

**Mode d’emploi**

RHOULAM Mehdi

Mehdi.rhoulam@epitech.eu

06.35.59.11.88

Promotion 2022

Epitech Marseille

# Contents

Table des matières

[Contents 2](#_Toc17892507)

[Lettre de voiture entrée 3](#_Toc17892508)

[Template Lettre Voiture entrée 5](#_Toc17892509)

[ Images – 5](#_Toc17892514)

[o Image.1 5](#_Toc17892515)

[Bon Livraison Entrée 7](#_Toc17892516)

[Unité Manutention Entrée 10](#_Toc17892517)

[4 : 11](#_Toc17892518)

[Etiquette UMe 11](#_Toc17892519)

[Paramètre de L’API 15](#_Toc17892520)

[Gestion images menus : 15](#_Toc17892521)

[Crée ou géré les utilisateurs : 16](#_Toc17892522)

[Crée les types de fournisseurs : 17](#_Toc17892523)

[Crée des zones de dépôts : 17](#_Toc17892524)

[Crée les types de destinataires : 18](#_Toc17892525)

[Crée les types de contacts : 19](#_Toc17892526)

[Crée les types d’articles : 20](#_Toc17892527)

[Crée ou géré les types de zones : 21](#_Toc17892528)

[Crée les pays pour destinataire : 22](#_Toc17892529)

[Crée ou modifié les litiges et / ou décision litiges : 23](#_Toc17892530)

[Bon de commande sortie 24](#_Toc17892531)

[Annexes 28](#_Toc17892532)

[Listes des annexes 28](#_Toc17892533)

# Lettre de voiture entrée

1

4

2

Liste des LVE :

3

1 : Redirection

Ce bouton fait appel à un évènement « click » de javascript, il va faire une redirection

Vers l’url Django « lvemodify » donc en html « lettrevoitureentree/modify ».

2 : Supression

Ce bouton fait appel à un évènement « click » de javascript, il va faire appel à une url

Ajax vers l’url en html « lve/modify/deletebl/ ».

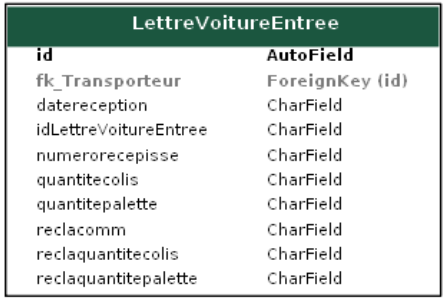
3 : Création

Ce bouton fait appel à un évènement « click » de javascript, il va faire une redirection vers

L’url Django « lveadd » en html « lettrevoitureentree/add ».

4 : Données

Toutes les données ici, viennent de la table « LettreVoitureEntree ».



# Template Lettre Voiture entrée

## 3 -

## - 1

## 2 -

## 4 -

Image.1

**Image**

## Images –

Utilisation d’un formulaire pour envoyer 2 images et les stocker dans les entrées « fichier » et « photo » dans la table lettre voiture entrée, ils sont stockés dans le dossier « media\_lve » dans le media racine et non le « static » le static sert aux fichier type css ou javascript, mais pour le stockage via la base de données le faire via le « media root » est bien plus logique.

### Image.1

Ce formulaire a pour « cible » l’url « upload\_lvef » qui fait appel à la fonction « upload\_lvef » dans « lettrevoitureentreemodify », il va se charger d’enregistrer les deux nouvelles images, stocké, il va supprimer effacer la dernière sélection faite s’il y avait déjà des enregistrements de fait.

1 -

Ici on a toutes les données pour la lettre de voiture entrée que l’on crée, dans la zone de texte identifiant, on a une variable qui s’incrémente à chaque création (automatique),

Le bouton « save » envoi une requête à « lve/modify/deletebl/ » qui renvoie vers la fonction « deletebl » dans la class « lettrevoitureentreemodifiy »

2 -

Button qui fait appel à la fonction « createbl » via l’url html « /lve/modify/createbl/ »

Située dans la class « lettrevoitureentreemodify »,

Elle a pour mission de crée des bons de livraison entrée, on utilise la variable « identifiant »

Pour crée un bon de livraison entrée avec le fk\_LettreVoitureEntree qui sera égale à « identifiant »,

Comme ça on a un bl qui va être lié à notre lv.

3 -

Redirection vers l’url Django « bonLivraisonEntreemodify » ou en html « bonLivraisonentree/modify/ » ça va être la page de rendu du bl qui aura été créé via cet lv.

4 -

Ce bouton qui se lance via un « click » va faire appel à la fonction « deletebl » se trouvant dans la class « lettrevoitureentreemodify » via l’url html « lve/modify/deletebl/ »

Il va supprimer le bon de livraison entrée lié à la lettre dans laquelle on se trouve, via un attribut « id » dans les attributs du bouton, qui va me servir à le retrouver dans les bons de livraison entrée

# Bon Livraison Entrée

6 ->

5

-> 7

A

4 ->

-> 6

--> 3

1 ->

2

9 ->

8 ->

D

C

B

A :

Appelle la fonction « left » dans « bonLivraisonentreemodify » via l’url « left\_ble/ »,

Va vérifier si une « entrée » de bon de livraison existe avant celle-là

B :

Redirige vers le template d’accueil des bl entrée « bonLivraisonentreemodify »

C :

Redirige vers le template d’accueil des bl entrée « bonLivraisonentreeadd »

D :

Appelle la fonction « right » dans « bonLivraisonentreemodify » via l’url « right\_ble/ »

Va vérifier si une « entrée » de bon de livraison existe après celle-là

1 :

Affichage de la variable « idBonLivraisonEntree » venant de la table « BonLivraisonEntree ».

2 :

Bouton qui redirige vers « clientmodify » en html « client/modify?id=1 » avec la variable « Client ».

3 :

Select combobox (widget JQuery) pour choisir un « Client » dans la BDD via une auto-completion si la liste de clients est grande.

4 :

Input (zone de texte) pour identifier la Lettre de voiture entrée qui a créé ce « bon de livraison entrée) (foreign\_key « fk\_LettreVoitureEntree »), le bouton ensuite va rediriger vers

« Lvemodify » en html « lettrevoitureentree/modify?id=1 » avec la variable « Lettre de Voiture Entrée ».

5 :

Input mettant en avant le « type de zone » à utiliser pour ce Bon de livraison,

En général il dépend du client, une fois un client choisit, le type de zone de dépôt adéquat va être automatiquement ramener en avant, car chaque client devra avoir un type de zone assigné.

6 :

« Save » va appeler la fonction attribuée à « save » via son id « #addble », qui

Va parcourir la page, est récupérer toutes les données que ce soient des « lignes de BL » où

Des données interne à la page sauf les « fichiers », via l’url « ble/modify » pour stocker les infos basiques, et « ligneble/create » va stocker les lignes du ble.

7 :

Ce bouton va créer une « Unité Manutention Entrée », elle va la lier automatique au « Bon de Livraison Entrée » qui va la crée, via l’url « bonLivraisonentree/modify/createume »

On envoi l’id du Bon de livraison entrée qui va servir de foreign-key

8 :

Formulaire html via Bootstrap avec comme cible (action) -> « upload\_blef » avec comme paramètre « enctype="multipart/form-data" » sinon Django ne reçois pas les fichiers, et le fameux « {% csrf\_token %} » sinon Django Bloque la requête Post !

Une fois les fichiers chargés on clique sur « upload » qui va faire appel à la fonction vers laquelle « upload\_blef » renvoi.

