faut d'abord configurer l'environnement docker. Dans docker, nous simulons l'environnement d'exécution en ligne, et la base de données utilisée est Postgresql, cohérent avec celui en ligne. Ensuite, je dois écrire un fichier sql pour créer des nouvelles tables de base de données et insérer toutes les données de test. Ensuite, je combine le framework de test Jest pour faire le tester dans docker. On stoppe le docker à la fin du test. J'ai écrit tous ces commandes dans un script pour faciliter l'exécution automatisée.

## 6. Difficultés rencontrées

Ce stage de 6 mois n'a pas été tout à fait simple, j'ai également rencontré plusieurs difficultés:

Tous les débuts sont durs. Au cours des deux premières semaines du stage, en raison du manque de connaissance du nouveau framework et de la documentation incomplète, j'ai dû comprendre la fonction du site Web directement à partir du code, et parce que je ne connaissais pas assez Vue.js, j'avais même du mal à lire le code. Dans ce cas, j'ai d'abord rapidement parcouru le site officiel et la documentation de vue, puis j'ai regardé quelques cours vidéo, ce qui m'a permis de démarrer rapidement. De plus, s'il y a un code ou un problème que je ne comprends pas, je demande mon tuteur à tout moment, il y répondra toujours avec enthousiasme et m'apportera beaucoup d'aide.

Lorsque j'étais dans une équipe agile pour la première fois, il était un peu difficile pour moi de suivre les progrès du développement. Parfois, je ne pouvais pas équilibrer qualité et efficacité. Habituellement, lorsque nous aurons un cycle de développement suffisamment long, nous accorderons plus d'attention à l'amélioration des produits pour obtenir une meilleure qualité, et lorsque nous aurons besoin de livrer des produits rapidement, nous nous concentrons sur l'efficacité. Plus précisément, j'ignore parfois un code imparfait de la tâche précédente car j'ai besoin de développer rapidement la tâche suivante. Cependant, lorsque j'ai rencontré des difficultés, mon tuteur m'a donné beaucoup de conseils et mes collègues m'ont aussi beaucoup aidé. Maintenant je me suis adapté au rythme du sprint.

Une autre difficulté est le changement de framework de back-end. Ces deux frameworks sont de nouveaux frameworks pour moi, donc je dois apprendre les deux frameworks en même temps. Je dois d'abord comprendre ce qui est implémenté dans le programme feathers, puis réfléchir à la manière de l'implémenter dans le programme nest, ce qui m'oblige à dépenser plus de temps. Mais heureusement, ils sont tous basés sur le framework express de nodejs, et les principes de base de la construction du site Web sont similaires. Utiliser les deux cadres en même temps me permet également de comprendre différentes solutions au même problème par comparaison et me permet de voir plus facilement l'essence du problème. Bien que cela demande plus d'efforts, cela en vaut la peine.



## 7. Conclusion

Ce stage de six mois m'a été très bénéfique. En plus de mettre en pratique ce que j'ai appris, j'ai également acquis un travail d'équipe et des méthodes de travail, qui sont très importants pour ma carrière.

Ce stage m'a offert l'opportunité de m'immerger de manière concrète dans le monde de l'entreprise et ainsi mesurer les différences entre le monde universitaire et le monde professionnel. Durant ma formation à l'université de Paris Diderot, j'ai pu acquérir un ensemble de savoir, de connaissances dans le domaine informatique et plus précisément dans les langages et programmation. En stage, j'ai mis ces connaissances en pratique et appris de nombreux outils et techniques pratiques. A l'école, nous prêtons attention à la formation à la réflexion et à l'apprentissage de l'ensemble du système informatique de base. Dans l'entreprise, nous utilisons divers cadres et méthodes de développement matures de haut niveau pour améliorer autant que possible la qualité et l'efficacité du développement dans un domaine.

En termes de technologie, j'ai une compréhension approfondie de l'ensemble du processus de développement Web, y compris la détermination des exigences, le développement, les tests, le déploiement et la surveillance. Je maîtrise également mieux les langages et les frameworks utilisés dans le développement Web, tels que Javascript, Typescript, HTML, CSS et Vue. J'ai également pratiqué divers outils de travail d'équipe et de développement tels que Gitlab, WebStorm, Jira, etc.

En termes de compétence personnelle, dans le processus de travail avec différentes personnes telles que les développeurs, les chefs de produit et les chefs de projet, mes compétences en communication et collaboration se sont grandement améliorées. J'ai appris à comprendre et à résoudre de nouveaux défis rapidement et de manière indépendante, et j'ai une meilleure compréhension de l'industrie informatique.

Mon stage est terminé, pourtant le site Esendeo continuera d'être maintenu et de nouvelles fonctionnalités seront ajoutées. Ce stage m'a donné un plan plus clair pour mon avenir. Node est de plus en plus populaire pour le développement de sites Web back-end. Je vais également poursuivre ma carrière dans le développement full-stack basé sur node.

# Bibliothèque

## 3.1 Restful

1 <https://blog.51cto.com/silencezone/2491494>

## 3.2 Séparation front et back :

2 <https://github.com/lifesinger/blog/issues/184>

3 <https://www.jianshu.com/p/a947bfac0a00>

## 3.3 SPA

4 <https://juejin.im/post/6844903865242877959>

5 <https://juejin.im/entry/6844903688075509774>

## 3.5 Feathers Vs Nest:

6 <https://risingstars.js.org/2019/en/#section-nodejs-framework>

7 <http://www.neusofts.com/articles/node-js-%E5%BC%80%E5%8F%91%E6%A1%86%E6%9E%B6%E6%AF%94%E8%BE%83.html>

8 <https://www.jianshu.com/p/fe2d719c6d6f>

9 <https://ithelp.ithome.com.tw/articles/10190659>

10 <https://juejin.im/post/6844903776785006605>

11 <https://www.cnblogs.com/pjl43/p/9826748.html>

## 3.1 Méthodologie agile :

12 [https://fr.wikipedia.org/wiki/M%C3%A9thode\\_agile](https://fr.wikipedia.org/wiki/M%C3%A9thode_agile)

13 <https://juejin.im/post/6844903833588465671>

## 3.2 DevOps

14 <https://www.zhihu.com/question/58702398>

15 <https://zhuanlan.zhihu.com/p/81119733>

## 4.2 Nouveau Module

16 <https://docs.feathersjs.com/api/services.html>

17 <https://www.cnblogs.com/SamWeb/p/6391373.html>

### 4.2.2 Refactoring Module

18 <https://cn.vuejs.org/v2/guide/components.html>

19 <https://vuex.vuejs.org/zh/guide/>

20 [https://blog.csdn.net/qq\\_35176916/article/details/87921097](https://blog.csdn.net/qq_35176916/article/details/87921097)

21 <https://juejin.im/entry/6844903681972781069>

### 4.2.4 Migration client/user/role

22 <https://docs.nestjs.cn/7/techniques?id=%e8%ae%a4%e8%af%81%ef%bc%88authentication%ef%bc%89>

23 <https://juejin.im/post/6844903925435498504>

24 <https://docs.nestjs.cn/7/techniques?id=%e8%ae%a4%e8%af%81%ef%bc%88authentication%ef%bc%89>

### 4.2.5 Migrate Billing et test

25 <https://docs.nestjs.com/fundamentals/testing>