



INSTITUT NATIONAL
DE L'INFORMATION
GÉOGRAPHIQUE
ET FORESTIÈRE

Documentation utilisateur API Carto module wfs-geoportail

Version 1.0

SOMMAIRE

I Description générale du module wfs-geoportail d'API Carto	3
I.1 Présentation d'API Carto	3
I.2 Présentation du module wfs-geoportail d'API Carto	3
I.3 Fonctionnement général du module wfs-geoportail d'API Carto	5
I.4 Description des paramètres à fournir en entrée	5
I.4.a Couche source	5
I.4.b Clé de géoservices	5
I.4.c Requête par géométrie	6
I.4.d Limite du nombre d'objets dans le résultat	6
I.4.e Format du résultat en sortie	6

GLOSSAIRE

API	« application programming interface », « interface de programmation applicative ».
API Carto	API développée initialement dans le cadre d'un PIA (projet d'investissement de d'avenir) dans le but de faciliter l'usage de la cartographie dans les formulaires administratifs

I Description générale du module wfs-geoportail d'API Carto

I.1 Présentation d'API Carto

API Carto est une brique logicielle offrant des webservices de traitements et de calculs, facilement intégrables dans les interfaces avec les usagers des services publics et reposant sur un ensemble de données géographiques de référence détenues par différents organismes. API Carto est aujourd'hui composée de différents modules thématiques : RPG, Cadastre, Codes Postaux, appellations viticoles et Urbanisme.

Les services proposés permettent de croiser des données entre elles, et donc de récupérer automatiquement certaines informations requises dans des formulaires administratifs (ex. : pour une demande de permis de construire, l'identifiant de la parcelle cadastrale peut souvent être directement obtenu à partir de l'adresse) ou d'extraire et télécharger des données.

API Carto a vocation à être utilisée par les utilisateurs finaux au travers de services clients.

Les grands principes techniques d'API Carto sont les suivants :

- API Carto est une collection d'API Rest respectant la spécification OpenAPI,
- le format utilisé pour les données est JSON/GeoJSON,
- la projection utilisée est WGS84 (coordonnées décimales en longitude, latitude),
- les API offrent des opérations génériques de filtrage simple:
 - par attribut (?nom_attribut=valeur),
 - par intersection géométrique (?geom=géométrie GeoJSON),
- l'enchaînement d'appels successifs aux différentes API (pour répondre aux besoins métiers) est à réaliser côté client,
- les traitements géométriques métiers (calcul de surface, filtrage des résultats, etc.) sont réalisés côté client, à l'aide de bibliothèques de calcul géométrique.

Une documentation technique est disponible en ligne pour plus d'informations : <https://apicarto.ign.fr/api/doc/wfs-geoportail> .

I.2 Présentation du module wfs-geoportail d'API Carto

Le module wfs-geoportail d'API Carto permet d'obtenir des informations sur toutes les couches WFS du géoportail disponibles dans le référentiel géographique EPSG:4326 (WGS 84).

Seules les requêtes géométriques sont possibles, un tri sur les valeurs d'attribut devra être fait en aval côté client.

Les utilisateurs finaux utilisent le module wfs-geoportail d'API Carto au travers de services clients. Cela peut être pour :

- remplir un formulaire
- extraire et télécharger des données de couches WFS du géoportail

Le schéma ci-dessous décrit le cas d'utilisation générale du module nature de l'API Carto par un tel outil d'assistance en ligne :