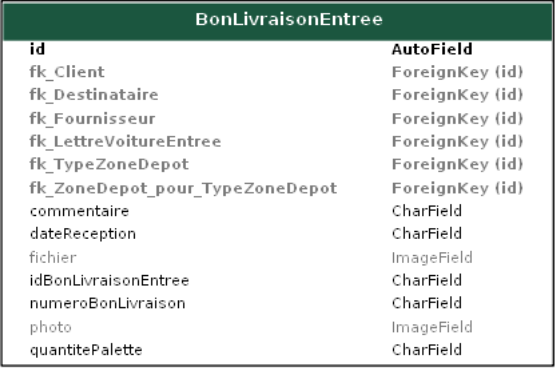
9 :

Ce bouton qui quand on clique dessus va d’abord supprimer toutes les lignes déjà présentes s’il y en a via l’url « ligneble/delete » et avec de l’algo, on va aller dans toutes les palettes (unité manutention entrée) on va voir tous les colis et comparé le « nom » des articles,

Si on retrouve le nom d’un article plusieurs fois, on va changer la quantité et le nombre de colis à chaque fois, quand on a fini on affiche le résultat dans les lignes puis on sauvegarde avec « Save ».

Données :

La bdd de cette table « bon livraison entree »



Comme illustré ici, elle détient des foreign-keys et des entrées pour stocker des fichiers.

# Unité Manutention Entrée

25

8

12

Save

24

23

19

20

21

22

18

17

16

15

14

13

11

9

10

4

7

5

6

3

2

1

1 :

Ce « bouton » va me permettre de changer la donnée « Stock » à 1 ou à 0 (bool), elle permet aux dispatcheurs de vérifier que la palette sur laquelle ils travaillent est « complète » ou pas.

2 :

Input d’affichage de la foreign\_key Bon Livraison Entrée, elle permet de savoir quel « Bons de Livraison Entrée » a créé la palette « actuelle », le bouton lui renvoie vers la page du « bons de livraison entrée » sur l’url « bonLivraisonEntreemodify » avec une url en html « bonLivraisonentree/modify/?id=3 »

3 :

Zone de choix de la zone de stockage, elles dépendent du « Type de zone » qui lui dépend du « Client », c’est aussi un select usant du widget JQuery avec auto-complet, car s’il y a beaucoup de zone, il va faire une découpe de tout ce qui ne ressemble pas à ce que l’on écrit

## 4 :

Button qui va ouvrir un nouvel onglet vers l’url « etiquetteUMe » soit « etiquetteUMe/?id=2 « en html, avec en paramètre un id appartenant à la palette

### Etiquette UMe

Dans cette étiquette on aura accès aux données de la palette tel que le nombre de colis, la zone de stockage, etc…

5 :

C’est trois boutons alignés me permettent de lancer une fonction qui va afficher ou caché des lignes dans la table si la ligne respect la condition, livrable = 0 litige, si litige = au moins 1 litige, si Tout, toutes les lignes devront être visible.

6 :

Input d’affichage de la variable « IdUniteManutentionEntree » pour savoir sur quel ume on travail variable qui s’incrémente.

7 :

Liste des articles présents dans les colis, chaque article dans une palette sera situé ici,

Et la checkbox d’un article dans cette zone, permet de filtrer les articles que l’on veut voir (1 article à la fois)

8 :

Ce bouton va dupliquer x fois le nombre de ligne que l’on va sélectionner par rapport au nombre que l’on va rentrer en paramètre dans le popup.

9 :

Ce bouton permet de faire plus rapidement ce que fait le 8, il permet de copier 1 fois la ligné choisie.

10 :

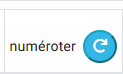
Ceci est le numéro (id) du Colis sur lequel on travail, ne pas se fier à cette valeur tant que l’on n’a pas sauvegarder le colis (cette valeur n’est que spéculative, elle peut être ensuite modifier par le serveur s’il est déjà prise), se fier la numérotation dans du colis qui elle est forcément unique si on clique sur le bouton numéroter de la page.

11 :

Zone de sélection d’article par soit le code produit, soit le nom du produit,

Une sélection d’un ou de l’autre fait en sorte que l’autre select soit rempli que du produit sélectionné dans le premier

12 :

Cette zone est automatique rempli par un bouton qui va numéroter dans l’ordre de création les colis, par ordre croissant,

Ça permet d’identifier un colis par sa palette plutôt que par son « identifiant » (ça facilite le tri).

13 :

Bouton qui va rediriger vers la palette de sortie si le colis est en cours de sortie vers le l’url « unitemanutentionsortiemodify » soit en html « umsortie/modify/?id=2 »

14 :

Bouton pas encore sur de son bon fonctionnement, il permet de réassigner un colis seulement dans la palette

15 :

Si un colis est en « sortie » il va avoir une valeur dans cet input, il s’agit de la foreign\_key pour une « Unité Manutention Sortie ».

16 :

Il s’agit du numéro de l’Unité Manutention Entrée ! = à l’id, son but est de différencier les ume dans le même ble, car à chaque ume dans un ble on incrémente de 1 le numéro de l’Um (pas encore fait)

17 :

Ceci est le code UM du destinataire venant du bon de commande sortie, via l’Unité de Manutention Sortie, si le colis est en phase de « sortie ».

18 :

Ceci est un élément qui se ferme ou s’ouvre selon la valeur du colis « confirmé » ou non.

19 :

Si l’Unité Manutention Sortie du colis est « expédié » elle à une date d’expédition, c’est celle-là que je récupère ici.

20 : Délie UMS

Ce bouton appelle un évènement javascript « click » qui va envoyer une requête Ajax en Post,

Avec l’id du Colis (ligne) qui doit être délié, et en backend on récupère le colis ayant cet « id »,

Puis on le « libère » de cet Um sortie.

21 : Delete Colis

Ce bouton va faire appel à Ajax, et envoyer l’id du colis en question, puis côté serveur, on va supprimer le colis ayant le même id.

22 :

Ce bouton va nous signifier que l’étiquette a été collé !

Mais on ne peut pas l’activer que si on a préalablement imprimé l’étiquette

23 :

Ce bouton renvoi vers une url « etiquetteUMecol » en html « etiquetteUMe/colis?id=1 »

24 :

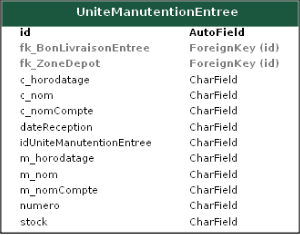
Ce bouton sert à créer un « colis » vierge, il ne peut être utilisé que une fois par rafraichissement de la page, pour éviter des erreurs de manutention.

25 :

Ce bouton va faire appel à la fonction de « tri » des colis en stock, non confirmé,

Il va appeler la fonction « sort\_colis » via le lien html « /sort\_cols/ » il prend en paramètre l’id de l’unité manutention entrée, et puis coté serveur il y a un algorithme de tri.

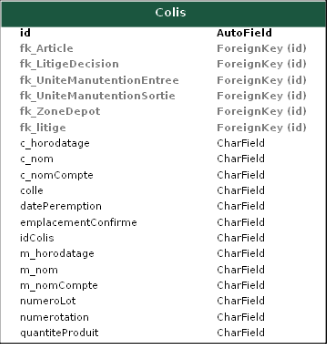
Unite Manutention Entree (Models) :



