



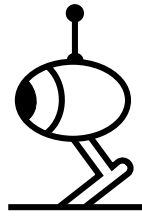
## Transports & données ouvertes (Notes)

Les données ouvertes promettent beaucoup : transparence, réutilisation, innovation. Dans le domaine des transports, elles sont abondantes, standardisées (GTFS, CSV...) et d'un accès aisé.

En théorie.

Dans la pratique, transformer ces jeux de données en une information réellement utilisable, comme afficher les transports disponibles autour d'un point géographique, soulève de nombreuses difficultés.

Dans ce retour d'expérience, je raconte comment j'ai construit un système exploitant plusieurs sources de données ouvertes sur les transports.



**FRÉDÉRIC BISSON**

Développeur web à Rouen

Amateur de vieilleries



## Qui suis-je ? (Notes)

Je me présente : Frédéric BISSON.

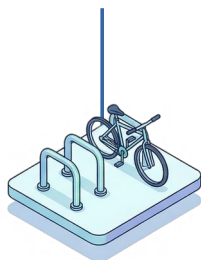
Je suis développeur web à la Ville de Rouen.

Je suis aussi amateur de vieilleries : Minitel, micro-informatique 8 bits, imprimerie à l'ancienne etc.

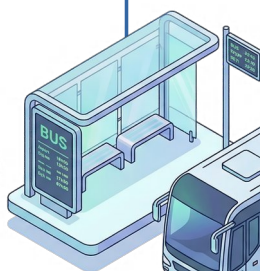
Je m'occupe du site web de la ville de Rouen. Sur ce site, nous avons beaucoup de pages sur lesquelles nous donnons des informations aux usagers afin qu'ils puissent rejoindre un lieu donné.

## INDIQUER LA PROXIMITÉ DE...

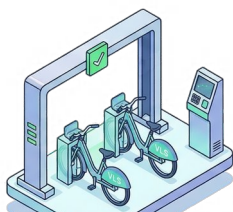
Arceaux vélo



Arrêts de bus



Vélos en libre-service



Lignes de bus/car



## Indiquer la proximité de... (Notes)

L'idée est de présenter des informations comme la présence d'arceaux vélo, de vélos en libre-service (les Lovélo), d'arrêts de bus et des lignes de bus les desservant.

À Rouen, c'est l'agglomération (la Métropole Rouen Normandie) qui gère principalement ces services. Pour les arrêts et lignes de bus, elle a mis en place le Réseau Astuce. Mais elle n'est pas la seule : la Région Normandie dispose du réseau Atoumod et des compagnies comme FlixBus et BlaBlaCarBus font passer des lignes de cars par Rouen.

Ce sont des informations qui ont traditionnellement été renseignées à la main, posant bien sûr des problèmes en ce qui concerne leur mise à jour.

## BUT

### ACCÈS TRANSPORTS

À moins de 300 mètres :

- Lignes 35, 529, 26, F4, T2, T3, T1 (arrêt Mont-Riboudet)
- Lignes 511, 530, 526 (arrêt Rouen - Mont Riboudet - Kindarena)



# 4

Données ouvertes et Drupal 11  
Transformer des données brutes  
en infos utiles

## But (Notes)

Le but du projet est d'automatiser la génération de ces blocs d'information pour que le contributeur n'ait plus qu'à préciser où se trouve le lieu sur une carte.

En voici un exemple. Ce sont des informations avant tout textuelles afin de satisfaire nos obligations en matière d'accessibilité, même si la production d'une carte interactive reste parfaitement envisageable en complément.

## BÉNÉFICES ATTENDUS

- Mises à jour globales
- Homogénéité des informations affichées
- Exhaustivité des facilités
- Facilité d'insertion des informations



**Données ouvertes et Drupal 11**  
Transformer des données brutes  
en infos utiles

**5**

## Bénéfices attendus (Notes)

Nous attendons quelques bénéfices de cette automatisation. L'intérêt est, tout d'abord, de permettre la mise à jour globale des informations, de ne pas être contraint de visiter chaque page pour vérifier à la main.

Cela va aussi nous permettre d'homogénéiser la présentation des informations fournies, d'être exhaustif.

On veut enfin ne pas être bloqué sur la façon dont on insère les informations sur une page, qu'on ait de la souplesse.

# OBLIGATION DE PUBLICATION

- Règlement européen 2017/1926
  - Mise à disposition des données de mobilités via un PAN
  - Dans des formats normalisés ou interopérables
- Loi LOM (2019)
  - Précise les acteurs soumis à l'obligation
  - Étend l'ouverture à d'autres catégories de données
- Décret n°2020-183
  - [transport.data.gouv.fr](https://transport.data.gouv.fr) est le PAN



**Données ouvertes et Drupal 11**  
Transformer des données brutes  
en infos utiles

6

## Obligation de publication (Notes)

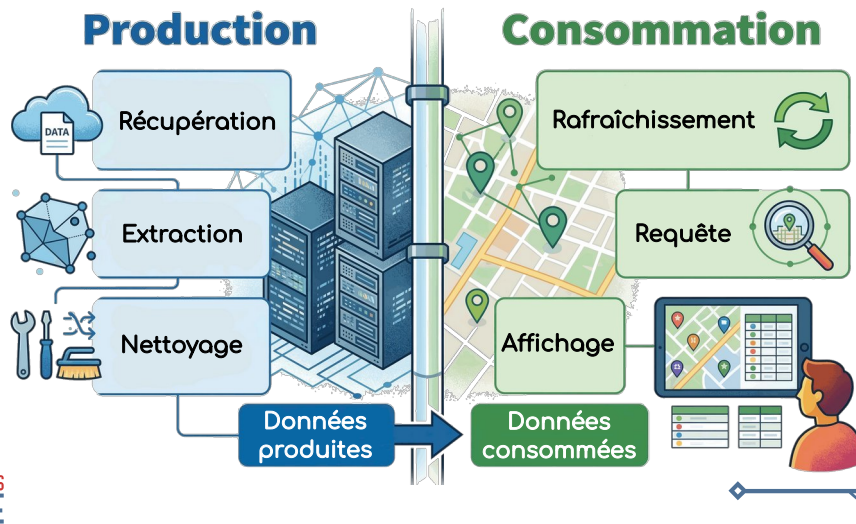
Pourquoi peut-on automatiser la fourniture de ce type d'information? Parce que sont des données ouvertes disponibles en ligne! Les acteurs des transports ont l'obligation légale de les publier.

