

GeoPackage → GeoJSON / Shapefile

1. À quoi sert cet outil

L'outil convertit un fichier **GeoPackage** (**.gpkg**) en :

Sortie	Usage	Contenu
GeoJSON (partie A)	Une couche à la fois, avec transformations (reprojection, filtre, simplification, éclatement des multi-géométries)	Un .geojson , ou un .zip de .geojson en mode « toutes les couches »
Shapefile (partie B)	Toutes les couches (ou une sélection) en une fois, éventuellement fusionnées par type de géométrie	Un .zip contenant les .shp/.shx/.dbf/.prj

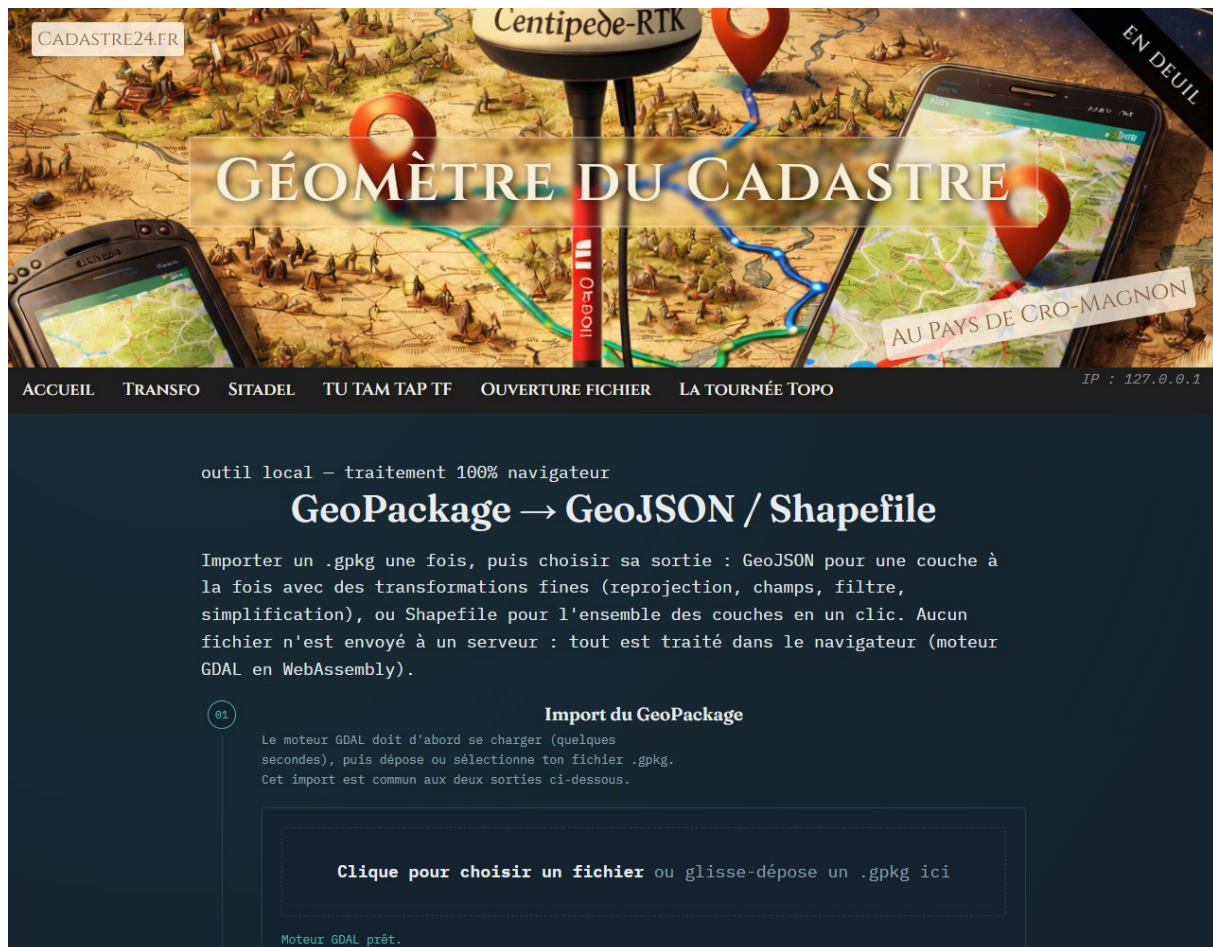
Tout est traité dans le navigateur : le fichier n'est jamais envoyé au serveur. La conversion utilise GDAL/OGR compilé en WebAssembly (bibliothèque *gdal3.js*).

