

Génération d'instances pour le prototype

Données nécessaires pour la résolution:

- Un tableur .csv pour les établissements (cantines):
 - Les colonnes suivantes (et leurs informations) sont requises : « Nom de la structure », « Commune », « Domaine », « Nombre de repas par jour », « Type de restauration », « x » (coordonnées géographiques), « y » (coordonnées géographiques)
 - Les colonnes « x », « y » peuvent être générés, nécessite une colonne « Adresse »
 - Récupération depuis l'annuaire de la restauration collective possible :
 - Récupérez le .csv de l'annuaire, effacez les autres colonnes,
 - Les colonnes « x », « y » n'y sont pas,
 - S'assurer que « Nombre de repas par jour » ne contient pas d'espaces.
- Un tableur .csv pour les fournisseurs (producteurs):
 - Les colonnes suivantes (et leurs informations) sont requises : « Nom », « Commune », « x » (coordonnées géographiques), « y » (coordonnées géographiques), « Filières »
 - Les colonnes « x », « y » peuvent être générés, nécessite une colonne « Adresse »,
 - La colonne « Filières » peut être généré, mais il est fortement conseillé de remplir soi même. Pour indiquer la/les filières de produits que proposent un fournisseur, respecter le format : « [*Une filière*] » ou « ['Une filière', 'Une autre filière'] » avec toujours un « , » entre chaque filière et des guillemets « ' » encadrant la dénomination .
 - Récupération depuis l'annuaire de la restauration collective possible :
 - Récupérez le .csv de l'annuaire, effacez les autres colonnes, à l'exception de la colonne « Adresse »
 - Les colonnes « x », « y », « Filières » n'y sont pas.
- Un tableur .csv pour les plateformes :
 - Les colonnes suivantes (et leurs informations) sont requises : « Nom de la structure », « Commune », « x », « y », « Cout d'ouverture »
 - Les colonnes « x », « y » peuvent être générés, nécessite une colonne « Adresse »,
 - La colonne « Cout d'ouverture » peut être générée.
- Un tableur .csv pour les commandes :
 - Les colonnes suivantes (et leurs informations) sont requises :
 - une colonne « E », correspondant à « Nom de la structure » d'un établissement dans le tableur concerné,
 - une colonne « P », correspondant à « Nom » d'un producteur dans le tableur concerné,
 - une colonne « F », correspondant à une filière de produits,
 - une colonne « d », correspondant à la quantité des produits de filières « F » commandés.
 - Toutes les entrées peuvent être générées, voir détail page suivante.

A propos de la génération de colonnes et de leur contenu :

- « x », « y » pour les différents tableurs:
 - Génération effectuée par l'usage d'un script effectuant des requêtes, nécessite une connexion Internet. Le script renvoie les adresses qu'il a essayé de rechercher sans succès.
 - S'assurer que le contenu de la colonne « Adresse » et « Commune » forment ensemble une adresse complète. Dans le cadre de l'annuaire de la restauration collective, la colonne « Adresse » est parfois donnée en entière par exemple « Adresse » = « 32 rue des Roses Donges 44480 » et « Commune » = « Donges ». L'adresse recherchée par le script devient « 32 rue des Roses Donges 44480 Donges » et risque fortement d'échouer. Aussi, la colonne « Adresse » comporte parfois juste le nom de la rue, il est nécessaire d'avoir un numéro « rue des Roses » → « 32 rue des Roses ».
 - Une fois que le script a terminé son exécution, un fichier du même nom avec « _map.html » apparaît dans le dossier. Il est fortement conseillé de vérifier la disposition des localisations sur la carte générée pour s'assurer de l'exactitude de la génération.
 - Dans le cadre où il y a une erreur de génération et/ou que vous souhaitez rentrer manuellement les informations, il vous est possible de le faire avec Google Maps :
 - Une fois sur Google Maps, trouvez la localisation en question, ici la « Ferme du Bois Joubert » (image 1),
 - Faites un clic droit exactement sur la localisation, un menu apparaît avec les coordonnées et d'autres options. Un clic gauche sur les coordonnées permet de les copier directement (image 2),
 - Dans le tableur, dans l'entrée correspondant, le premier nombre correspond à « x » et le second correspond à « y ».