C'est une conséquence du règlement 2017/1926 de l'Union Européenne qui demande la mise en place de PAN par les états et la mise à disposition des données dans des formats normalisés ou interopérables.

Elle a été suivie en France de la loi LOM en 2019 qui précise les acteurs soumis à l'obligation et étend l'ouverture à d'autres catégories de données comme les données dynamiques, les points de recharges, les parkings de covoiturage etc.

Cette loi a été elle-même suivie du décret 183 de 2020 qui désigne le site [transport.data.gouv.fr](https://transport.data.gouv.fr) comme point d'accès national (ou PAN) unique.

# DU PRODUCTEUR AU CONSOMMATEUR



**Données ouvertes et Drupal 11**  
Transformer des données brutes  
en infos utiles

7

## Du producteur au consommateur (Notes)

Pour pouvoir exploiter ces données de façon automatisée, le projet a été scindé en deux parties : la production et la consommation.

La production des données va consister à récupérer, extraire, nettoyer et produire un paquet de données prêt à être exploité. La production est conçue pour avoir lieu en dehors de Drupal pour des raisons évoquées plus loin.

Côté consommation, la tâche va consister à rafraîchir, requêter et afficher les données afin de fournir les informations utiles à l'utilisateur final. La consommation se passe entièrement sous Drupal version 11, la version sur laquelle nous travaillons actuellement.

## ÉTAPE 1

# PRODUCTION DES DONNÉES



## Production des données (Notes)

Commençons par la production des données, hors Drupal comme cela vient d'être dit.

# FORMATS DE DONNÉES

- **GTFS**  
General Transit Feed Specification  
Fichiers CSV dans un ZIP (et un peu de JSON)
- **GBFS**  
General Bikeshare Feed Specification  
Fichier JSON
- **CSV**  
Comma Separated Value  
Pour les arceaux vélo



**Données ouvertes et Drupal 11**  
Transformer des données brutes  
en infos utiles

9

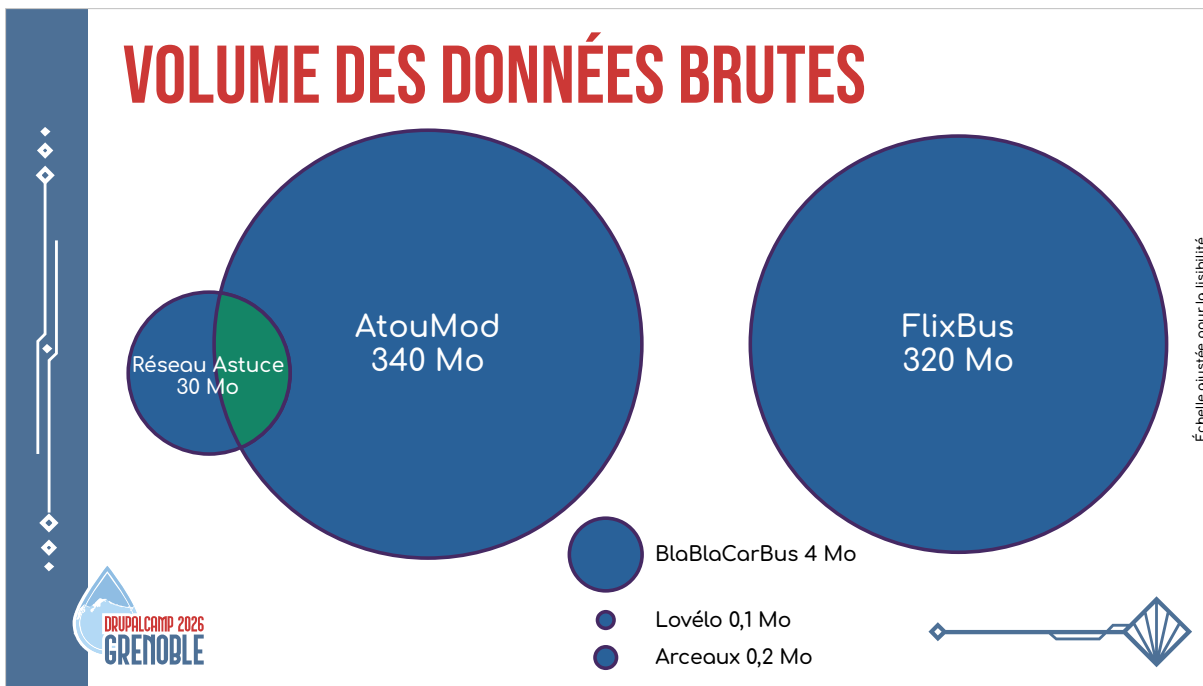
## Formats de données (Notes)

Le règlement européen privilégie NeTEx et SIRI pour la diffusion des informations. On trouve cependant très facilement les données au format GTFS sur [transport.data.gouv.fr](http://transport.data.gouv.fr). Ils sont plus simples à produire et sont reconnus par des plateformes comme Google Maps.

Le GTFS est un format de données dédié aux horaires des transports et aux trajets effectués. Les données sont livrées sous forme d'un fichier ZIP contenant des fichiers CSV standardisés et un peu de JSON. Quand on parle de transport, cela inclut même les... ascenseurs! (qu'il faudra penser à retirer)

On a ensuite le GBFS, un format de données dédié aux mobilités en libre-service, comme les vélos en libre-services. C'est un fichier JSON standardisé.

Pour les arceaux vélo, étant donné qu'il n'y a pas de complexité pour leur traitement, ils sont généralement recensés dans un fichier CSV dont le contenu reste à la discrétion du producteur.



## Volume des données brutes (Notes)

Dans le cas de la ville de Rouen, on a devoir croiser plusieurs sources.

Il y a tout d'abord le « Réseau Astuce ». Ce sont les transports en commun de l'agglomération rouennaise. Les données GTFS représentent environ 30 Mo (5 Mo zippés).

Il y a ensuite les données d'AtouMod, une structure qui englobe les transports en commun de la région Normandie, pour environ 340 Mo (46 Mo zippés). Une partie des données d'AtouMod sont communes à celles du Réseau Astuce, ce qui va nécessiter du filtrage.

On a également les données de FlixBus pour 320 Mo (55 Mo zippés) ou de BlaBlaCarBus pour 4 Mo (400 Ko zippés) car ces opérateurs ont des lignes passant par Rouen.