2. Prérequis

- Un navigateur récent (Chrome, Edge, Firefox).
- **Une connexion Internet** : le moteur GDAL (~quelques Mo), Leaflet, JSZip et le fond de carte OpenStreetMap sont chargés depuis des CDN (jsdelivr, unpkg). Hors ligne, la page affiche « Échec du chargement du moteur GDAL ».
- Un fichier **.gpkg** contenant au moins une couche vectorielle.

Le traitement se faisant en mémoire dans le navigateur, un très gros GeoPackage (plusieurs centaines de Mo) peut ralentir ou saturer l'onglet.

3. Vue d'ensemble de la page



Vue générale de la page

La page est organisée en trois blocs, de haut en bas :

1. **Étape 01 — Import du GeoPackage** : commun aux deux sorties.
2. **Partie A — export GeoJSON, couche par couche** : étapes 1 (couche et champs), 2 (transformations) et 3 (génération).
3. **Partie B — export Shapefile, toutes les couches**.

Les pastilles numérotées à gauche suivent la progression : contour turquoise = étape en cours, pastille pleine = étape terminée.

Voir la page complète (avant import)



4. Étape 01 — Import du GeoPackage

4.1 Attendre le moteur GDAL

À l'ouverture de la page, le moteur GDAL se charge (quelques secondes). Pendant ce temps la zone de dépôt est **désactivée** :



Moteur GDAL en cours de chargement

Lorsque le message passe à « **Moteur GDAL prêt.** » (en turquoise), on peut importer un fichier :



Moteur GDAL prêt

4.2 Choisir le fichier

Deux possibilités :

- **cliquer** dans la zone en pointillés et choisir le **.gpkg** dans l'explorateur ;
- **glisser-déposer** le **.gpkg** sur la zone (elle s'éclaire en turquoise au survol).

Après lecture, un message confirme le nom du fichier et le nombre de couches détectées :

01

Import du GeoPackage

Le moteur GDAL doit d'abord se charger (quelques secondes), puis dépose ou sélectionne ton fichier .gpkg. Cet import est commun aux deux sorties ci-dessous.

Clique pour choisir un fichier ou glisse-dépose un .gpkg ici

Moteur GDAL prêt.

demo_cadastre.gpkg – 4 couche(s) détectée(s).

Fichier importé : 4 couches détectées

Dès ce moment :

- la liste « Couche source » (partie A) est remplie ;
- la liste des couches de la partie B est remplie (toutes cochées) ;
- les boutons **Générer le GeoJSON** et **Exporter en Shapefile** deviennent actifs.

Importer un nouveau fichier remplace le précédent (le jeu de données précédent est fermé).

5. Partie A — Export GeoJSON, couche par couche

A1. Couche et champs



Étape « Couche et champs »

Élément	Rôle
Exporter toutes les couches du GeoPackage	Active le mode multi-couches (voir plus bas).
Couche source	La couche à convertir. Le nombre d'entités est indiqué entre parenthèses.
Attributs à conserver	Une case par champ de la couche, toutes cochées par défaut, + liens <i>tout sélectionner</i> / <i>tout désélectionner</i> . Décocher un champ le retire du GeoJSON (ci-dessus, <i>commune</i> est exclu).

La liste des champs est rechargée à chaque changement de couche. Si la structure du fichier ne peut pas être lue, la page affiche « Aucun champ détecté... » et tous les attributs sont exportés.

A2. Transformations

Ces options sont appliquées par `ogr2ogr` pendant la conversion.

2

Transformations

Ces options sont appliquées pendant la conversion (via ogr2ogr).

Reprojection (CRS de sortie)

WGS84 (EPSG:4326)

Tolérance de simplification

vide = pas de simplification

Dans l'unité du CRS de sortie choisi ci-dessus (mètres pour Lambert-93, degrés pour WGS84).

