

onboarding FM : projet CHL

*Bienvenue chez Fioulmarket. Vous avez passé les étapes de recrutement avec brio. Il faut maintenant se mettre dans le bain (ou dans le fioul) en se faisant la main sur un petit projet métier afin d'assimiler une des briques de Fioulmarket. Le thème de ce mini projet est le **CHL** (Créneau Horaire de Livraison).*

Le but ici est de permettre à un client de commander du Fioul.

Partons du principe que ce **client habite à Rueil-Malmaison** (92500). Fioulmarket travaille avec des filiales de livraison. Dans le cas présent, une filiale nous fournit ses capacités de livraison pour le mois de février 2020.

Il va falloir se baser sur les deux fichiers JSON fournis par la filiale de livraison pour générer le calendrier de livraison :

- **creneaux.json**: liste des créneaux pour Rueil-Malmaison et leur camion dédié.
- **camions.json**: liste des camions et de leurs volumes/créneaux pour le département du 92.

Spécifications techniques de votre site

- Langage de programmation : **PHP**
- Framework PHP : **Symfony 2.8**
- Partir front : **Libre (frameworks JS libres ou même vanilla JS)**
- Base de données : **MariaDB**

Installer le docker suivant :

<https://github.com/FioulMarket/docker>

C'est à vous de choisir si vous sauvegarder les créneaux en BDD ou si vous désirez vous embêter faire vos calculs depuis les fichiers JSON.

Il est recommandé de mettre votre environnement de dev dans DOCKER. (Exemple avec des containers php-fpm, nginx, phpmyadmin).

Spécifications métier

Votre site est composé de **2 pages** :

- page de commande du fioul (route = /commande-fioul)
- page de confirmation (route = /confirmation-commande-fioul)

la page /commande-fioul :

- la page est composée de 3 éléments principaux:
 - un champ d'entier permettant de choisir un **volume** en litres (entre 500L et 10000L).
 - le **calendrier** de livraison
 - un **bouton** "passer ma commande"
- Quand la valeur du champ **volume** est **mise à jour** :
 - Le **calendrier** se **régénère** de manière **asynchrone**
- Quand un utilisateur **valide le formulaire, sauvegarder** celle-ci en BDD :
 - **numéro de commande** (ce n'est pas l'ID technique, générer des références de commande de manière logique)
 - **date de la commande**
 - **date de la livraison**
 - **volume** commandé
- Génération du calendrier :
 - La **sélection** du créneau se fait à la **journée**
 - le contenu des fichiers **créneaux.json** et **camions.json** doit être pris en compte
 - les dates affichées dans le calendrier généré doivent être les **6 prochains jours ouvrés à partir de J+2 (ouvrés)** la date de la commande (ne pas prendre en compte les samedi/dimanche ainsi que les jours fériés).
 - les **commandes en base de données** doivent également être **prises en compte lors du calcul** pour la génération des créneaux

la page /confirmation-commande :

- **Récapitulatif du volume et du créneau**
- Un **lien** vers /commande-fioul pour re-passer commande

Exemples et cas d'usages

Cas Créneaux 1 :

Si je passe commande le Lundi 10 février, je dois voir les créneaux suivants :

- mercredi 12 février
- jeudi 13 février
- vendredi 14 février
- lundi 17 février
- mardi 18 février
- mercredi 19 février

Cas Créneaux 2 :

Si je passe commande le jeudi 13 février:

- mardi 18 février
- mercredi 14 février
-

Cas Créneaux 3 :

Je désire passer une commande de 2000L et être livré le 13 février. Dans le fichier JSON de la filiale j'ai 2400L de capacité pour ce jour. Or une commande de 1000L est déjà présente en BDD. Le créneau du 13 février doit donc être fermé pour les commandes de 1400L ou plus.

PARTIE FACULTATIVE MAIS RECOMMANDÉE (UTILISATEUR)

Ajouter 3 pages :

- page d'inscription (route = /inscription).
- page de connexion (route = /connexion)
- page de déconnexion (route = /logout)
- **La page d'inscription** est composée d'un formulaire email / mot de passe ainsi que d'un lien vers /connexion
 - Une fois le formulaire validé, il faut sauvegarder l'utilisateur en base de données.
- **La page de connexion** est composée d'un formulaire email / mot de passe ainsi que d'un lien vers /inscription
 - Une fois l'utilisateur connecté, le rediriger sur /commande-fioul.
 - Le header du site doit comprendre le texte "connecté en tant que [email de l'utilisateur]" ainsi que d'un lien sur /logout.

Si l'utilisateur tente d'accéder à n'importe quelle page (autre que les pages de connexion / inscription) sans être identifié, il doit être rediriger sur /connexion.

Si l'utilisateur tente d'accéder à /inscription ou /connexion sans être connecté, le rediriger sur /commande-fioul.

Quand un utilisateur passe commande, rattacher ce dernier à la commande en base de données.