Reste enfin les données des vélos en libre-service, Lovélo, pour 100 Ko, et celles des arceaux vélos pour 200 Ko. Ces deux fichiers sont fournis non compressés.

# DONNÉES ORIENTÉES « TRAJETS »



LIGNE	DATE	HEURE	N°	STATION
222	10/04/26	7h55	1	Route de Yainville
222	10/04/26	7h58	2	École de Jumièges
222	10/04/26	8h02	3	Rue Fessard
222	10/04/26	8h04	4	Rue du Quesney
222	10/04/26	8h06	5	La Petite Charbonnière
222	10/04/26	8h07	6	L'Essartel
222	10/04/26	8h09	7	Haute Ville
222	10/04/26	8h14	8	Le Vide du Bois
222	10/04/26	8h16	9	Ancienne École
222	10/04/26	8h18	10	La Trimarée
222	10/04/26	8h19	11	Rue Saint-Jean
222	10/04/26	8h23	12	Abbaye de Jumièges
222	10/04/26	8h25	13	École de Jumièges

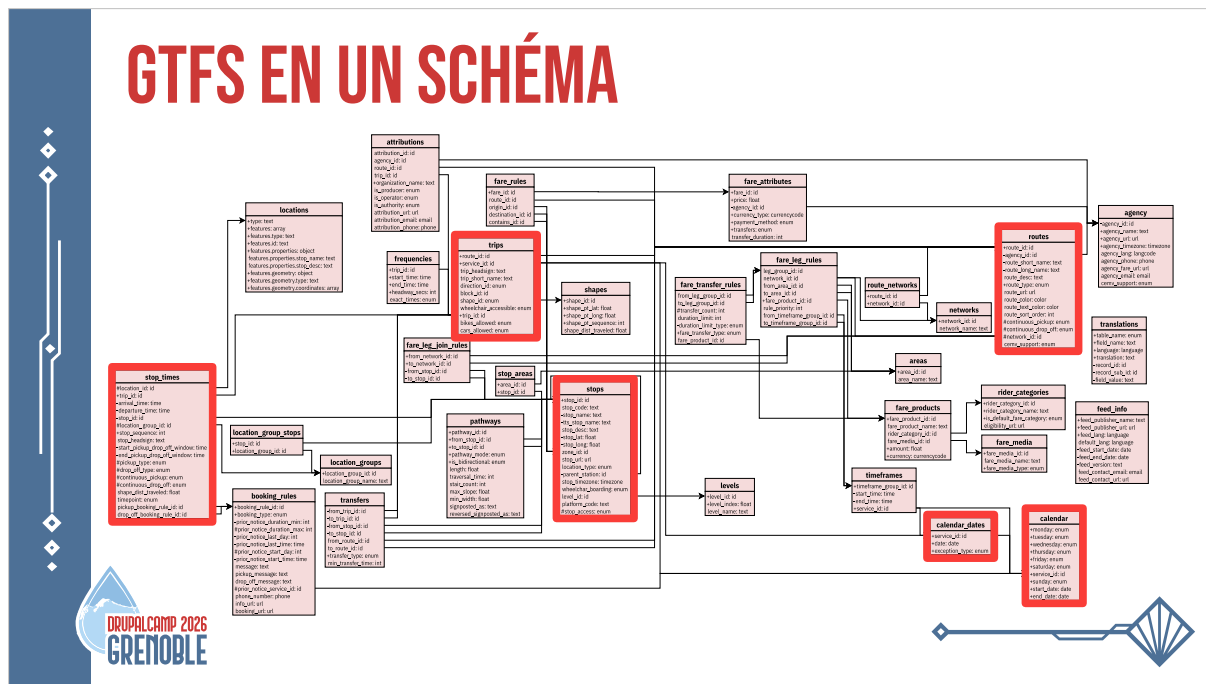
DRUPALCAMP 2026  
GRENOBLE

# 11

Données ouvertes et Drupal 11  
Transformer des données brutes  
en infos utiles

## Données orientées « trajets » (Notes)

Les données contenues dans les fichiers GTFS sont orientées « trajets », c'est-à-dire qu'elles permettent d'identifier tous les trajets avec les heures de passage à chaque arrêt, jour par jour. Cet exemple montre la ligne 222 et ses 13 arrêts de ce vendredi 10 avril, de 7h5 à 8h25. Il s'agit d'une ligne de transport scolaire n'ayant que deux trajets par jour, le trajet présenté étant celui du matin.



## GTFS en un schéma (Notes)

Bien que les données GTFS soient livrées sous forme de fichiers CSV, chaque fichier référence d'autres fichiers à la manière d'une base de données relationnelle.

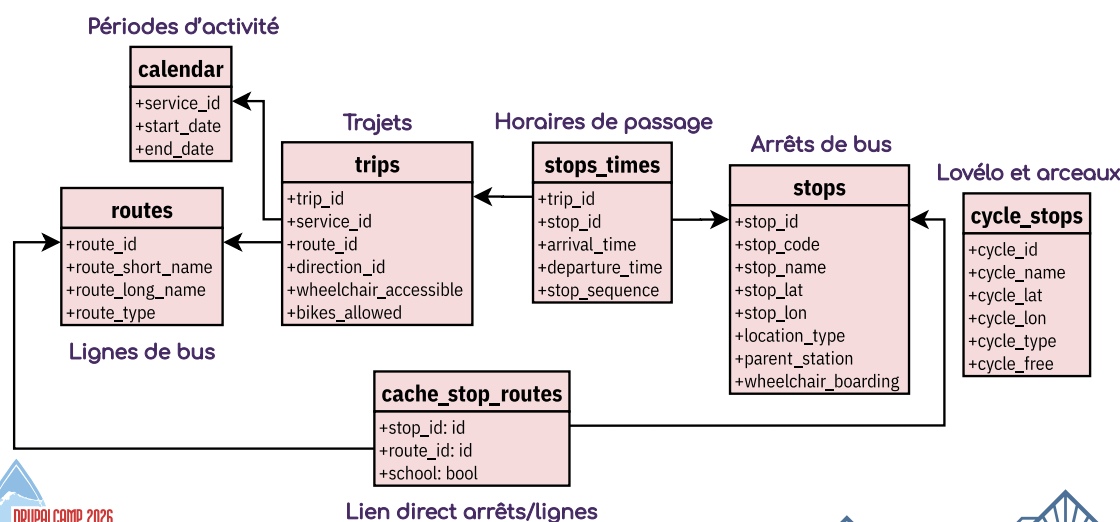
Le schéma présenté ici l'est uniquement pour se rendre compte de la complexité du format GTFS. Certaines tables sont en double pour représenter les mêmes données différemment comme calendar et calendar\_dates.