Filtre attributaire (SQL WHERE)

statut = 'accepte'

Condition seule, sans « WHERE ». Valeurs texte entre guillemets simples : statut = 'accepte', contenance_m2 > 2000.

Multi-géométries

☐ Éclater en géométries simples (-explodecollections)

Transformations : WGS84 + filtre

Reprojection (CRS de sortie)

Choix	Effet
Conserver le CRS d'origine	Aucune reprojection.
Lambert-93 (EPSG:2154)	Reprojette en Lambert-93.
WGS84 (EPSG:4326)	Reprojette en longitude/latitude — recommandé pour un GeoJSON (c'est le CRS attendu par la norme RFC 7946 et par la plupart des outils web).
Autre — code EPSG personnalisé	Fait apparaître un champ où saisir le code : 3857 ou EPSG:3857 .

2

Transformations

Ces options sont appliquées pendant la conversion (via ogr2ogr).

Reprojection (CRS de sortie)

Autre — code EPSG personnalisé

3857

Tolérance de simplification

vide = pas de simplification

Dans l'unité du CRS de sortie choisi ci-dessus (mètres pour Lambert-93, degrés pour WGS84).

Filtre attributaire (SQL WHERE)

ex : statut = 'accepte'

Condition seule, sans « WHERE ». Valeurs texte entre guillemets simples : statut = 'accepte', contenance_m2 > 2000.

Multi-géométries

☐ Éclater en géométries simples (-explodecollections)

Code EPSG personnalisé

Si « Autre » est choisi mais que le champ est laissé vide, aucune reprojection n'est faite.

Tolérance de simplification

Valeur numérique passée à **-simplify**. Vide = pas de simplification. L'unité est celle du **CRS de sortie** : mètres en Lambert-93 (ex. **0.5**), degrés en WGS84 (ex. **0.00001** ≈ 1 m).

Filtre attributaire (SQL WHERE)

Restreint les entités exportées. On saisit **uniquement la condition**, sans le mot **WHERE**. La page construit la requête :

```
SELECT * FROM "<couche>" WHERE <votre condition>
```

Syntaxe (dialecte OGR SQL) :

Exemple	Signification
<code>statut = 'accepte'</code>	Texte : guillemets simples
<code>contenance_m2 > 2000</code>	Nombre
<code>section = 'AB' AND statut <> 'refuse'</code>	Combinaison
<code>numero IN ('0001','0002')</code>	Liste de valeurs
<code>commune LIKE 'Péri%'</code>	Motif

⚠ **N'utilisez pas de guillemets doubles autour d'une valeur texte** (`statut = "accepte"`) : en OGR SQL les guillemets doubles désignent un *nom de champ*, la conversion échoue. Il suffit de corriger le filtre et de relancer. Voir Dépannage.

Multi-géométries

Éclater en géométries simples (-explodecollections) : chaque partie d'un MultiPolygon / MultiLineString / MultiPoint devient une entité distincte (les attributs sont dupliqués). Le nombre d'entités en sortie peut donc augmenter.

A3. Génération et export

Cliquer sur **Générer le GeoJSON**. Pendant le traitement le bouton est grisé et une roue tourne. Une fois terminé :

3

Génération et export

Lance la conversion, vérifie l'aperçu, puis télécharge le GeoJSON.

Générer le GeoJSON

Réinitialiser

GeoJSON généré : parcelles.geojson

11

3.0 ko

4

entités

taille du fichier

champs conservés

Télécharger le GeoJSON

► Détails techniques (structure GDAL brute)

