

**Mode d’emploi**

RHOULAM Mehdi

06.35.59.11.88

Mehdi.rhoulam@epitech.eu

Promotion 2022

Epitech Marseille

# Contents

[Contents 2](#_Toc16069949)

[Lettre de voiture entrée 3](#_Toc16069950)

[Liste des LVE : 3](#_Toc16069951)

[1 : Redirection 3](#_Toc16069952)

[2 : Supression 3](#_Toc16069953)

[3 : Création 3](#_Toc16069954)

[4 : Données 4](#_Toc16069955)

# Lettre de voiture entrée

## Liste des LVE :

4

2

3

1

## 1 : Redirection

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

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

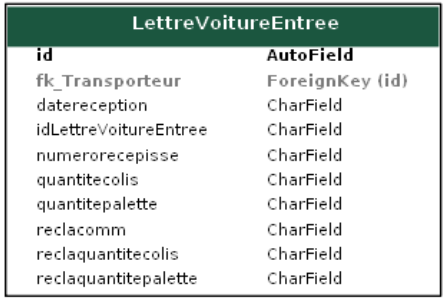
## 3 : Création

Ce button 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 form 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 donnée 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 entree 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 renvoi 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 entree, on utilise la variable « identifiant »

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

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

## 3 -

Redirection vers l’url Django « bonLivraisonEntreemodif » 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 a la fonction « deletebl » se trouvant dans la class « lettrevoitureentreemodify » via l’url html « lve/modify/deletebl/ »

Il va supprimer le bon de livraison entree 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 Entree

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 « entreé » de bon de livraison existe avant celle-la

## B :

Redirige vers le template d’acceuil des bl entree « bonLivraisonentreemodify »

## C :

Redirige vers le template d’acceuil des bl entree « bonLivraisonentreeadd»

## D :

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

Va vérifier si une « entreé » de bon de livraison existe après celle-la

## 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 Entree ».

## 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 soit 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 « Unite Manutention Entree », elle va la lier automatique au « Bon de Livraison Entree » qui va la crée, via l’url « bonLivraisonentree/modify/createume »

On envoi l’id du Bon de livraison entree 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 fichier chargé on clique sur « upload » qui va faire appel a la fonction vers laquel « upload\_blef » renvoi.s

## 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 (unite manutention entree) 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 ».

# 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 la quelle ils travaillent est « complète » ou pas.

## 2 :

Input d’affichage de la foreign\_key Bon Livraison Entree, elle permet de savoir quel « Bons de Livraison Entree » a créé la palette « actuelle », le bouton lui renvoie vers la page du « bons de livraison entree » 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 si il y a beaucoup de zone, il va faire une découpe de tout ce qui ne ressemble pas a 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 au donnée 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 article 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 :

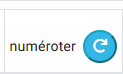
Ceci est le numéro (id) du Colis sur lequel on travail, ne pas se fier à cette valeur tant que l’on a pas sauvegarder le colis (cette valeur n’est que spéculative, elle peut être ensuite modifier par le serveur si il est déjà prise), se fier la numérotation dans du colis qui elle est forcément unique 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 redirigé 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 numero de l’Unite Manutention Entree ! = 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

## 17 :

Ceci est le code UM du destinataire venant du bon de commande sortie, via l’Unite 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’Unite 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 back-end 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 peu 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 htmp « /sort\_cols/ » il prend en paramètre l’id de l’unité manutention entree, et puis coté serveur il y a un algorithme de tri.

# Paramètre de L’API

# 

## Gestion graphique :

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 a droite avec une flèche perme de supprimer les images de la bdd en même temps.

## Crée ou géré les utilisateurs :

## Crée les types de fournisseurs :

## Crée ou géré les types de zones :

## Crée les types de destinataires :

## Crée les types de contacts :

## Crée les types d’articles :

## Crée des zones de dépôts :

## Crée les pays pour destinataire :

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