Ces données sont orientées « trajet » et elles contiennent de très nombreuses informations comme les horaires, les temps de trajets, les tarifs...

Or, dans notre cas, nous avons besoin de données orientées « arrêts ».

Les 6 fichiers qui nous seront utiles sont repérés grâce au cadre rouge sur le schéma.

# SIMPLIFICATION DU SCHÉMA



## Simplification du schéma (Notes)

Les fichiers CSV du format GTFS ne sont pas très pratiques à manipuler. On va donc créer une base de données classique composée de 7 tables seulement.

Nous avons d'abord besoin des arrêts avec leurs noms et coordonnées GPS. À l'autre bout, il nous faut également les lignes de bus avec leurs noms. Chaque ligne de bus va être concernée par plusieurs trajets. Ces trajets sont affectés par des périodes d'activité et des horaires de passages à des arrêts de bus.

Ces quelques tables correspondent à la vision « trajets » du format GTFS. Pour accélérer la recherche correspondant à notre cas et éviter de par de trop nombreuses jointures, on crée une table de cache réalisant un lien direct entre arrêts et lignes de bus.

Il faut enfin disposer d'une table pour contenir la liste des bornes de vélos en libre-service et les arceaux vélos.

## QUELQUES « SUBTILITÉS » DE GTFS

- Une information → plusieurs formats autorisés  
Par exemple avec les tables `calendar` et `calendar_dates`
- Une station → plusieurs arrêts  
La différence se fait sur les coordonnées GPS
- Des journées de plus de 24 heures  
24:10:00 ou 25:30:00 sont des horaires valides
- Pas d'identifiant universel  
Les identifiants doivent être réécrits pour la fusion



## Quelques « subtilités » de GTFS (Notes)

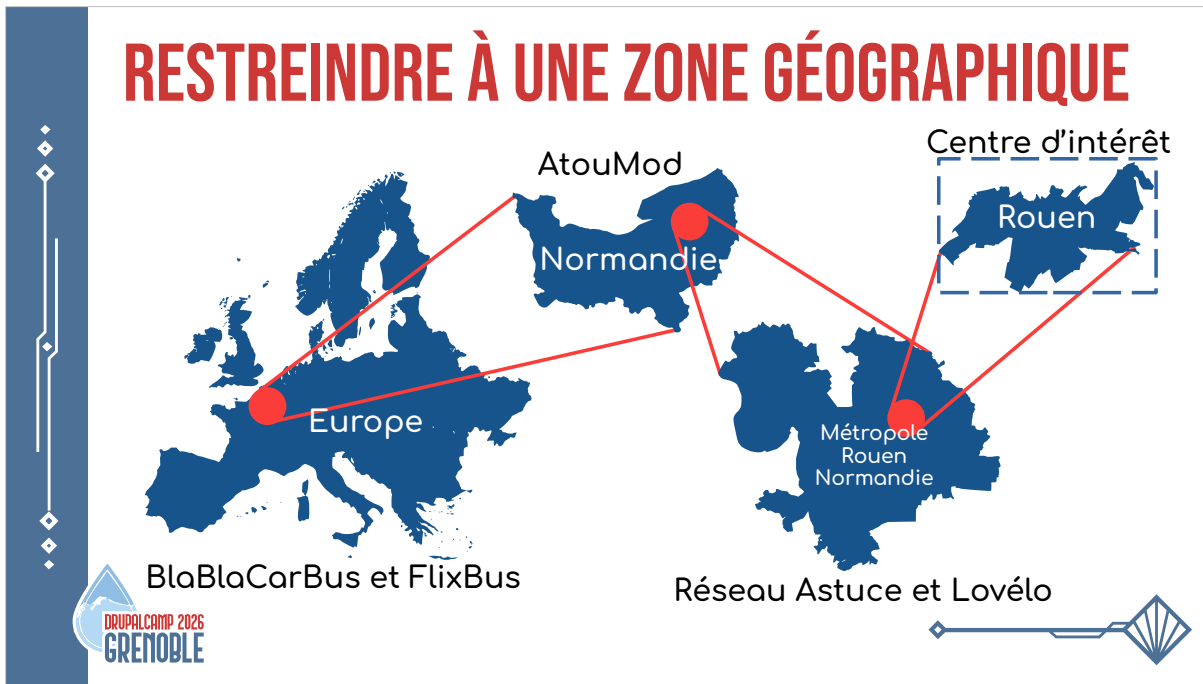
Élaguer les données n'est pas suffisant car GTFS autorise plusieurs façons d'enregistrer une information. Par exemple, les informations de période d'activité ont droit à deux fichiers : `calendar` (plages de validité et jours de la semaine) ou `calendar_dates` (exceptions de dates). Un mélange également possible.

Les données GTFS distinguent les stations et les arrêts. Par exemple, la station Hôtel de Ville à Rouen contiendra 4 arrêts, chacun avec des coordonnées GPS différentes.

Bien que cela n'ait pas d'incidence dans notre cas, les heures données dans des fichiers GTFS peuvent aller au-delà de 24 heures. Par exemple, 25h30 correspond à 1h30 du matin. Cela permet une gestion plus naturelle de services en décalage par rapport à un jour entier.

Il n'y a pas d'ID universel dans GTFS. Une fusion de plusieurs sources doit donc renommer les ID.

## RESTREINDRE À UNE ZONE GÉOGRAPHIQUE



Données ouvertes et Drupal 11  
Transformer des données brutes  
en infos utiles

## Restreindre à une zone géographique (Notes)

Pour réduire la taille des données, l'approche la plus fructueuse consiste à restreindre les données à une zone géographique.

Par exemple, BlaBlaCarBus et FlixBus ont un réseau européen, AtouMod couvre la Normandie et Réseau Astuce et Lovélo couvrent l'agglomération de Rouen.