Résultat de la génération GeoJSON

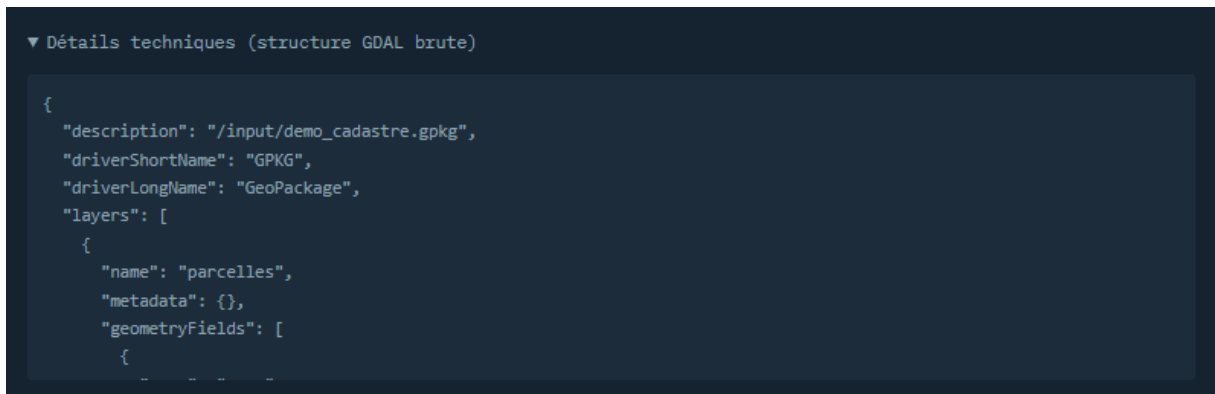
Élément	Contenu
Message d'état	GeoJSON généré : <couche>.geojson
entités	Nombre d'entités dans le fichier (après filtre / éclatement). Ici 11 parcelles au statut accepte sur 24.
taille du fichier	Poids du GeoJSON produit.
champs conservés	Nombre d'attributs gardés (ici 4 sur 5, commune ayant été décoché ; – si la liste des champs n'a pas pu être lue).
Carte d'aperçu	Les entités sur fond OpenStreetMap, cadrées automatiquement. Polygones/lignes en turquoise, points en orange. Molette désactivée : zoomer avec les boutons + / –.
Télécharger le GeoJSON	Enregistre <nom_de_couche>.geojson (caractères spéciaux remplacés par _).

Notes :

- L'aperçu est **toujours affiché en WGS84**, quel que soit le CRS choisi pour l'export. Il applique le filtre, mais pas la simplification ni l'éclatement.
- Les coordonnées sont écrites avec **7 décimales** (**COORDINATE_PRECISION=7**, soit ~1 cm en WGS84).

Détails techniques

Le bloc repliable « **Détails techniques (structure GDAL brute)** » affiche la description du fichier renvoyée par `ogrinfo -so -json` (couches, nombre d'entités, champs et types, type de géométrie, emprise, CRS). Utile pour diagnostiquer un fichier qui ne se lit pas correctement.



Détails techniques

Mode « toutes les couches »

En cochant **Exporter toutes les couches du GeoPackage**, le bloc couche/attributs est grisé (inactif) et le bouton devient **Générer les GeoJSON (toutes les couches)** :



Mode toutes les couches activé

Dans ce mode :

- la couche source et la sélection d'attributs sont **ignorées** — chaque couche est exportée avec tous ses champs ;
- la reprojection, le filtre, la simplification et l'éclatement **s'appliquent à chaque couche**. Attention : un filtre portant sur un champ absent d'une couche fera échouer cette couche (elle est alors ignorée et signalée) ;
- le résultat est un fichier **export_geojson.zip** contenant un **<couche>.geojson** par couche ;

- les statistiques affichent le total des entités, la taille du **.zip**, et « tous » pour les champs ; **il n'y a pas d'aperçu carte**.

3

Génération et export

Lance la conversion, vérifie l'aperçu, puis télécharge le GeoJSON.

Générer les GeoJSON (toutes les couches)

Réinitialiser

4 couche(s) exportée(s) – 19.4 ko.

98	19.4 ko	tous
entités	taille du fichier	champs conservés

Télécharger le GeoJSON

► Détails techniques (structure GDAL brute)

Résultat en mode toutes les couches

6. Partie B — Export Shapefile, toutes les couches

Cette partie est indépendante de la partie A (elle n'utilise ni le filtre ni la simplification).

partie B — export Shapefile, toutes les couches

Toutes les couches → Shapefile

Convertit en une fois l'ensemble des couches du GeoPackage en Shapefile (une copie complète par couche, champs et entités inclus), regroupées dans un .zip.

