



INSTITUT NATIONAL
DE L'INFORMATION
GÉOGRAPHIQUE
ET FORESTIÈRE

Documentation utilisateurs API Carto module cadastre

Version 1.0

SOMMAIRE

I Description générale du module Cadastre d'API Carto	3
I.1 Présentation de l'API Carto	3
I.2 Présentation du module cadastre d'API Carto	3
I.3 Fonctionnement général du module cadastre d'API Carto	5
I.4 Modes d'interrogation des données disponibles	5
I.4.a Commune	5
I.4.b division/feuille	5
I.4.c Parcelle	6
I.4.d localisant	6
I.5 Description des paramètres à fournir en entrée	6
I.5.a Clé de géoservices	6
I.5.b Requête par géométrie	7
I.5.c Limite du nombre d'objets dans le résultat	7
I.5.d Format du résultat en sortie	7
II Conseils d'utilisation du module cadastre	10
II.1 Paramétrage de la ressource	10
II.2 Géométrie en entrée de la requête	10
II.3 Paramétrage des requêtes avec plusieurs valeurs d'un même paramètre	10
II.4 Requêtes avec plus de 1000 réponses	11
II.5 Détection des résultats vides	11

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
BD Parcellaire	Produit BD Parcellaire© de l'IGN dont le descriptif de contenu est disponible au travers de ce lien .
PCI Express	Produit Parcellaire Express (PCI) de l'IGN créé à partir des PCI (Plan Cadastral Informatisé) vecteur de la DGFIP. Son descriptif de contenu est disponible au travers de ce lien .

I Description générale du module Cadastre d'API Carto

I.1 Présentation de l'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. L'API Carto est aujourd'hui composée de différents modules thématiques : RPG, Cadastre, Codes Postaux, AOC, 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 de l'API Carto sont les suivants :

- API Carto est une collection API Rest respectant la spécification OpenAPI,
- le format utilisé pour les données est JSON/GeoJSON,
- la projection utilisée est WGS84 (coordonnées 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 différents 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/>.

I.2 Présentation du module cadastre d'API Carto

Le module cadastre d'API Carto permet d'obtenir des informations cadastrales à partir des flux WFS du géoportail de l'IGN. L'API fait appel à des ressources WFS du géoportail de l'IGN. Il est possible d'utiliser les ressources de la BD Parcellaire ou de PCI Express.

Les produits BD Parcellaire et PCI Express ne sont pas identiques, y compris géométriquement. Un comparatif est disponible au travers de [ce lien](#). Dans le module cadastre d'API Carto, ceci est particulièrement visible dans les sous-modules « Division BD Parcellaire » et Feuille PCI Express », division et feuille étant des termes représentant les mêmes notions mais avec une sémantique différente entre deux produits.