Seules les données concernant Rouen nous intéressent. Et, pour cela, le plus simple reste une zone rectangulaire représentant les coordonnées extrêmes de la ville (latitude minimum et maximum, longitude minimum et maximum).

## CORRECTIONS TYPOGRAPHIQUES

- *Salle N. Armstrong* ⇒ Salle Neil Armstrong  
Développement d'abréviations
- *Pont S.N.C.F.* ⇒ Pont SNCF  
Suppression des points dans les sigles
- *VICTOR HUGO* ⇒ Victor Hugo  
Capitales abusives
- *MIN Lecoeur* ⇒ MIN Lecœur  
Accents et ligatures

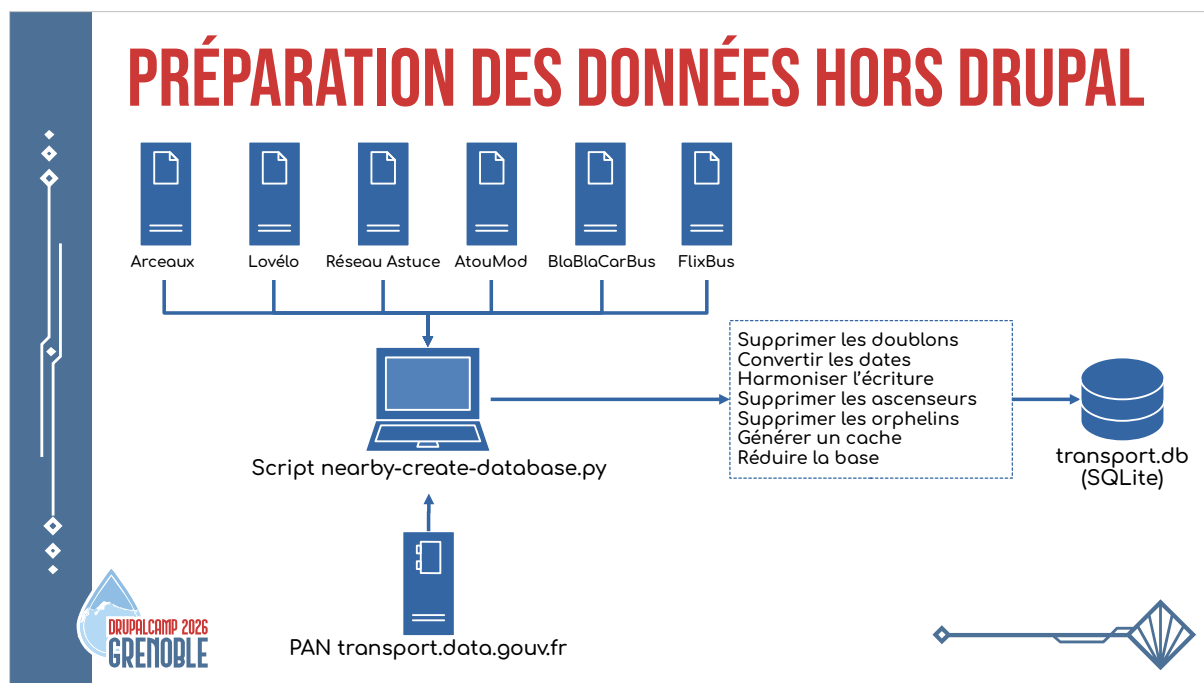


## Corrections typographiques (Notes)

Des corrections typographiques sont également opérées pour deux raisons : la première pour permettre une fusion propre, la seconde pour standardiser l'affichage des informations à l'utilisateur final.

Dans ces corrections typographiques, les abréviations vont être développées (si certaines semblent évidentes, il n'en va pas de même pour toutes les autres), les points dans les sigles vont être supprimés, les capitales abusives seront retirées et les accents et ligatures restaurés (car les noms tout en majuscule manquent souvent d'accents).

Ces corrections sont réalisées sur la base d'un simple fichier texte de correspondance chaîne à remplacer/chaîne de remplacement.



## Préparation des données (Notes)

Toutes ces opérations sont effectuées au sein d'un seul et unique script autonome: `nearby-create-database.py`.

Le script interroge le PAN afin d'obtenir les URLs vers les 6 ressources nécessaires: arceaux vélo, infos Lovélo, infos Réseau Astuce, infos AtouMod, BlaBlaCarBus et FlixBus. Il gère même le cas où les identifiants de ressource changent.

Le script procède ensuite au nettoyage (suppression des doublons, conversions des dates, harmonisation de l'écriture, génération du cache, suppression des orphelins...) et crée une base SQLite `transport.db`.

SQLite permet de réduire la taille physique d'une base après effacement des données inutiles. MariaDB ne permet pas cette opération: l'espace alloué n'est jamais rendu, il est juste marqué comme disponible.

## NEARBY-CREATE-DATABASE.PY EN ACTION

```
fred@prezigion:~/Programmation/transport-nearby/nearby-creates$
```

Tâche automatisable

Facilité de débogage

Peu de dépendances

Téléchargement: 27 s

Création de la base: 34 s



### nearby-create-database.py en action (Notes)

Voici ce que donne l'exécution du script. Cette tâche est automatisable par un cron ou par systemd.

Comme il s'agit d'une tâche prévue pour être exécutée sans intervention, les choix ont été orientés vers des solutions et pratiques facilitant le débogage : un seul script Python, un style procédural, des messages à chaque étape... Le script repose sur des bibliothèques standards de Python 3 à l'exception de requests qui est néanmoins disponible sur tous les dépôts des distributions Linux.

Les fichiers téléchargés sont placés dans un répertoire temporaire. Cela permet de corriger et relancer. La base de données elle-même est visualisable avec des outils comme sqlitebrowser. Tout est là pour mettre les mains dans le cambouis en cas de problème.

L'affichage a été accéléré. La partie téléchargement et la partie création de la base prennent environ 30 secondes chacune à s'exécuter.

# RISQUES LIÉS À LA RÉCUPÉRATION

- Indisponibilité d'une ressource  
Connexion réseau, indisponibilité, droits...
- Intégrité des données  
Cas imprévu, fichier corrompu, données manquantes...
- Instabilité de l'offre de transport  
Création/disparition, changement d'opérateur...
- Évolution des formats utilisés  
GTFS a évolué en septembre 2025 et GBFS fin 2024



## Risques liés à la récupération (Notes)

L'aspect débogage a été exploré car la récupération de données ouvertes peut être sujette à beaucoup de problèmes.