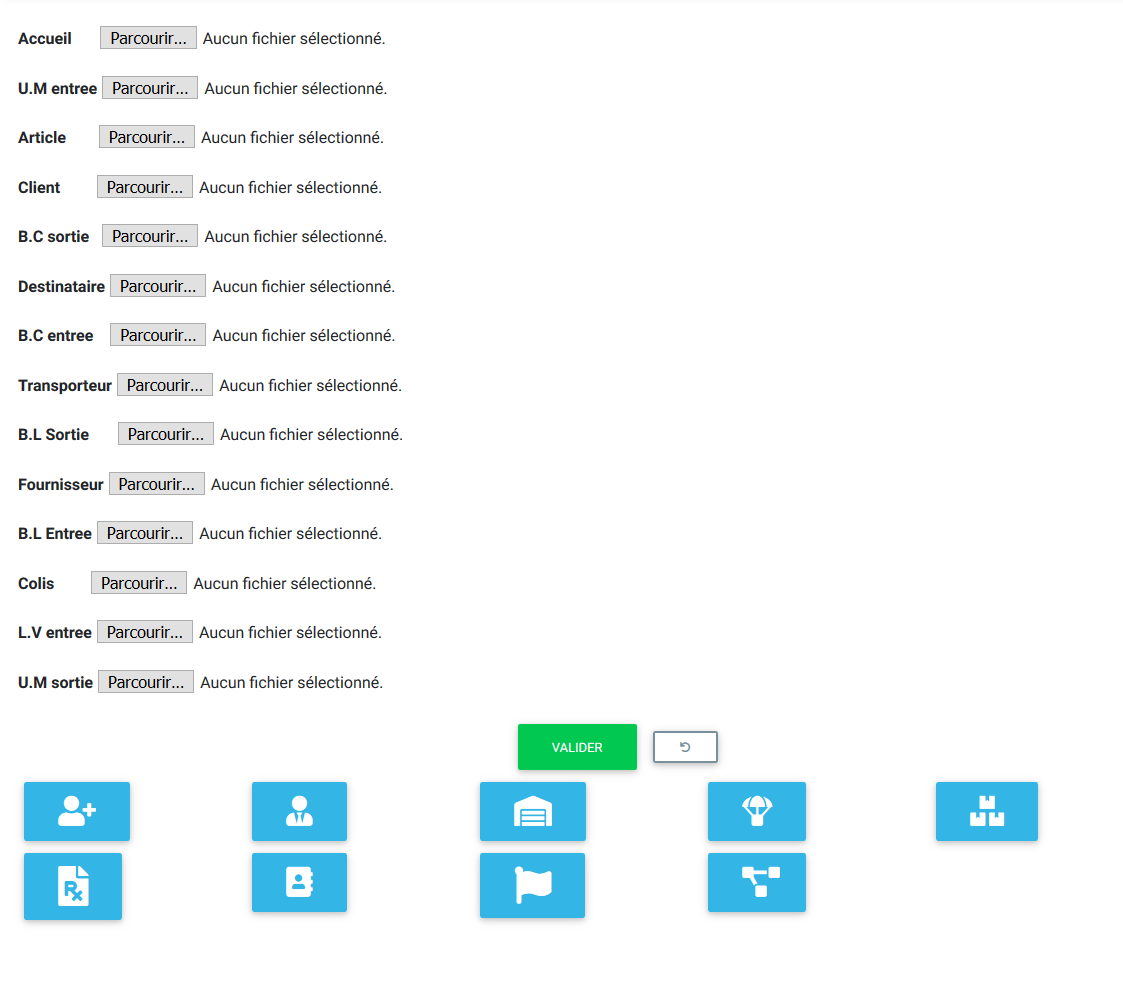
Comme on le voit les « fk » grisée permettent la création d’un objet même si un « parent » ou une information est absente, dans le but de permettre d’avoir des données non complètes,

Comme la non-existence d’une zone de dépôt ou ne pas avoir de bon de livraison, mais pouvoir quand même stocker une palette…

Colis (Models) :



# Paramètre de L’API



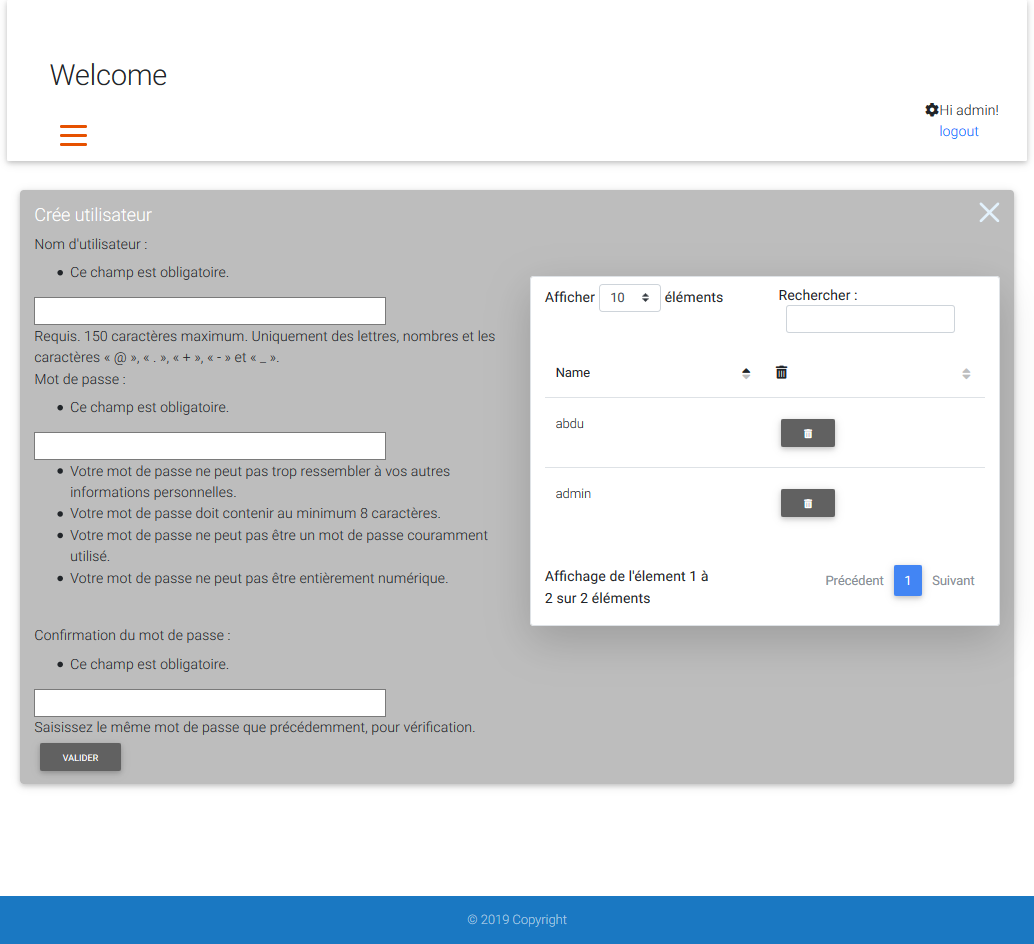
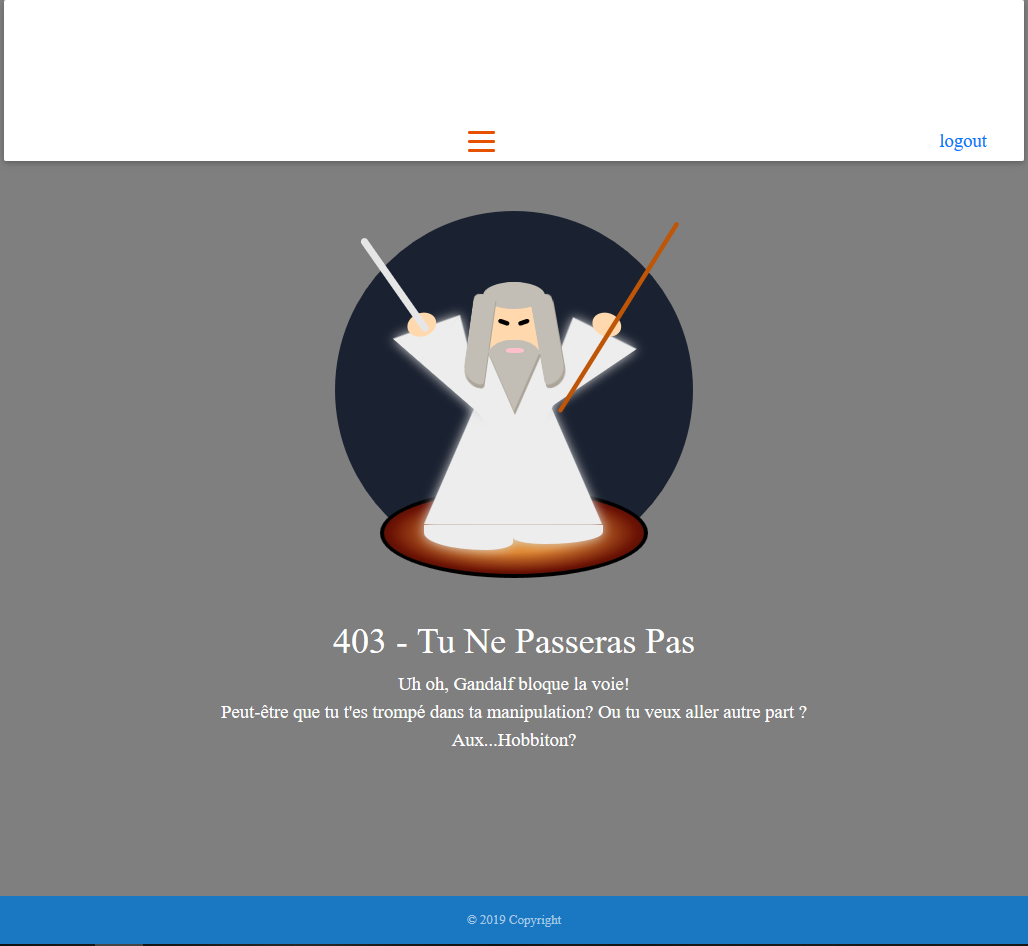
## Gestion images menus :

Ici nous avons des zones d’upload d’image, elles serviront d’icônes pour les menus,

Le bouton valider permet de mettre à jours les champs remplis, ceux qui restent vide, ne seront pas changé, le bouton à droite avec une flèche perme de supprimer les images de la bdd en même temps.