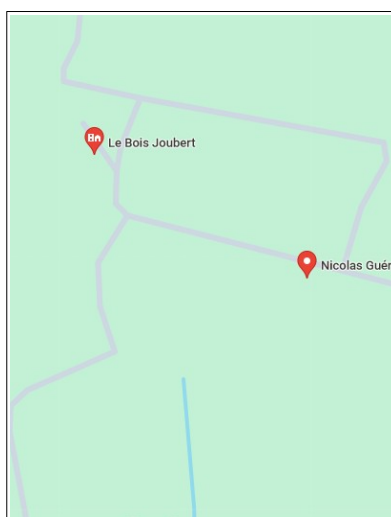


Image 1



Image 2

- **« Filières » pour les producteurs :**

- Il est recommandé d'observer les données présentes pour chaque producteurs et leur identifier manuellement les filières de produits qu'ils proposent. Par manque de formalisme dans les données de l'annuaire de la restauration collective, il n'était pas possible de les déterminer efficacement en lecture.
- Sinon , un script permet de générer l'affectation des producteurs aux filières. Il prend deux paramètres :
 - « *proba_mult_f* » pour indiquer la probabilité qu'un producteur soit affilié à plus d'une filière,
 - « *ratio_f* » pour indiquer la répartition des affectations des producteurs. Ex : pour nos 4 filières de produits → [0.4,0.4,0.1,0.1] indique 40 % de chance que le producteur soit affecté à la première filière, la seconde, et 10 % pour la troisième et pour la quatrième. La somme doit être égal à 1 (100%).

(Ce script est très approximatif, et ne prend pas aucune information de l'annuaire de la restauration collective, d'où la recommandation d'effectuer une saisie manuelle des affectations).

- **Coût d'ouverture d'une plateforme**

- Un script permet de générer des coûts pour toutes les plateformes. Ce dernier prend trois paramètres :
 - « *Prix_mètre_carré* » pour indiquer le prix d'un mètre carré en moyenne,
 - « *Surface* » pour indiquer le nombre de mètre carré requis pour la plateforme au minimum,
 - « *Ecart* » pour indiquer un pourcentage faisant varier le prix total pour une plateforme définie par les deux précédents paramètres,
 - « *ratio* » pour indiquer un ratio d'amortissement pour mettre le coût total à une échelle permettant de comparer les coûts d'ouverture aux coûts d'acheminements.
- Nous avons utilisé en valeur par défaut un prix de m² de 1000 €, une surface minimum de 100 m², un écart de 30 % et un ratio de 1/52 semaines. Le prix et l'écart est défini par l'étude du cas de Local'Planet, où le coût d'aménagement des locaux de 680 €/m² est considéré comme très contenu. Nous le comparons alors au 1000 €/m² cité, et nous récupérons son écart (arrondi).

- **Toutes les informations du tableur « Commandes »,**

- Pour pouvoir générer des commandes, il est nécessaire d'en faire une autre auparavant, celle qui va déterminer l'ensemble des quantités nécessaires par cantine.
 - Cette génération nécessite deux fichiers spéciaux (dans « in/data ») appelées « *demandFiller.json* » et « *demandFillerRatio.json* ». Ces fichiers « traduisent » les

données du Réseau Manger Local, sur les quantités de produits nécessaires à l'année :

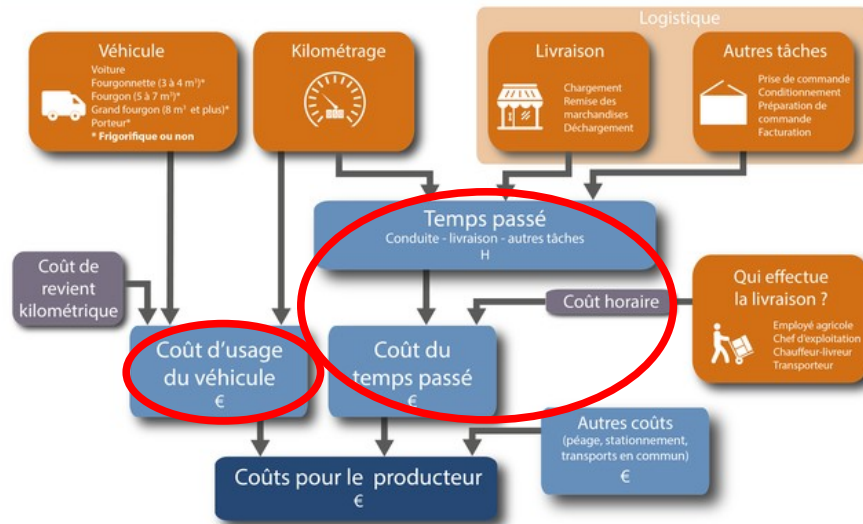
- Pour chaque entrée dans « *demandFiller.json* », les informations sont présentées de la manière suivante : « Filières de produits » → « Produits » → « Quantité requise à l'année pour un établissement servant 100 repas par jour »
 - Pour chaque entrée dans « *demandFillerRatio.json* », les informations sont présentées de la manière suivante : « Filières de produits » → « Produits » → « Ratio transformant la quantité dans « *demandFiller.json* » en une autre quantité »
 - Le résultat est exprimé comme étant « Quantité de produit à l'année pour 100 repas par jour »
- Le script va ensuite récupérer les informations de la colonne « Nombre de repas par jour » dans le tableur des établissements, puis générer toutes les quantités requises de produits.
 - Un paramètre « *mult* » est un ratio multipliant « Quantité de produit à l'année pour 100 repas par jour ». Il permet notamment de ramener ces quantités à une échelle.