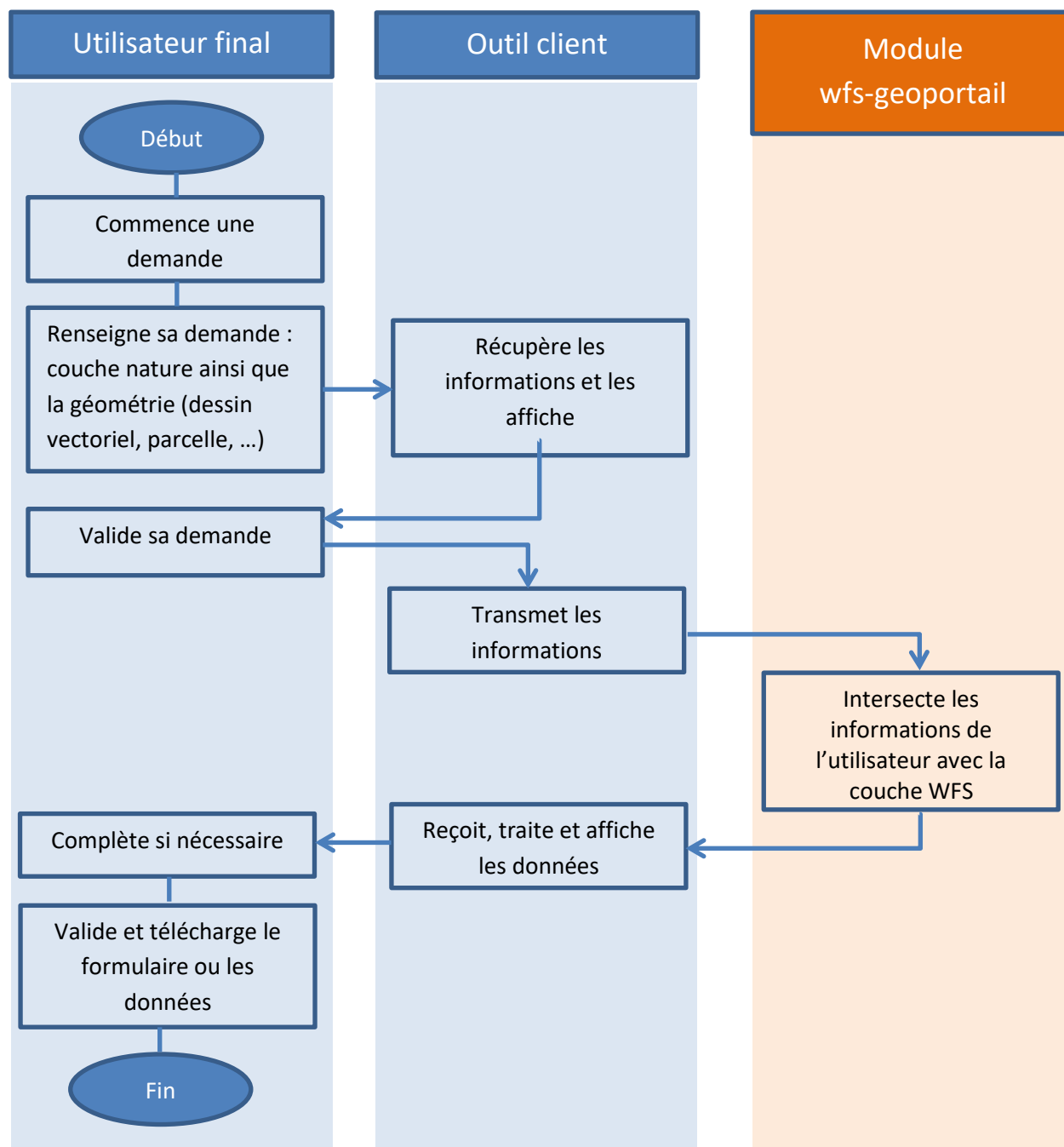


Figure 1 : cas d'utilisation générale du module nature

Le module wfs-geoportail d'API Carto récupère les données de la couche qui correspondent aux informations saisies par l'utilisateur final :

- un flux WFS (paramètre *source* obligatoire)
- une géométrie (paramètre *geom* obligatoire)

Note : il faut obligatoirement une clé de géoservices valide dans le paramètre *apikey*.

Les informations de la couche WFS intersectant la géométrie transmise sont ensuite renvoyées à l'outil client, qui se charge de leur traitement et de leur transfert sous la forme requise pour

l'utilisateur final (pré-remplissage des champs d'un formulaire, affichage cartographique, affichage des attributs sous forme de tableau etc...).

La documentation technique du module wfs-geoportail d'API Carto est disponible à cette adresse : <https://apicarto.ign.fr/api/doc/wfs-geoportail> .

I.3 Fonctionnement général du module wfs-geoportail d'API Carto

Le fonctionnement du module wfs-geoportail d'API Carto est soumis à la possession d'une clé de géoservices du géoportail de l'IGN valide ouvrant le droit d'utiliser la couche WFS choisie en paramètre de la requête.

Les requêtes peuvent se faire en POST ou en GET.

L'outil client doit envoyer une géométrie au format GeoJSON au module wfs-geoportail.

Le module wfs-geoportail récupère parmi les données de la couche paramétrée dans l'attribut « *source* » les objets (avec leurs attributs) intersectant la géométrie fournie en entrée et les renvoie à l'outil client sous forme de géométries au format GeoJSON.

Deux paramètres optionnels supplémentaires permettent de restreindre la réponse (Cf. §1.4.d ci-dessous) :

- `_limit` qui restreint le nombre d'objets dans la réponse
- `_start` qui définit à partir de quelle position dans la liste les objets sont renvoyés.

I.4 Description des paramètres à fournir en entrée

I.4.a Couche source

La liste des couches WFS du géoportail peut être obtenue via l'URL

<https://wxs.ign.fr/geoportail/wfs?SERVICE=WFS&VERSION=2.0.0&REQUEST=GetCapabilities>

Actuellement seules les couches exprimées dans le référentiel WGS 84 (EPSG:4326) peuvent être utilisées.

Il est possible de vérifier le référentiel d'une couche en consultant la valeur de la balise `<DefaultCRS>`.