Une ressource peut être indisponible, soit par une connexion réseau temporairement close, soit par une indisponibilité du serveur interrogé ou un problème de droits d'accès.

L'intégrité des données est également un souci. Il n'est pas rare de faire face à un cas imprévu, à des fichiers corrompus voire des données manquantes.

L'offre de transport elle-même fluctue. Des réseaux peuvent être créés ou disparaître, les opérateurs changent aussi.

Et, même si GTFS et GBFS sont des standards, ceux-ci continuent d'évoluer régulièrement. La version actuelle de GTFS date de septembre 2025, celle de GBFS de 2024.

## SAVOIR S'ADAPTER



TYPE	ASTUCE	ATOUMOD	FLIXBUS	BLABLACARBUS
Erreurs	0	4 660	0	1
Warnings	102 686	113 617	109 049	7 874
Notice	0	0	4	0

- Valideur GTFS
  - <https://gtfs-validator.mobilitydata.org>
  - Limité à 100000 messages/type



## Savoir s'adapter (Notes)

La règle d'or est donc de savoir s'adapter à la situation.

Les dernières données récupérées et soumises à un valideur GTFS ont mis en évidence de nombreux problèmes.

Plus de 100000 avertissements et même des erreurs sur certains opérateurs.

## ÉTAPE 2


## CONSOMMATION DES DONNÉES



## Consommation des données (Notes)

Une fois les données produites, il va être temps de les consommer du côté de Drupal.

## LIVRAISON DES DONNÉES

- Push ou pull
- Manuel ou automatisé
- Base de données  SQLite = ~10 Mo
- Mode lecture seule
- Rafraîchissement du cache



Données ouvertes et Drupal 11  
Transformer des données brutes  
en infos utiles



### Livraison des données (Notes)

Il faut tout d'abord penser à la livraison de ces données.

Il s'agit de livrer un seul et unique fichier transport.db d'environ 10 Mo.

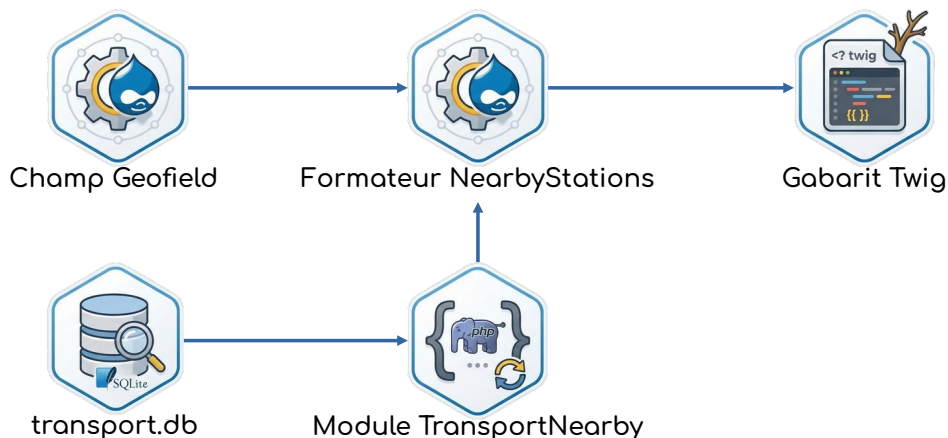
Cela peut se faire en push ou en pull, manuellement ou de façon automatisée, la solution est suffisamment souple pour cela.

À noter également que la base de données est en mode annule et remplace, les accès qu'elle supportera seront uniquement en lecture. Il n'y a donc pas de cas où un verrou sur une table pendant une écriture pourrait bloquer la lecture par un autre processus.

Un point à améliorer reste le rafraîchissement du cache de Drupal, celui-ci ne gérant pas directement la mise à jour de la base SQLite

# UTILISATION DES DONNÉES SOUS DRUPAL

# 23



Données ouvertes et Drupal 11  
Transformer des données brutes  
en infos utiles

## Utilisation des données (Notes)


Pour pouvoir afficher les informations sous Drupal, le principe est de s'appuyer sur un champ de type Geofield pour contenir des coordonnées GPS.

À ce champ Geofield on associera un formateur custom « NearbyStations » qui va jouer le rôle du contrôleur.

Ce formateur va utiliser le module custom TransportNearby pour récupérer les données. Ce module joue le rôle du modèle. C'est lui qui va interroger la base de données SQLite transport.db.

Les données récupérées sont ensuite envoyées par le formateur à des gabarits Twig.

## POURQUOI UN FORMATEUR ?



OPTION	AVANTAGES	LIMITES
Bloc	Simple à ajouter	Contexte moins naturel
View	Idéal avec des données Drupal	Peu adapté à notre source
Preprocess	Rapide	Peu configurable par UI
Champ calculé	Élégant sur le papier	Rigide
Formateur	Natif, réutilisable, configurable	Nécessite un peu de code



## Pourquoi un formateur ? (Notes)

Il est légitime de se demander pourquoi ne pas avoir utilisé un bloc, une View, un preprocess ou un champ calculé.

La réponse tient en deux mots : cohérence et flexibilité.

En utilisant un formateur, on reste dans le flux standard de rendu de Drupal. Contrairement à un bloc ou un controller custom, nous n'avons pas besoin de « deviner » ou de forcer le contexte de l'entité. Le formateur reçoit les coordonnées GPS du champ Geofield directement.

L'usage d'un preprocess a été écarté car il verrouille l'affichage dans le code, empêchant les éditeurs de déplacer l'info dans l'interface « Gérer l'affichage » ou de changer le rayon de proximité.

Les Views sont excellentes pour travailler sur des entités Drupal. Ici nous interrogeons une source externe pour enrichir un affichage. Le formateur fait ici office de « pont » léger et performant entre notre modèle de données et nos gabarits Twig.

The screenshot shows the 'Gérer les champs' (Manage fields) configuration page in Drupal. At the top, there are two buttons: '+ Créer un nouveau champ' and '+ Ré-utiliser un champ existant'. Below these are four tabs: 'Modifier', 'Gérer les champs' (highlighted with a red box), 'Gérer l'affichage du formulaire', 'Gérer l'affichage', and 'Traduire types de paragraphes'. The main content area is a table with three columns: 'Champ', 'Type de champ', and 'Actions'. The first row is for 'Coordonnées de référence' (field\_name: field\_reference\_point), which is highlighted with a red box. It is of type 'Geofield' and has 'Single' and 'Requis' (Required) options. The second row is for 'Titre' (field\_name: field\_title), which is of type 'Texte (brut)' and has a 'Single' option. Each row has a 'Modifier' button with a dropdown arrow.