Couches à exporter [tout sélectionner](#) [tout désélectionner](#)

- ☒ parcelles (24 entités)
- ☒ batiments (24 entités)
- ☒ bornes (48 entités)
- ☒ voirie (2 entités)

☐ Fusionner les couches sélectionnées par type de géométrie (point / ligne / polygone)

Un Shapefile ne peut contenir qu'un seul type de géométrie : la fusion produit donc jusqu'à 4 fichiers (point / ligne / polygone / sans géométrie) au lieu d'un par couche, avec un champ couche_src indiquant la couche d'origine de chaque entité. Les tables sans géométrie (tables attributaires pures) sont regroupées à part, sans géométrie. Si deux couches ont des champs de même nom mais de nature différente, ou des noms tronqués à 10 caractères en collision, GDAL les renomme automatiquement (ex. champ_1).

Reprojection (CRS de sortie)

Conserver le CRS d'origine

Le Shapefile n'accepte qu'un seul type de géométrie et des noms de champs limités à 10 caractères (tronqués automatiquement par GDAL, sans perte de données).

Exporter en Shapefile (.zip)

Partie B — liste des couches

6.1 Choisir les couches

Toutes les couches sont cochées par défaut. Décocher celles à exclure, ou utiliser *tout sélectionner* / *tout désélectionner*. Au moins une couche doit être cochée (sinon : « Sélectionne au moins une couche. »).

6.2 Export simple (une couche = un Shapefile)

Sans l'option de fusion, chaque couche cochée donne un jeu .shp / .shx / .dbf / .prj portant le nom de la couche (caractères spéciaux remplacés par _, 40 caractères max).

Choisir éventuellement une **reprojection**, puis cliquer **Exporter en Shapefile (.zip)**. Le message indique le nombre de fichiers et la taille, puis le bouton **Télécharger le .zip** apparaît (`export_shapefile.zip`) :

Toutes les couches → Shapefile

Convertit en une fois l'ensemble des couches du GeoPackage en Shapefile (une copie complète par couche, champs et entités inclus), regroupées dans un .zip.

Couches à exporter [tout sélectionner](#) [tout désélectionner](#)

- ☒ parcelles (24 entités)
- ☒ bâtiments (24 entités)
- ☒ bornes (48 entités)
- ☒ voirie (2 entités)

☐ Fusionner les couches sélectionnées par type de géométrie (point / ligne / polygone)

Un Shapefile ne peut contenir qu'un seul type de géométrie : la fusion produit donc jusqu'à 4 fichiers (point / ligne / polygone / sans géométrie) au lieu d'un par couche, avec un champ `couche_src` indiquant la couche d'origine de chaque entité. Les tables sans géométrie (tables attributaires pures) sont regroupées à part, sans géométrie. Si deux couches ont des champs de même nom mais de nature différente, ou des noms tronqués à 10 caractères en collision, GDAL les renomme automatiquement (ex. `champ_1`).

Reprojection (CRS de sortie)

Conserver le CRS d'origine

Le Shapefile n'accepte qu'un seul type de géométrie et des noms de champs limités à 10 caractères (tronqués automatiquement par GDAL, sans perte de données).

Exporter en Shapefile (.zip)

16 fichier(s) – 34.0 ko.

Télécharger le .zip

Conversion effectuée localement via `gdal2shp` (GDAL/OGC compilé en WebAssembly)

Résultat de l'export Shapefile

Exemple avec le fichier de démo : 4 couches → 16 fichiers.

6.3 Export fusionné par type de géométrie

Cocher **Fusionner les couches sélectionnées par type de géométrie** regroupe toutes les couches cochées en **au plus 4 Shapefiles** :

Fichier	Contenu
<code>fusion_point</code>	toutes les couches de points
<code>fusion_ligne</code>	toutes les couches de lignes
<code>fusion_polygone</code>	toutes les couches de polygones
<code>fusion_sansgeom</code>	les tables attributaires sans géométrie (<code>.dbf</code> seul)

Chaque entité reçoit un champ `couche_src` contenant le nom de sa couche d'origine. Les géométries simples sont promues en multi (`PROMOTE_TO_MULTI`) pour pouvoir cohabiter.

Toutes les couches → Shapefile

Convertit en une fois l'ensemble des couches du GeoPackage en Shapefile (une copie complète par couche, champs et entités inclus), regroupées dans un .zip.