Ils sont « hoverable », et affichent donc l’image actuelle de cette donnée.

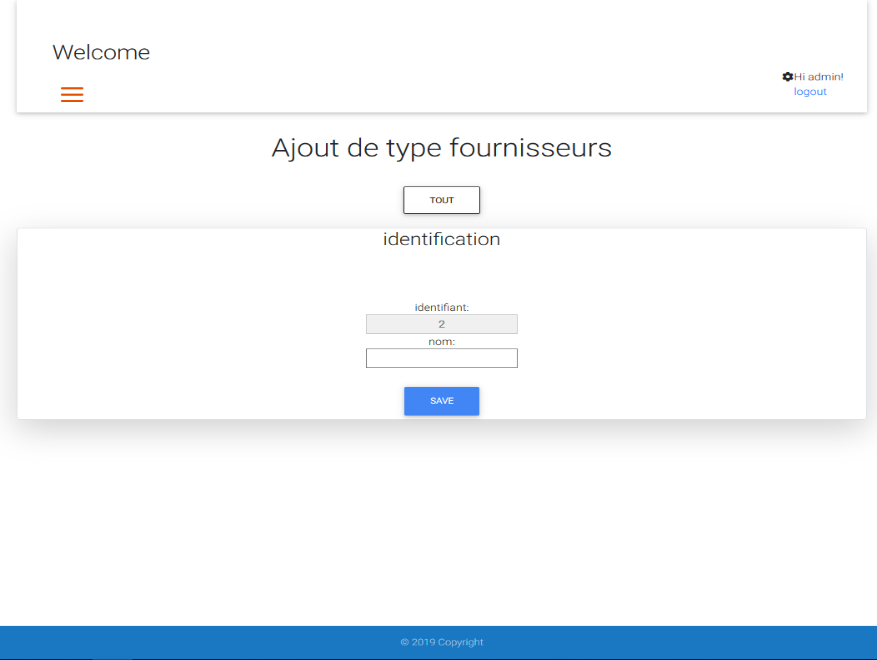
## Crée ou géré les utilisateurs :



Chaque utilisateur autre qu’un « administrateur », ne pourra avoir un accès à cette page,

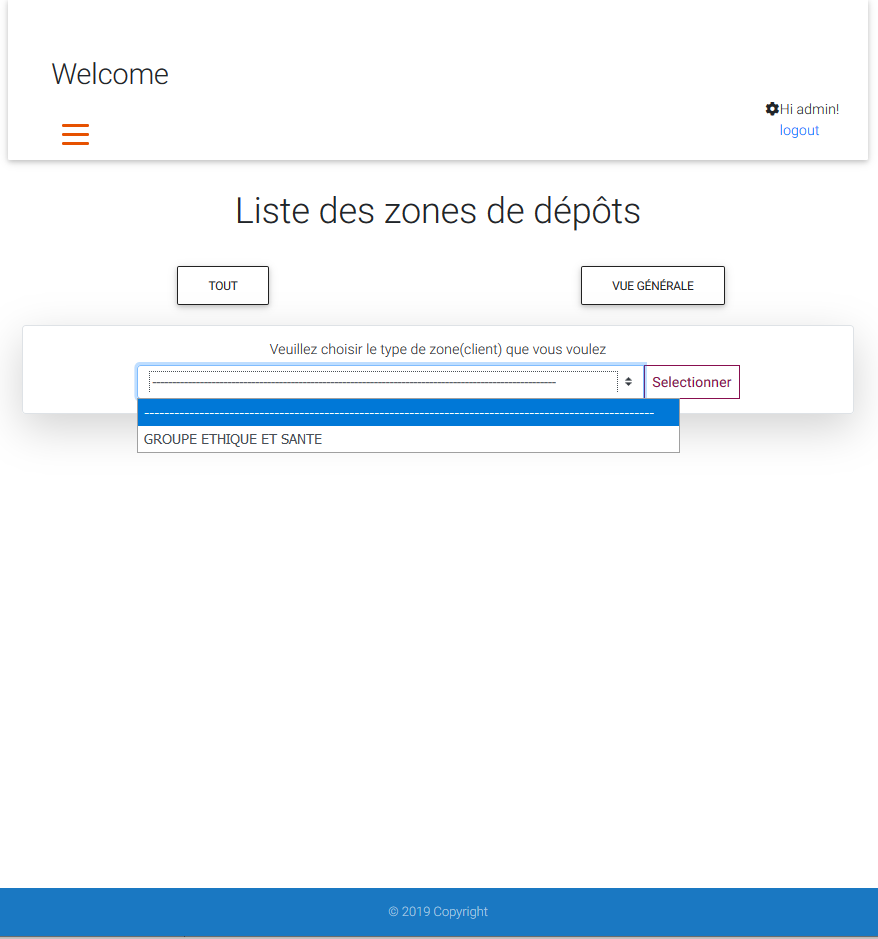
Le compte administrateur est accessible si ajouter par un terminal sur le serveur, via la commande « python manage.py createsuperuser », une fois créé on aura un accès administrateur.

## Crée les types de fournisseurs :



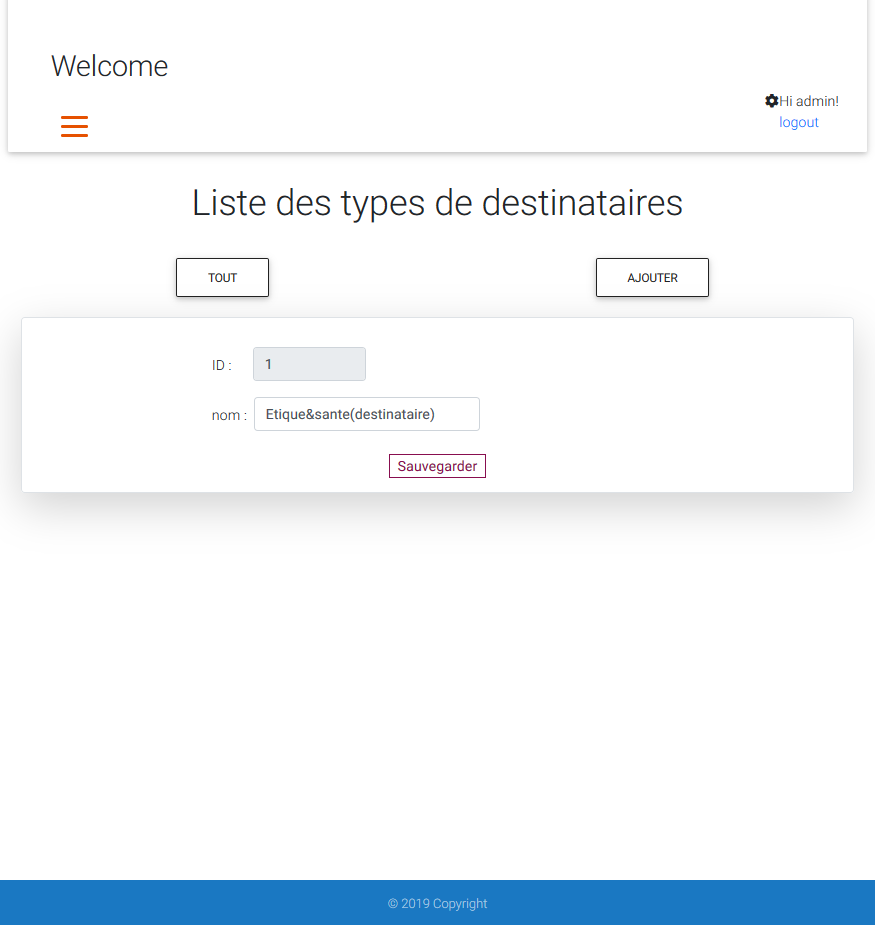
L’identifiant est généré automatiquement, il s’incrémente tant que les ids sont pris.