Par exemple en divisant les quantités par 36 semaines ouvrables, nous avons la quantité requise par un établissement à la semaine. Par la suite, multiplier par un pourcentage nous permet de quantifier plus spécifiquement les commandes, exemple : $mult = 1/36 * 0.5 \rightarrow 50\%$ de la quantité requise pour un établissement par semaine.

- Le script va enregistrer ces résultats dans un tableur similaire avec des colonnes supplémentaires. Ces colonnes correspondent aux quantités de chaque produit, avec une colonne « filière de produit » qui est la somme des quantités de tous les produits de la filière.
- Une fois cette génération terminée, il est possible de passer à la génération des commandes. Le script va récupérer les informations générées lors de la génération précédente, celles qui donnent l'affectation des producteurs à une/des filières de produits et va construire des commandes à partir de celle là.
 - Un paramètre est utilisé, « *ratios_pc* » qui est un ensemble de valeurs où il y a une valeur par filières de produits. Ce paramètre indique que pour la filière de produits correspondante, les établissements vont se fournir chez autant de producteurs en question.
 - Pour chaque producteurs qui est désigné pour une filière de produits, nous vérifions s'il propose des produits de filières qui ne sont toujours pas commandés, et si oui, passe automatiquement une commande pour ces autres filières.
 - La génération se termine par l'enregistrement d'un tableur indiquant uniquement les valeurs représentant les établissements, producteurs dans leurs tableurs, de même pour les filières.

Données exclusivement générées : Coût des transports

- Dans le cadre des coûts de transports, nous admettons qu'ils ne peuvent uniquement qu'être générées. Pour cela, nous utilisons le logiciel *OpenRouteServices*, qui à l'image de *Logicoût*, propose un calcul des distances et du temps requis pour un itinéraire indiqué. Nous imitons partiellement la formule utilisée par *Logicoût* pour déterminer le coût financier du trajet :



Représentation de la formule de calcul du coût pour le producteur

Nous nous concentrons uniquement sur le coût du temps passé, définie par un coût horaire et légèrement sur le coût d'usage du véhicule, que nous définissons comme étant la consommation de carburant. Cette consommation est calculé en divisant la distance du trajet par la consommation moyenne en litre pour 100km d'un véhicule, le tout multiplié par un coût du carburant au litre.

- La génération des coûts de transports impliquent d'avoir pour chaque acteur (plateforme/cantine/producteur/transformateur) leurs coordonnées géographiques. Il est donc obligatoire de générer « x » et « y »
- Nous utilisons donc 3 paramètres :
 - « Prix » pour le prix du carburant au litre,
 - « Consommation » indiquant la consommation moyenne en litre pour 100 km du véhicule,
 - « Coût horaire » pour le salaire de l'employé effectuant les trajets.
- Par défaut « Prix » correspond au prix de l'essence (à un moment donné), « Consommation » est tirée du site de l'entreprise ECOLOW, vendeur d'économisateurs de carburant, et « Coût horaire » représente le SMIC horaire HT.
- Pour un trajet, le coût total est donc égal à :

$$\text{coût total} = \text{durée trajet} * \text{coût horaire} + \text{distance trajet} * \text{consommation} * \text{prix}$$
- Contrairement à *Logicoût*, les coûts « environnementaux » ne sont pas calculés.