Valeur du référentiel géographique (CRS) des flux utilisables par le module :

`<DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>`

Le nom technique de la couche source doit être utilisé. Celui-ci est indiqué par la balise `<Name>`.

Exemple : `<Name>BDTOPO_V3_BDD_WLD_WGS84G_LATEST_204:troncon_de_voie_ferree</Name>`

I.4.b Clé de géoservices

Pour accéder aux ressources du géoportail il est nécessaire de souscrire une clé de géoservices.

La clé de géoservices doit inclure les ressources **WFS** de la couche utilisée.

Note : pour connaître les ressources disponibles au travers de votre clé utilisez la commande

[https://wxs.ign.fr/\[CLE\]/geoportail/wfs?SERVICE=WFS&VERSION=2.0.0&REQUEST=GetCapabilities](https://wxs.ign.fr/[CLE]/geoportail/wfs?SERVICE=WFS&VERSION=2.0.0&REQUEST=GetCapabilities) où `[CLE]` est le nom de votre clé de géoservices.

La clé doit être sécurisée par `referer`.

Les ressources et la sécurisation sont à paramétrer dans l'espace client du site geoservices.ign.fr .

Le nom de la clé doit être rempli dans le paramètre *apikey*. Ce paramètre est obligatoire.

I.4.c Requête par géométrie

Toutes les ressources peuvent être interrogées par géométrie.

Les flux utilisés doivent géoréférencés en WGS 84 EPSG:4326, c'est dans ce référentiel que doivent être exprimées les coordonnées des requêtes géométriques.

Ce paramètre est obligatoire.

La géométrie fournie par l'outil client doit :

- être une géométrie valide au sens OGC,
- être une géométrie de type Polygon, MultiPolygon ou Point,
- avoir des coordonnées définies en projection WGS84 (coordonnées longitude, latitude exprimées en degrés décimaux),
- être envoyée au format JSON.

Exemple de géométrie JSON pour une surface :

```
{"type":"MultiPolygon","coordinates":[[[[-67061.292025474525872,5941189.475168794393539],[192234.901351706386777,5951541.333317489363253],[209093.148515937849879,5796262.406441408209503],[103398.038114650276839,5784629.464328233152628],[-67061.292025474525872,5941189.475168794393539]]]]}
```

Exemple de géométrie JSON pour un localisant ponctuel :

```
{"type":"Point","coordinates":[103398.038114650276839,5784629.464328233152628]}
```

L'interrogation du module wfs-geoportail d'API Carto avec plusieurs géométries n'est pas permise. Si le besoin de l'utilisateur concerne plusieurs géométries, il faudra donc que l'outil client enchaîne plusieurs requêtes API Carto (une par géométrie), puis traite les réponses successives de l'API pour présenter un résultat unifié à l'utilisateur final.

I.4.d Limite du nombre d'objets dans le résultat

Afin de ne pas saturer les serveurs pas des requêtes trop lourdes, le résultat renvoyé est limité à 1000 objets.

Les paramètres *_limit* et *_start* permettent gérer côté client le nombre d'objets renvoyés.

_limit limite le nombre d'objets renvoyés par un appel de l'API. Ce nombre doit être inférieur ou égal à 1000.

_start permet de récupérer les objets à partir du rang *_start* dans la liste des objets répondant aux critères saisis dans l'appel à l'API.

I.4.e Format du résultat en sortie

Le résultat de la requête est renvoyé par l'API au format JSON, sous forme de **FeatureCollection**. Le nombre de résultats est indiqué dans l'attribut **totalFeatures**.

Le fichier geoJSON en sortie respecte donc la structure suivante :

```
{
  "type":"FeatureCollection",
  "features":
  [.....]
  "totalFeatures": ,
}
```

Pour chaque feature, on retrouve ensuite son type, son **identifiant**, sa **géométrie** (type + coordonnées en WGS84) et la liste de ses **propriétés** sous la forme « attribut » : « valeur » ainsi que son rectangle englobant.

Description des features dans le fichier résultat :

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": "pn.18",
      "geometry": {
        "type": "MultiPolygon",
        "coordinates": [...]
      },
      "geometry_name": "the_geom",
      "properties": {
        "id_mnhn": "FR3300001",
        "nom": "Vanoise",
        "url": "https://inpn.mnhn.fr/espace/protege/FR3300001",
        "bbox": [
          6.60387389,
          45.22348386,
          7.18033886,
          45.56770488
        ]
      }
    },
    {
      "totalFeatures": 1,
      "numberMatched": 1,
      "numberReturned": 1,
      "timeStamp": "2020-12-09T15:34:44.015Z",
      "crs": {
        "type": "name",
        "properties": {
          "name": "urn:ogc:def:crs:EPSG::4326"
        }
      },
      "bbox": [
        6.60387389,
        45.22348386,
        7.18033886,
        45.56770488
      ]
    }
  ]
}
```