## Crée des zones de dépôts :



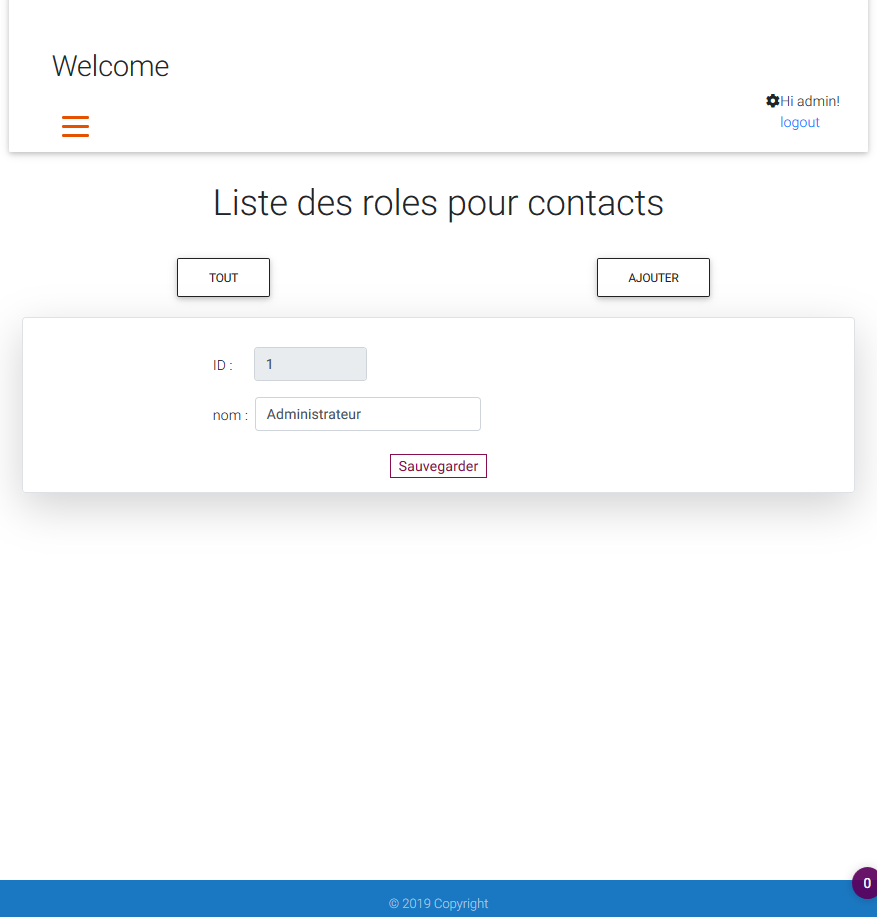
On peut choisir pour quel client on va créer une zone de dépôt, et ça va automatique prendre en compte son « typezone » attribué, on peut aussi créer des zones sans « clients » et on peut modifier toutes les zones via « vue générale » qui montre les zones de tous types (clients).

## Crée les types de destinataires :



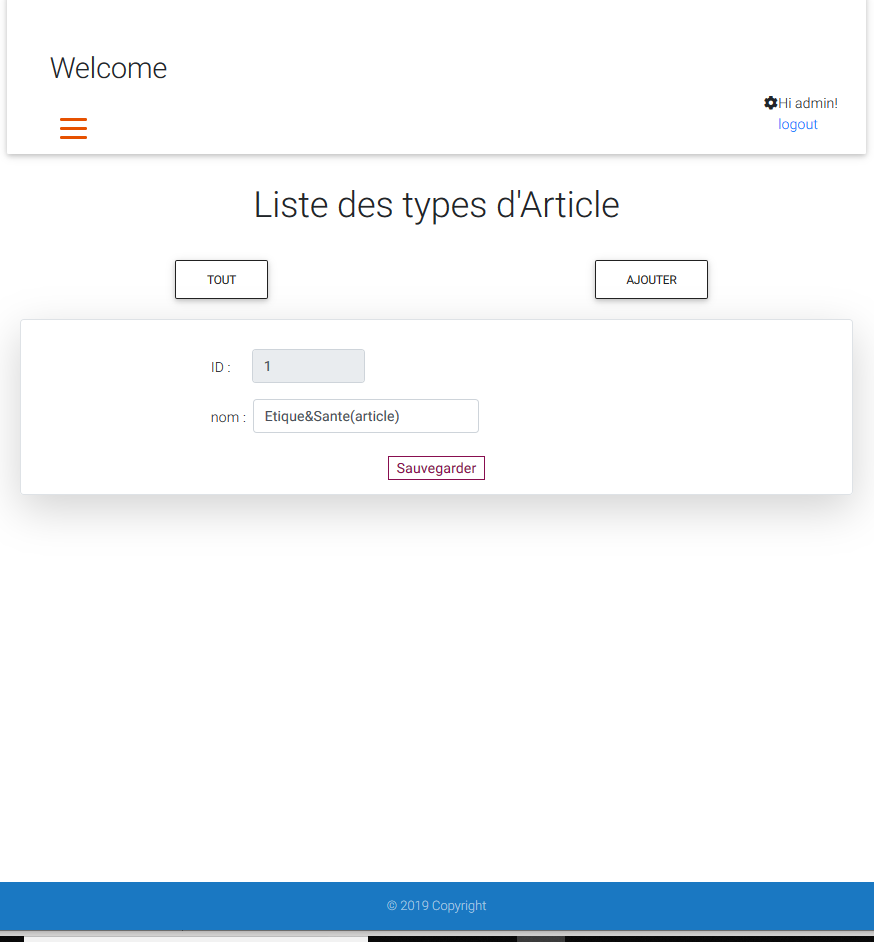
Ici la variable « ID » est aussi automatiquement attribué, le nom que l’on va rentrer va servir à décrire le nom du « type » de destinataire

## Crée les types de contacts :



ID est automatique, le nom attribué servira à « identifier » le « grade » de la personne avec qui on est en contact dans l’interface « Client »