Champ	Type de champ	Actions
Coordonnées de référence <small>field_reference_point</small> <small>Single</small> <small>Requis</small>	Geofield	<small>Modifier</small> ▼
Titre <small>field_title</small> <small>Single</small>	Texte (brut)	<small>Modifier</small> ▼

## Configuration : champ Geofield (Notes)

Un champ Geofield peut être ajouté à un type de contenu ou à un type de paragraphe, par exemple.

Il faut que ce champ ne contienne qu'un seul point. Le principe ne s'accommode pas de champs multi-valeurs ou de zones.

Paramètres du champ Coordonnées de référence pour Transports à proximité

Libellé \*

Coordonnées de référence

Stockage du champ

Ces paramètres s'appliquent au champ *Coordonnées de référence* partout où il est utilisé. Certains ont également un impact sur la manière dont les données sont stockées et ne peuvent plus être modifiées une fois que des données ont été créées.

Storage backend

Default (WKT)

Nombre de valeurs autorisées

Limité 1

Enregistrer les paramètres Supprimer

DRUPALCAMP 2026  
GRENOBLE

## Configuration : détails du champ (Notes)

Le champ Geofield doit avoir une configuration précise.

Les champs « Storage backend » et « Nombre de valeurs autorisées » doivent respectivement être définis à « Default (WKT) » et « Limité à 1 » pour pouvoir fonctionner avec le formateur

## Gérer l'affichage

Modifier Gérer les champs Gérer l'affichage du formulaire **Gérer l'affichage** Traduire types de paragraphes

Champ	Nom système	Libellé	Format	<a href="#">Afficher le poids des lignes</a>
<b>Content</b>				
Titre	field_title	- Non affiché -	Titre de niveau 2	⚙️
Coordonnées de référence	field_reference_point	- Non affiché -	<b>Nearby stations</b>	⚙️
<b>Désactivé(e)s</b>				
Aucun champ n'est caché.				



## Configuration : formateur (Notes)

Sur l'onglet « Gérer l'affichage », il suffit de sélectionner le formateur « Nearby stations ».

Gérer l'affichage

Modifier Gérer les champs Gérer l'affichage du formulaire **Gérer l'affichage** Traduire types de paragraphes

Champ	Nom système	Libellé	Format	<a href="#">Afficher le poids des lignes</a>
<b>Content</b>				
Titre	field_title	- Non affiché -	Titre de niveau 2	
Coordonnées de référence	field_reference_point	- Non affiché -	Paramètres de format: <b>Nearby stations</b> Distance maximum en mètres 300	
<b>Désactivé(e)s</b>				

Aucun champ n'est caché.

## Configuration : paramètres (Notes)

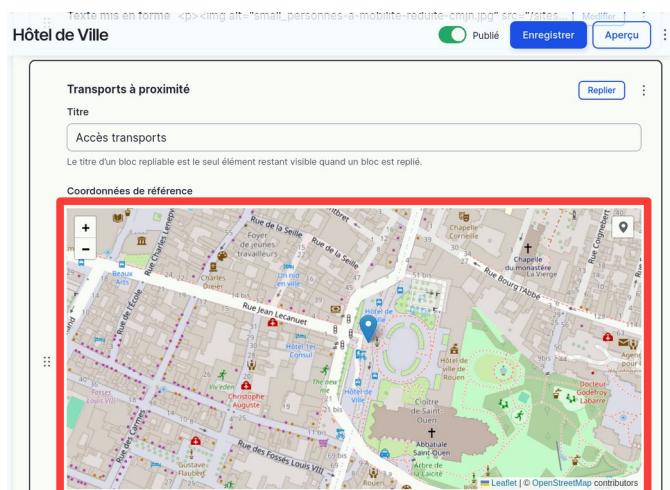
Il est possible de modifier la valeur par défaut de 300 mètres pour la distance maximale par rapport au point de référence.



## Édition : création d'un paragraphe (Notes)

Une fois le champs configuré, il ne reste plus qu'à l'utiliser.

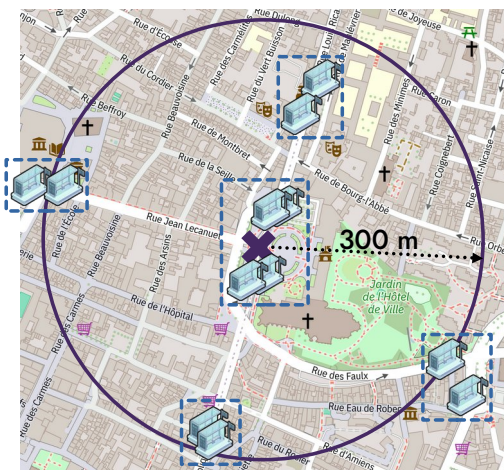
Dans cet exemple, un paragraphe de type « Transports à proximité » a été à un contenu dont le titre est « Hôtel de Ville » afin d'indiquer les transports à proximité de l'Hôtel de Ville.



## Édition : coordonnées (Notes)

L'utilisateur n'a plus qu'à sélectionner le lieu sur une carte interactive. Il est toujours possible de ne présenter qu'un champ basique permettant la saisie de coordonnées GPS.

## SOUS LE CAPOT



- Rayon = 300m
- 12 arrêts
- 5 stations
  - Lycée Corneille F1 20 305
  - Beaux-Arts F2 F7 11 15 22 98
  - Hôtel de Ville F1 F2 F7 11 15 20 22 98 305
  - Saint-Vivien F2 15 20 22 305
  - Saint-Nicolas F1 F7 11 98
- Après filtrage
  - Hôtel de Ville F1 F2 F7 11 15 20 22 98 305

DRUPALCAMP 2026  
GRENOBLE

## Sous le capot (Notes)

Voici comment les informations attendues vont être récupérées pour être affichées. La croix au centre correspond au point de référence.

Le formateur va rechercher tous les arrêts à 300 mètres autour de ce point. Dans cet exemple, on en compte 12.