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

- remplir un formulaire
- extraire et télécharger des données du cadastre

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

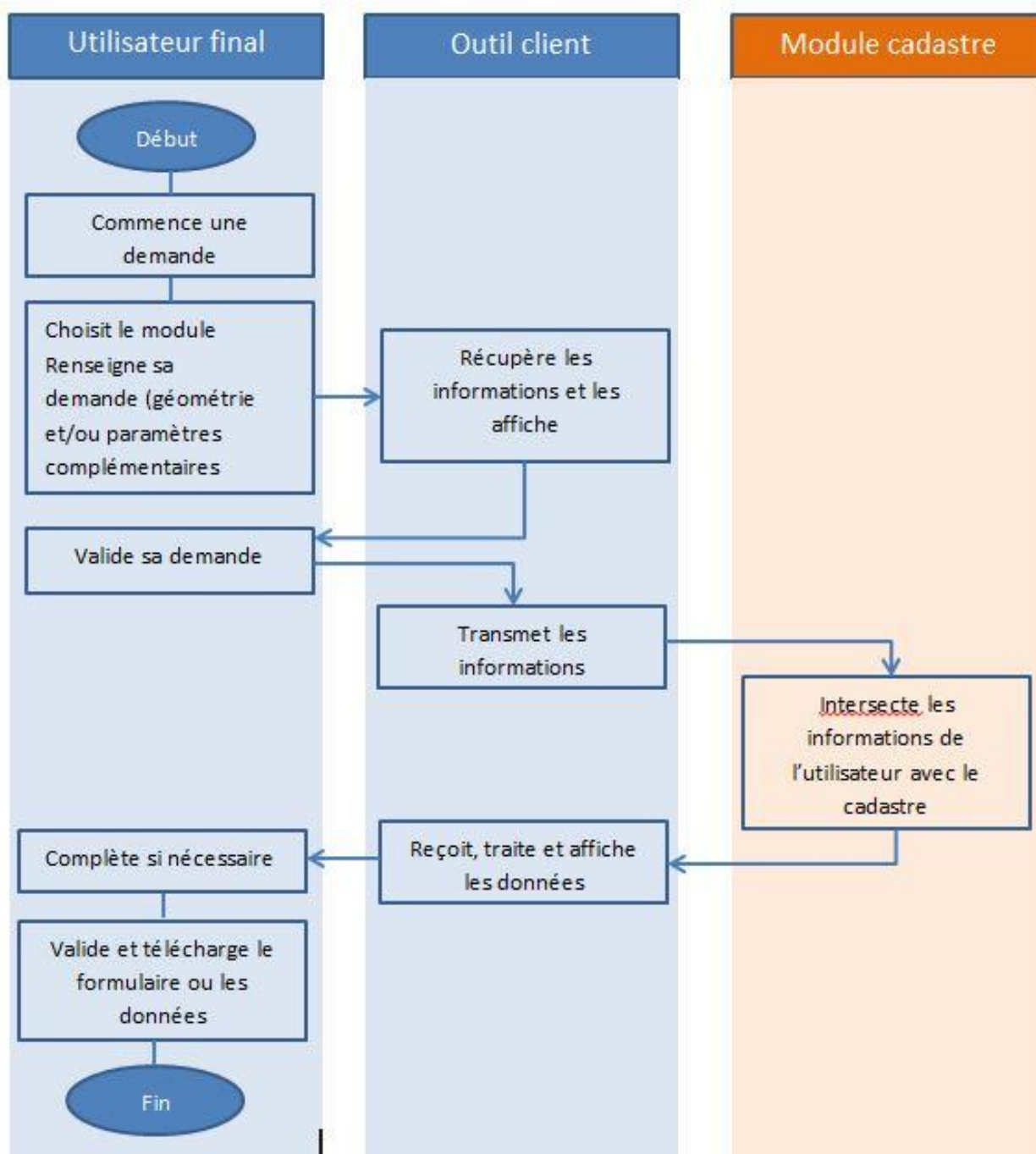


Figure 1 : cas d'utilisation générale du module cadastre

L'utilisation du module cadastre peut être une étape intermédiaire dans la saisie d'un formulaire. Par exemple récupérer la géométrie d'une commune et utiliser la réponse obtenue pour ensuite requêter les données cadastrales au travers de l'API.

La gestion d'appels multiples d'API Carto doit être faite côté client.

Le module cadastre d'API Carto récupère les données cadastrales qui correspondent aux informations saisies par l'utilisateur final.

Ces données peuvent être de différentes natures : commune, division/feuille, parcelle ou localisant. IL convient d'utiliser le sous-module correspondant.

Les informations du cadastre disponibles satisfaisant aux informations transmises 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 cadastre de l'API Carto est disponible à cette adresse : <https://apicarto.ign.fr/api/doc/cadastre>.

I.3 Fonctionnement général du module cadastre d'API Carto

Le fonctionnement du module cadastre d'API Carto est soumis à la possession d'une clé valide de Géoservice du géoportail de l'IGN ouvrant le droit d'utiliser la couche PCI Express ou BD Parcellaire choisie en paramètre de la requête.

Comme écrit ci-dessus, il est possible d'interroger les données cadastrales du PCI Express ou de la BD Parcellaire.