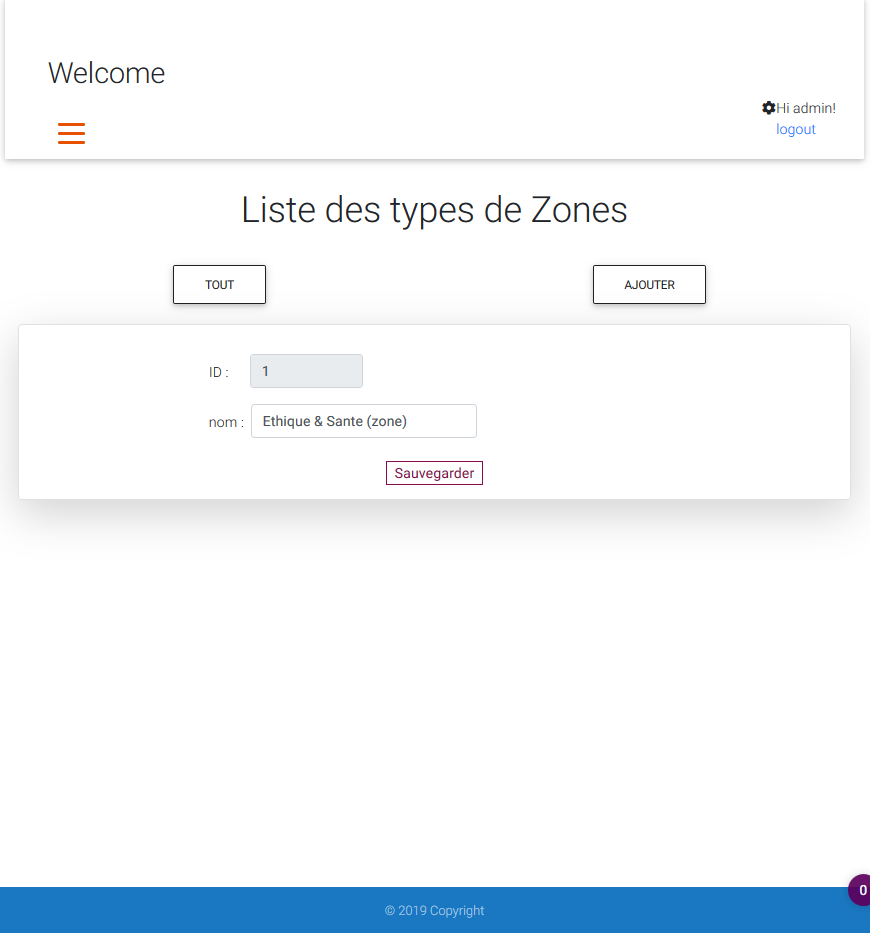
## Crée les types d’articles :



Pareil ici, la variable « ID » est toujours automatique,

Quant au nom il est entré manuellement, il permettra de savoir à quel « type » de client il appartient, chaque client aura ses types d’articles, de zones, etc…

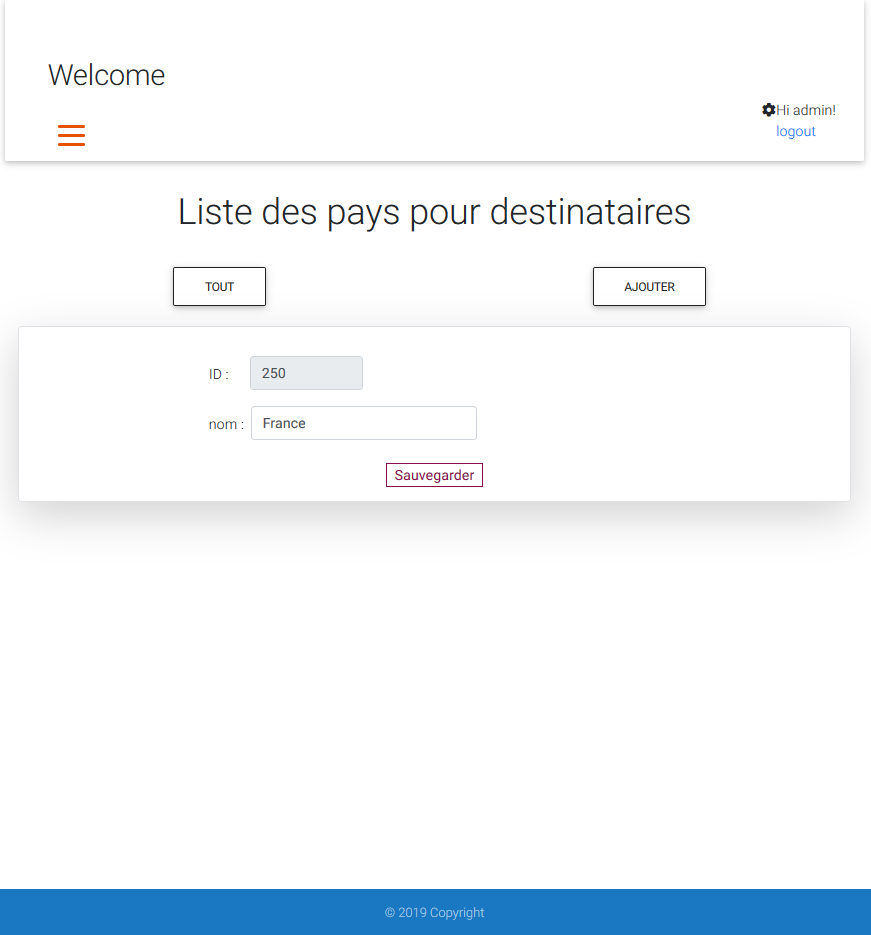
## Crée ou géré les types de zones :



La variable « ID » est automatique, la variable « nom » elle sera rempli manuellement,

Pour pouvoir savoir pour quel client « tel » zone est « faite ».

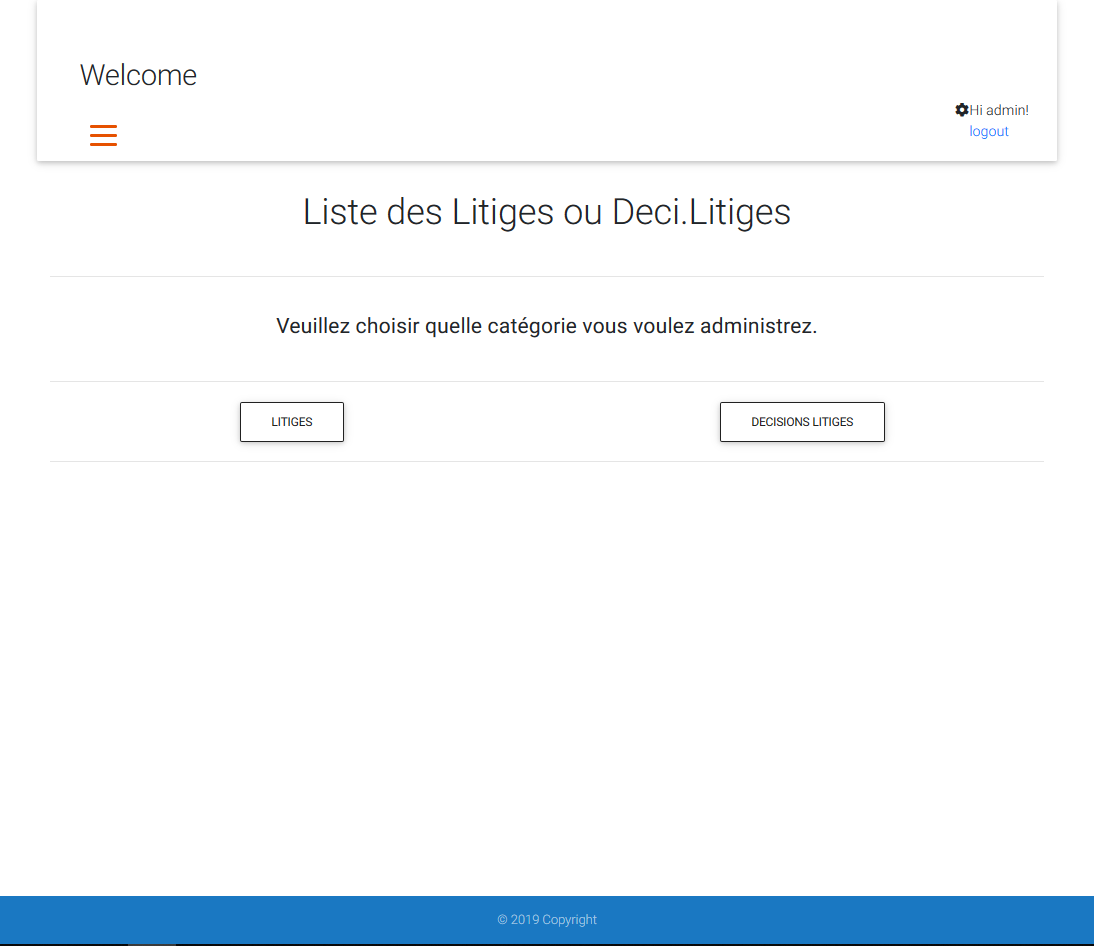
## Crée les pays pour destinataire :