Ces 12 arrêts correspondent à des arrêts physiques. Pour comprendre, il faut faire l'analogie avec les quais d'une gare. Ceux-ci sont des arrêts et la gare une station.

Ici les 12 arrêts correspondent à 5 stations : lycée Corneille, Beaux-Arts, Hôtel de Ville, Saint-Vivien et Saint-Nicolas.

Chaque station accueille plusieurs lignes de bus. On peut constater qu'une même ligne de bus passe par plusieurs stations. Ce qui va intéresser l'utilisateur final, c'est la station la plus proche du point de référence.

En filtrant, il ne reste plus qu'une seule station. Dans cet exemple, la station Hôtel de Ville est un carrefour.

## PERFORMANCE ET CACHE DANS DRUPAL

- Base SQLite en lecture seule
  - Lecture uniquement lors du rendu du champ
  - Mise à jour de transport.db simple et rapide
- Gestion du cache
  - Contexte de l'entité parente
  - Invalidation par changement de configuration



Données ouvertes et Drupal 11  
Transformer des données brutes  
en infos utiles



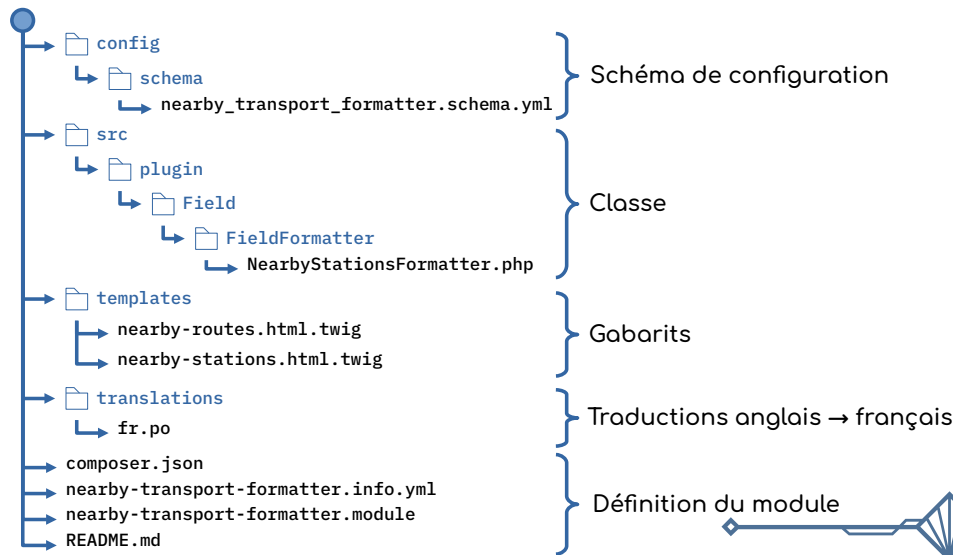
### Performance et cache dans Drupal (Notes)

La base SQLite étant en lecture seule exclusivement, il n'y a aucun verrou qui pourra interrompre sa consultation.

Par défaut, le rendu du formateur sera mis en cache par Drupal en suivant les contextes de l'entité parente (le nœud ou le paragraphe qui contient le champ Geofield). Puisque le rendu dépend des coordonnées GPS contenues dans le champ, la mise en cache au niveau de l'entité garantit que les résultats correspondent bien au lieu affiché.

Le rendu dépend aussi du réglage `max_distance`. Si un administrateur change cette distance dans l'interface « Gérer l'affichage », Drupal invalidera correctement le cache du formateur pour ce mode de vue.

## ORGANISATION DU FORMATEUR



## Organisation du formateur (Notes)

Si on plonge dans les aspects techniques du formateur, sa structure reste simple pour un module Drupal.

Elle contient un schéma de configuration pour la distance maximale.

Le fichier `NearbyStationsFormatter.php` contient la classe principale chargée de contrôler la récupération des données et leur affichage.

Le formateur utilise deux gabarits.

Un fichier de traductions anglais → français est inclus.

Et on trouve enfin les fichiers nécessaires à la définition du module pour Drupal.

# ET APRÈS ?



## Et après ? (Notes)

À partir de là, que fait-on ?

## FONCTIONNALITÉS SUPPLÉMENTAIRES

# 35

- Noms de lignes plus complets (A11Y)
- Accessibilité aux fauteuils roulants
- Accessibilité aux vélos
- Gestion du cache



- Support de zones
- Support d'autres villes
- Carte interactive
- Rayon de proximité individualisé



Données ouvertes et Drupal 11  
Transformer des données brutes  
en infos utiles



## Fonctionnalités supplémentaires (Notes)

L'affichage peut être enrichi d'infos supplémentaires comme des noms de lignes plus complets, ou l'accessibilité aux fauteuils roulants ou de l'accessibilité aux vélos.

On peut également supporter des zones plutôt que de points de référence. Cela peut être utile dans le cas de lieux ayant une grande superficie avec plusieurs points d'accès.

Côté open-source, le script de création de la base de données peut évoluer et supporter d'autres collectivités. Il n'y a en effet qu'un seul fichier à modifier pour y arriver.

On peut aussi imaginer présenter une carte interactive avec des icônes pour chaque élément à proximité du point de référence. Cela n'a pas été fait en premier lieu car le but est de proposer des informations accessibles au sens RGAA.

La possibilité de régler le rayon de proximité pour chaque valeur de champ serait un plus pour les cas particuliers.

## QUELQUES ENSEIGNEMENTS

- Un standard ne garantit pas la cohérence  
<https://gtfs-validator.mobilitydata.org>
- Nettoyer est plus long qu'exploiter
- Une architecture simple est souvent suffisante
- L'open data nécessite de la résilience



## Quelques enseignements (Notes)

Pour finir, quels enseignements peut-on tirer de cette expérience ?

Tout d'abord, avec les données ouvertes, ce n'est pas parce qu'il y a un standard officiel, même bien défini, que les données vont être cohérentes.

Souvent, le traitement le plus lourd se situe au niveau du nettoyage des données plutôt qu'au niveau de leur exploitation.

C'est pour cela qu'une architecture simple est souvent suffisante voire nécessaire.

Enfin, l'open data nécessite de la résilience, surtout quand on multiplie les sources de données. Mais cela reste un incontournable en matière de dev : ne jamais faire confiance aux données qu'on nous fournit.



## Merci à vous ! (Notes)

Merci à vous d'avoir suivi cette conférence.