Couches à exporter [tout sélectionner](#) [tout désélectionner](#)

- ☒ parcelles (24 entités)
- ☒ batiments (24 entités)
- ☒ bornes (48 entités)
- ☒ voirie (2 entités)

☒ Fusionner les couches sélectionnées par type de géométrie (point / ligne / polygone)

Un Shapefile ne peut contenir qu'un seul type de géométrie : la fusion produit donc jusqu'à 4 fichiers (point / ligne / polygone / sans géométrie) au lieu d'un par couche, avec un champ couche_src indiquant la couche d'origine de chaque entité. Les tables sans géométrie (tables attributaires pures) sont regroupées à part, sans géométrie. Si deux couches ont des champs de même nom mais de nature différente, ou des noms tronqués à 10 caractères en collision, GDAL les renomme automatiquement (ex. champ_1).

Reprojection (CRS de sortie)

Lambert-93 (EPSG:2154)

La fusion impose une reprojection commune (les couches d'origine peuvent avoir des CRS différents) : Lambert-93 est choisi par défaut, modifiable ci-dessus.

Exporter en Shapefile (.zip)

Conversion effectuée localement via `gdal2xyz` (GDAL / OGR compilé en WebAssembly)

Options de fusion

En cochant la fusion, **la reprojection passe automatiquement à Lambert-93** (si elle était sur « Conserver ») : les couches d'origine pouvant avoir des CRS différents, un CRS commun est obligatoire. Il reste modifiable. Si on remet « Conserver le CRS d'origine », Lambert-93 est malgré tout utilisé.

Toutes les couches → Shapefile

Convertit en une fois l'ensemble des couches du GeoPackage en Shapefile (une copie complète par couche, champs et entités inclus), regroupées dans un .zip.

Couches à exporter [tout sélectionner](#) [tout désélectionner](#)

- ☒ parcelles (24 entités)
- ☒ bâtiments (24 entités)
- ☒ bornes (48 entités)
- ☒ voirie (2 entités)

☒ Fusionner les couches sélectionnées par type de géométrie (point / ligne / polygone)

Un Shapefile ne peut contenir qu'un seul type de géométrie : la fusion produit donc jusqu'à 4 fichiers (point / ligne / polygone / sans géométrie) au lieu d'un par couche, avec un champ couche_src indiquant la couche d'origine de chaque entité. Les tables sans géométrie (tables attributaires pures) sont regroupées à part, sans géométrie. Si deux couches ont des champs de même nom mais de nature différente, ou des noms tronqués à 10 caractères en collision, GDAL les renomme automatiquement (ex. champ_1).

Reprojection (CRS de sortie)

Lambert-93 (EPSG:2154)

La fusion impose une reprojection commune (les couches d'origine peuvent avoir des CRS différents) : Lambert-93 est choisi par défaut, modifiable ci-dessus.

Exporter en Shapefile (.zip)

12 fichier(s) – 52.4 ko.

Télécharger le .zip

Conversion effectuée localement via gdal2.13.0 (GDAL/OGC compilé en WebAssembly)

Résultat de l'export fusionné

6.4 Contraintes du format Shapefile

- un seul type de géométrie par fichier (d'où la fusion par type) ;
- noms de champs limités à 10 caractères : GDAL les tronque automatiquement (contenance_m2 → contenance) ; en cas de collision il renomme (champ_1...) ;
- en mode fusion, deux couches ayant un champ de même nom mais de types différents peuvent aussi provoquer un renommage.

7. Réinitialiser

Le bouton **Réinitialiser** (partie A, étape 3) ferme le fichier et remet **toute la page** à zéro (partie A et partie B) : listes vidées, options remises par défaut, résultats et carte masqués. Il faut ensuite réimporter un fichier.

Le moteur GDAL reste chargé : pas besoin de recharger la page.