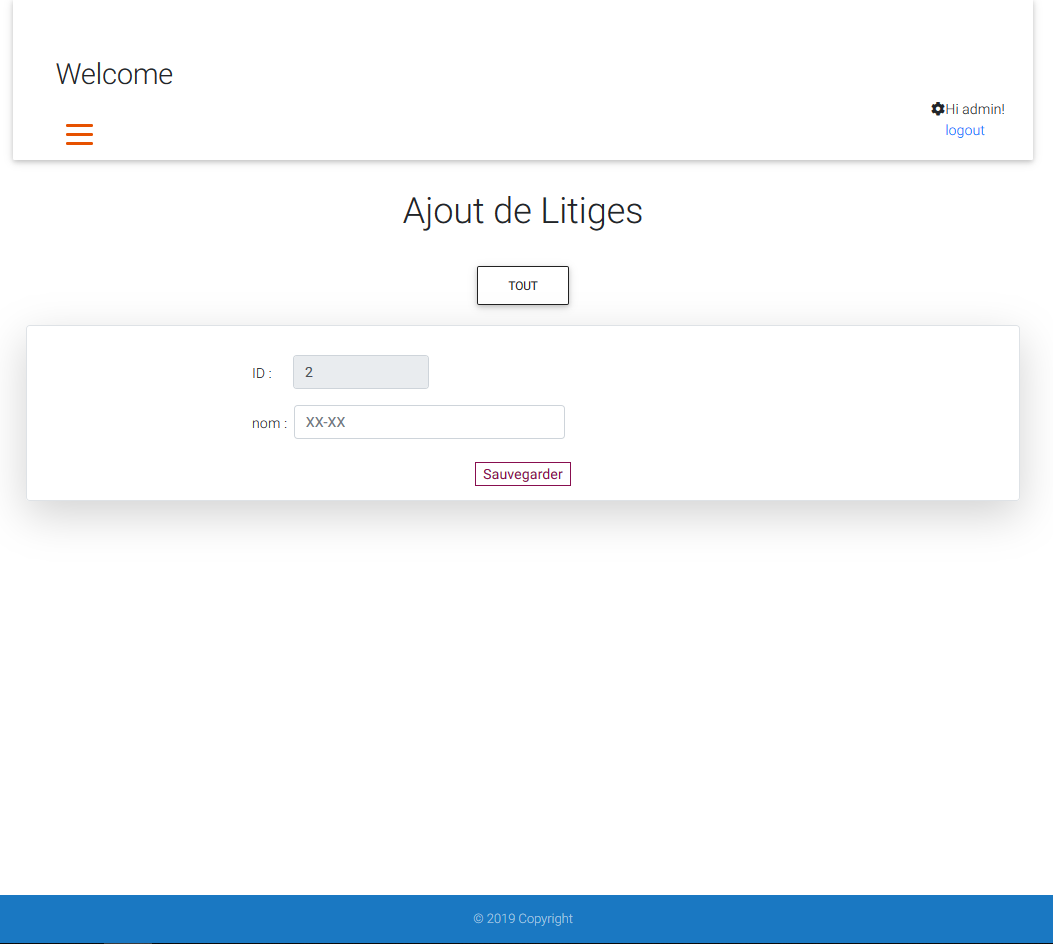
« ID » est automatiquement attribuée,

Le « nom » est à « décrire » manuellement, pour identifier dans quel pays se situe un destinataire.

## Crée ou modifié les litiges et / ou décision litiges :



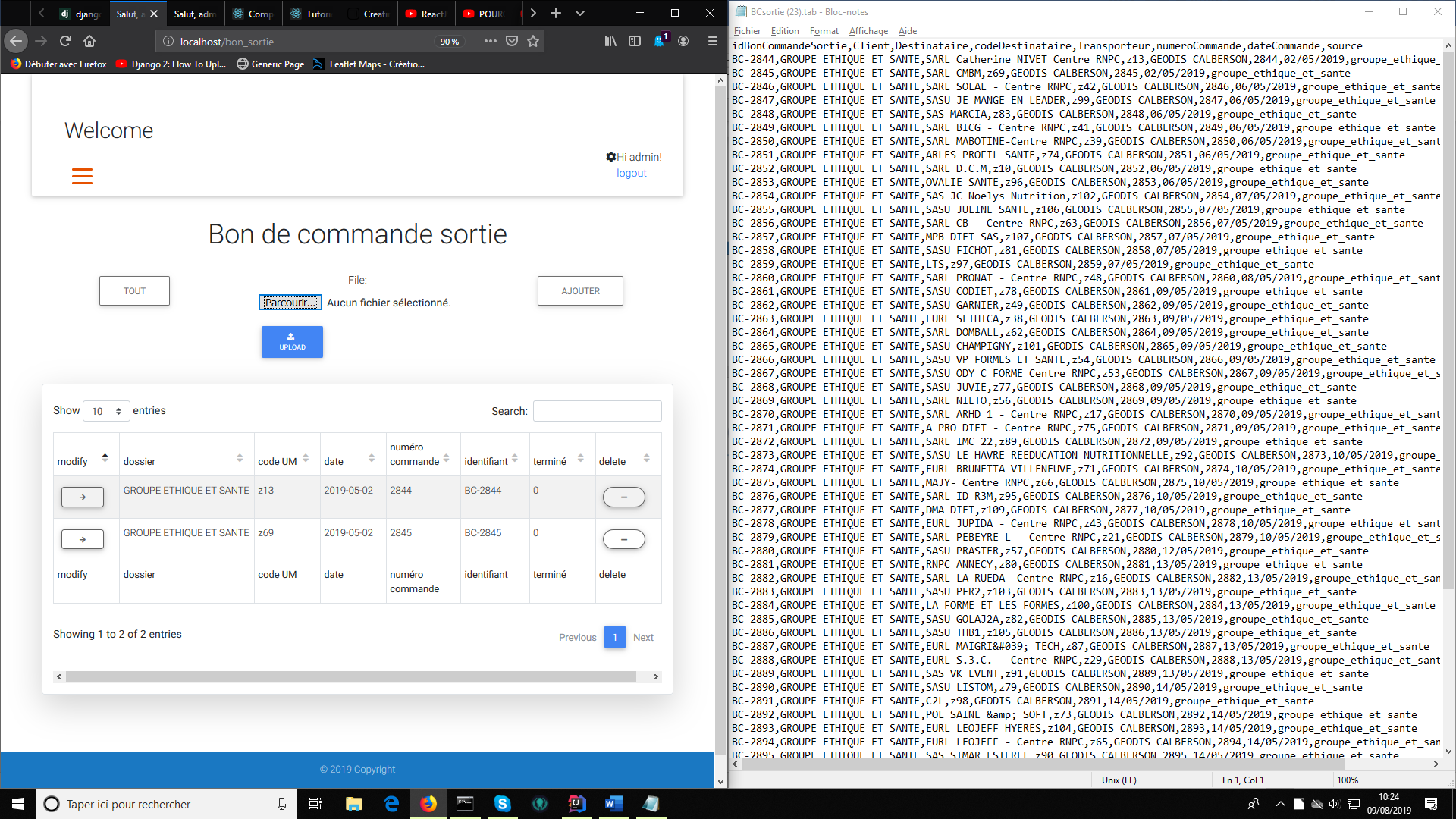
Pour modifier on ajouter les litiges (table « Litige »), on clique sur litige,



Pour les « Décision de litige » on clique sur le bouton adéquat.