Les requêtes peuvent se faire en POST ou en GET.

L'outil client doit envoyer une géométrie au format GeoJSON au module cadastre.

Le module cadastre récupère parmi les données cadastrales disponibles les objets correspondant aux valeurs saisies dans les paramètres intersectant la géométrie fournie en entrée et les renvoie à l'outil client sous forme de géométrie au format GeoJSON (avec les attributs).

I.4 Modes d'interrogation des données disponibles

Les données cadastrales sont interrogeables par type d'objet.

Il n'est pas autorisé de saisir plusieurs valeurs dans un paramètre mais plusieurs paramètres peuvent être remplis. Par exemple une requête sur deux communes devra être faite en deux étapes, une par commune. Le paramétrage des requêtes devra être fait en amont par le client qui devra aussi assurer le collationnement des résultats en aval.

Le paramètre *apikey* est disponible pour tous les types d'objet (Cf. § I.5.a ci-dessous).

Toutes les données sont interrogeables au travers d'une géométrie. (Cf. § I.5.b ci-dessous)

Pour tous les types d'objet, deux paramètres optionnels supplémentaires permettent de restreindre la réponse (Cf. § I.5.c 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.a Commune

Le sous-module renvoie la géométrie des communes correspondant aux critères saisis.

- *code_insee* : filtre les réponses sur le code INSEE saisi
- *code_dep* : filtre les réponses sur le code département saisi
- *source_ign* : a deux valeurs possibles, vide ou PCI. Dans le premier cas c'est la BD Parcellaire qui est interrogée, dans le second c'est le PCI Express. En plus de l'actualité différente des produits, la géométrie des objets est légèrement différente selon la source des données.

I.4.b division/feuille

La structure des données de la BD Parcellaire et du PCI Express présentent une différence sémantique dans l'appellation des planches cadastrales. La BD Parcellaire utilise la classe d'objets

DIVCAD et le PCI express la classe d'objets FEUILLE.

Ceci conduit à l'existence de deux modules distincts selon la source de données requêtée.

- DIVCAD pour requêter la BD Parcellaire
- FEUILLE pour requêter le PCI Express

Malgré cette différence sémantique, les données sont structurellement les mêmes, les paramètres de filtrage des résultats sont identiques dans les deux modules :

- *code_insee* : filtre les réponses sur le code INSEE saisi
- *code_dep* : filtre les réponses sur le code département saisi
- *code_com* : filtre les réponses sur le code commune saisi (note : le code commune correspond au 3 derniers chiffres du code INSEE de la commune)
- *section* : filtre les réponses sur le code de la section cadastrale saisi (sur 2 caractères)
- *code_arr* : filtre les réponses sur le code arrondissement saisi (concerne les arrondissements urbains Paris, Lyon et Marseille)

I.4.c Parcelle

- *code_insee* : filtre les réponses sur le code INSEE saisi
- *section* : filtre les réponses sur le code de la section cadastrale saisi (sur 2 caractères)
- *numero* : filtre les réponses sur le numéro de parcelle saisi (sur 4 caractères)
- *com_abs* : filtre les réponses sur le code de commune absorbée
- *code_arr* : filtre les réponses sur le code arrondissement saisi (concerne les arrondissements urbains Paris, Lyon et Marseille)
- *source_ign* : a deux valeurs possibles, vide ou PCI. Dans le premier cas c'est la BD Parcellaire qui est interrogée, dans le second c'est le PCI Express. En plus de l'actualité différente des produits, la géométrie des objets est légèrement différente selon la source des données.

I.4.d localisant

- *code_insee* : filtre les réponses sur le code INSEE saisi
- *section* : filtre les réponses sur le code de la section cadastrale saisi (sur 2 caractères)
- *numero* : filtre les réponses sur le numéro de parcelle saisi (sur 4 caractères)
- *code_arr* : filtre les réponses sur le code arrondissement saisi (concerne les arrondissements urbains Paris, Lyon et Marseille)
- *source_ign* : a deux valeurs possibles, vide ou PCI. Dans le premier cas c'est la BD Parcellaire qui est interrogée, dans le second c'est le PCI Express. En plus de l'actualité différente des produits, la géométrie des objets est légèrement différente selon la source des données.