8. Messages et dépannage

Message	Cause	Solution
Échec du chargement du moteur GDAL. Vérifie la connexion au CDN jsdelivr.	Pas d'Internet, CDN bloqué (proxy, pare-feu).	Vérifier la connexion, recharger la page.
La zone de dépôt ne réagit pas	Le moteur n'est pas encore prêt.	Attendre « Moteur GDAL prêt. ».
Aucune couche vectorielle détectée dans ce fichier.	GeoPackage raster uniquement, ou vide.	Vérifier le fichier dans QGIS.
Erreur à l'ouverture du fichier : ...	Fichier corrompu ou pas un GeoPackage.	Vérifier l'extension et le contenu.
Erreur pendant la conversion : Pointer 'hds' is NULL in 'GDALGetFileList'. — vérifie le filtre...	Le plus souvent : filtre SQL invalide (guillemets doubles, champ inexistant, faute de syntaxe) ou code EPSG inconnu.	Corriger le filtre (guillemets simples) ou le code EPSG, puis cliquer à nouveau sur <i>Générer</i> . Pas besoin de réimporter.
N couche(s) ignorée(s) — détail dans la console.	Une ou plusieurs couches n'ont pas pu être converties.	Ouvrir la console du navigateur (F12) pour le détail.
Aucun fichier n'a pu être généré...	Toutes les couches ont échoué.	Idem, consulter la console.

Exemple de l'erreur provoquée par un filtre avec guillemets doubles (`statut = "accepte"`) :

3

Génération et export

Lance la conversion, vérifie l'aperçu, puis télécharge le GeoJSON.

Générer le GeoJSON

Réinitialiser

Erreur pendant la conversion : Pointer 'hds' is NULL in 'GDALGetFileList'. — vérifie le filtre : valeurs texte entre guillemets simples (ex : statut = 'accepte') et noms de champs existants.

► Détails techniques (structure GDAL brute)

Erreur de filtre

9. Fonctionnement technique

Bibliothèques chargées par la page :

Bibliothèque	Version	Rôle
gdal3.js	2.8.1	GDAL/OGR en WebAssembly (ogr2ogr , getInfo)
Leaflet	1.9.4	Carte d'aperçu
JSZip	3.10.1	Création des archives .zip

Commande [ogr2ogr](#) équivalente pour l'export GeoJSON d'une couche :

```
ogr2ogr -f GeoJSON sortie.geojson fichier.gpkg \  
-dialect OGRSQL -sql 'SELECT * FROM "parcelles" WHERE statut = '""'accepte'""' \  
-lco COORDINATE_PRECISION=7 \  
-t_srs EPSG:4326 -simplify 0.00001 -explodecollections
```

Export Shapefile simple (par couche) :

```
ogr2ogr -f "ESRI Shapefile" -overwrite <nom_couche> fichier.gpkg \  
-dialect OGRSQL -sql 'SELECT * FROM "<couche>"' [-t_srs EPSG:xxxx]
```

Export fusionné : chaque couche est d'abord convertie en GeoJSON (dans le CRS commun), les entités sont regroupées par type de géométrie avec ajout de [couche_src](#), puis chaque groupe est écrit en Shapefile avec [-nlt PROMOTE_TO_MULTI](#) (ou [-nlt NONE](#) pour les tables sans géométrie).

La sélection d'attributs (quand elle est disponible) est appliquée **après** la conversion, en JavaScript, sur les propriétés du GeoJSON.

10. Limites connues

- Dépendance aux CDN : la page ne fonctionne pas hors ligne.
- Traitement en mémoire : les très gros fichiers peuvent saturer l'onglet.
- Le message d'erreur de GDAL reste technique ([Pointer 'hDS' is NULL...](#)) ; la page ajoute une indication sur le filtre quand un filtre est saisi.

Corrections apportées (octobre 2026)

Problème constaté	Correction
La liste « Attributs à conserver » restait vide (getInfo() de gdal3.js 2.8.1 ne renvoie ni champs ni type de géométrie).	Lecture de la structure avec Gdal.ogrinfo(['-so', '-json']) , repli sur getInfo() en cas d'échec.
Exemple du filtre trompeur (statut = "accepte").	Exemple corrigé en statut = 'accepte' + aide sous le champ.
Après une erreur de conversion, toutes les conversions suivantes échouaient jusqu'au réimport du fichier.	Chaque conversion écrit dans un fichier de sortie au nom unique (nextOutName()), le fichier laissé par l'échec ne bloque plus ; message d'erreur complété quand un filtre est saisi.
Classe CSS dimmed non définie : le bloc couche/champs n'était pas grisé en mode « toutes les couches ».	Classe .dimmed ajoutée (opacité réduite, bloc non cliquable).