# Bon de commande sortie

Gestion des CSV :



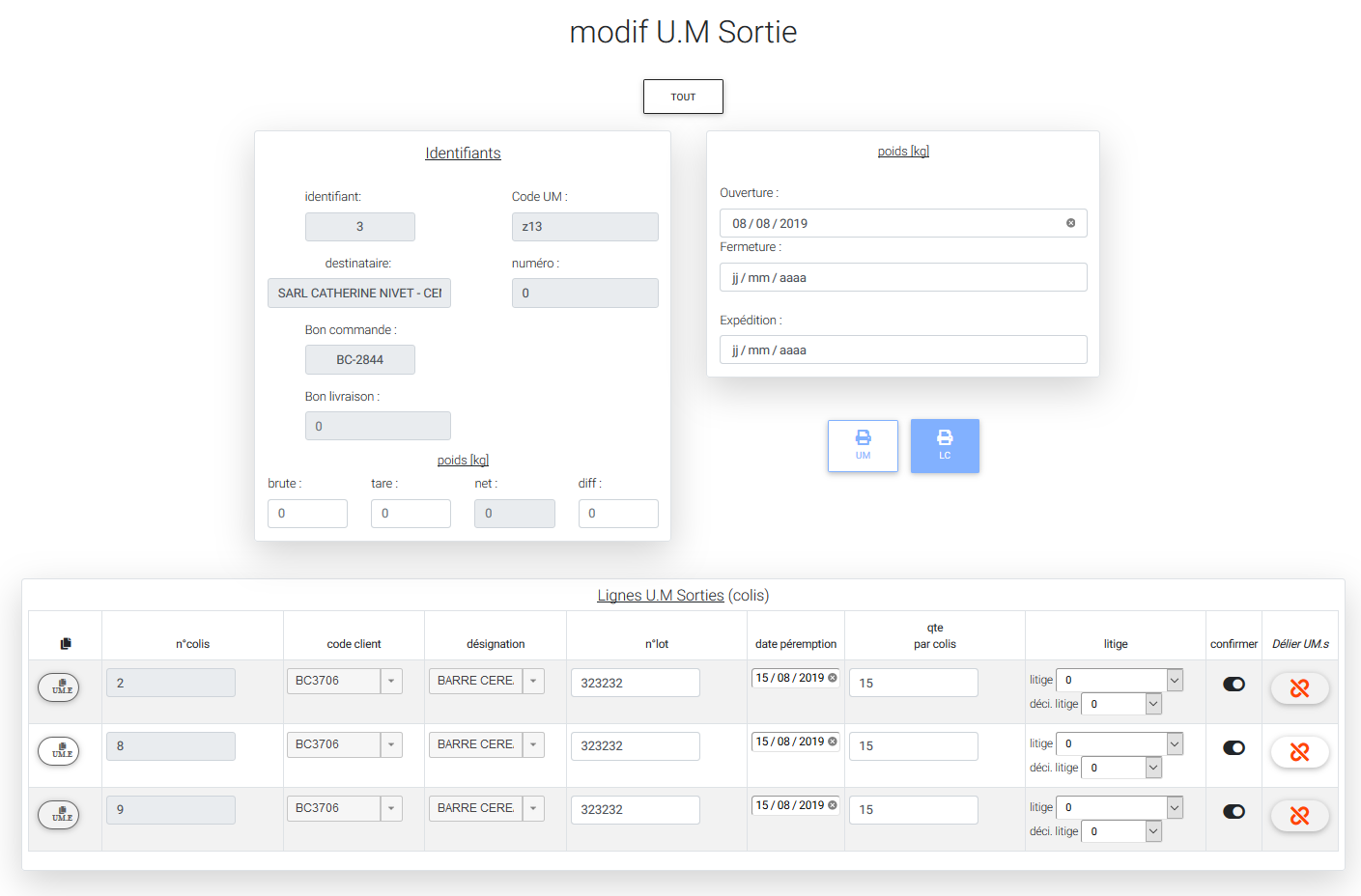
Sur ce template on gère les fichiers importés, de type .tab, avec un format interne en « CSV » soit « Comma separated values », les 8 colonnes du fichier sont vérifiées, si une colonne contient une donnée erronée la création des lignes du bons de commande ou du bon de commande est annulée, toutes les foreign key sont vérifier, si une foreign-key n’existe pas on annule tout, une fois les lignes du bon de commande sauvegarder, on commence à trier les colis en stock de libre,

Pour les placer dans les lignes par ordre de priorité.

Ps : si on supprime un bons de commande ça supprime les lignes & libère les colis pris.

Unité Manutention Sortie :

Les Unité Manutention Sortie sont l’équivalent des Unité Manutention Entrée, dans lesquels on met des « Colis » (Ce sont des palettes),



9

7

4

2

5

8

10

3

6

12

19

18

17

13

14

15

16

1

11

1 –

« Tout » va rediriger vers le template ou on voit les « ums », on pourra supprimer un ums ou aller dans sa page de présentation

2 –

Identifiant de l’unité manutention sortie, elle est attribuée automatiquement à la création de l’objet.

3 –

Le code um ici est « automatiquement » retranscrit, il est originaire du « Bon de commande » qui doit avoir un « Destinataire » en paramètre, qui lui possède la variable « codeUM »

4 –

Variable permettant de savoir quand une palette de sortie (unite manutention sortie) est créée,

Elle est automatique, après chaque algo de « trie » dans l’entrée « dateOuverture » est automatiquement mise à la date ou le script est lancer, elle se situe dans la table « UniteManutentionSortie »

5 –

Zone de texte contenant le « nom » du Destinataire, présent dans la table « UniteManutentionSortie » via « fk\_BonCommandeSortie » (table des bons de commande sortie) qui contiennent un « fk\_Destinataire » qui lui renvoi vers la table du destinataire, ou on a accès au fameux « nom »

6 –

Numéro de la palette venant lors de la création de la palette par un script de trie,

Chaque fois qu’une « UniteManutentionSortie » est crée pour un seul « BonCommandeSortie »,

Elle doit être incrémenter à chaque fois (pas encore le cas ici)

7 –

Cette zone de texte est une donnée générer « manuellement », une fois que l’on juge la palette pleine, on la « ferme », dès que nous avons sélectionner une date manuellement, une fonction « change » va envoyer une requête ajax(post), pour signifier au serveur de remplir la donnée de la « dateFermeture », situé dans la table « UniteManutentionSortie » via l’url « umsortie/modify/datefermeture » elle envoi en paramètre l’id de « UMS » et la valeur de l’input (date) le côté back est géré dans la class « umsmodify » dans la fonction « datefermeture ».

8 –

Zone de texte contenant l’identifiant du « BonCommandeSortie » responsable de cet « UMS »,

Renvoi vers la page du « BonCommandeSortie » responsable de cet « UMS ».

9 –

Cette zone de texte est une donnée générer « automatique » (mais on peut la changer manuellement), quand un « BonCommandeSortie » voit toutes ses commandes « respecter » on doit crée les ou le « BL » qui va mettre la donnée « dateexpedition » au jour « umsortie/modify/dateexpedition » elle envoi en paramètre l’id de « UMS » et la valeur de l’input (date) le côté back est géré dans la class « umsmodify » dans la fonction « dateexpedition ».

10 –

Si une palette de sortie (UMS) est « expédiée » elle sera forcément dans un « BonLivraisonSortie »,

Donc on pourra le voir dans cette zone de texte, elle permet aussi si on « trigger » l’évènement « click » sur cette zone de texte, elle redirige vers le template de ce « BonLivraisonSortie ».

11 –

Ce bouton ouvre dans un nouvel onglet un lien qui redirige vers la page des étiquettes pour la palette, via l’url Django « etiquetteUMs »

12 –

13 –

14 –

15 –

16 –

Bons de livraison sortie :

Les Bons de Livraison Sorties sont les équivalents des Bons de livraison Entrée, leur but est de « synthétisé » ce qui sort en un seul « papier ».



A

Les bons de livraison sortie ne peuvent être créer que via l’interface d’un bon de commande (A), qui remplit les conditions nécessaire (avoir au moins une unité manutention sortie fermée et non expédiée).

# Annexes

## Listes des explications

### Urls :

Les urls dans Django sont le moyen de lié un « lien » et une « vue »,

Via la « vue » on peut faire appel à des fonctions class ou des fonctions dans les « Views.py »

### Views :

C’est la partie ou on exécute du code « Python », avec des fonctions Django on peut call des fonctions Python, ou des méthodes de « Django ».

### Models :

C’est la partie ou on à la « structure » de la base de données qui va être utilisé,

On peu directement mettre des « méthodes » pour avoir des « scripts » dans la partie « models » sans avoir à les faire dans la « vue » ou les « templates ».