I.5 Description des paramètres à fournir en entrée

I.5.a 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 BD Parcellaire ou PCI Express** requêtées :

BDPARCELLAIRE-VECTEUR_WLD_BDD_WGS84G:commune
BDPARCELLAIRE-VECTEUR_WLD_BDD_WGS84G:divcad
BDPARCELLAIRE-VECTEUR_WLD_BDD_WGS84G:localisant
BDPARCELLAIRE-VECTEUR_WLD_BDD_WGS84G:parcelle

CADASTRALPARCELS.PARCELLAIRE_EXPRESS:commune
CADASTRALPARCELS.PARCELLAIRE_EXPRESS:feuille
CADASTRALPARCELS.PARCELLAIRE_EXPRESS:parcelle
CADASTRALPARCELS.PARCELLAIRE_EXPRESS:localisant

La clé doit être sécurisée par `referer`. La sécurisation est à paramétrer dans l'espace client du site geoservice.ign.fr

Le nom de la clé doit être rempli dans le paramètre `apikey`.

I.5.b Requête par géométrie

Tous les modules peuvent être interrogés par géométrie.

Ce paramètre peut être laissé vide. Toutefois, pour être valide, une requête doit inclure au moins un filtrage.

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 parcelle :

```
{"type":"MultiPolygon",  
  "coordinates":[[[[[2.4175293,48.8478826],[2.4175268,48.8478081],[ 2.4173646,48.8478107],[2.4173668,48.8478845],[2.4175293,48.8478826]]]]]
```

Exemple de géométrie JSON pour un localisant ponctuel :

```
{"type":"Point",  
  "coordinates":[2.120705, 44.168907]
```

L'interrogation du module cadastre 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.5.c 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.5.d 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": "divcad.2200981",    "geometry": {
        "type": "MultiPolygon",
        "coordinates": [...]
      },
      "geometry_name": "the_geom",
      "properties": {
        "feuille": 1,
        "section": "ZE",
        "code_dep": "14",
        "nom_com": "Fontaine-le-Pin",
        "code_com": "276",
        "com_abs": "000",
        "echelle": "2000",
        "edition": 4,
        "code_arr": "000",
        "bbox": [
          -0.30936609,
          48.95826856,
          -0.28291876,
          48.97010297
        ],
        "code_insee": "14276"
      }
    }
  ], "totalFeatures": 17,
  "numberMatched": 17,
  "numberReturned": 17,
  "timeStamp": "2020-04-02T10:03:03.561Z",
  "crs": {
    "type": "name",
    "properties": {
      "name": "urn:ogc:def:crs:EPSG::4326"
    }
  },
  "bbox": [
    -0.30265543,
    48.95023487,
    -0.271503,
    48.96577189
  ]
}
```

```
}
```

II Conseils d'utilisation du module cadastre

Cette partie a vocation à préciser quelques spécificités de l'utilisation du module cadastre.

II.1 Paramétrage de la ressource

Le module cadastre d'API Carto permet d'interroger deux sources de données :

- La BD Parcellaire
- Le PCI Express

La **BD Parcellaire** est un produit dont la production est arrêtée. **Son usage n'est plus recommandé car elle n'est plus mise à jour**. A terme il est prévu de retirer l'usage des données BD Parcellaire dont l'obsolescence ne fera que grandir.

L'utilisation du PCI Express est fortement recommandée et deviendra obligatoire.

II.2 Géométrie en entrée de la requête

La taille de la géométrie utilisée pour une requête a une influence sur le délai de réponse, il convient de limiter sa taille au minimum.

i. Récupération de la géométrie en entrée

La géométrie du terrain concerné par la demande de l'utilisateur final doit être choisie côté outil client, par exemple par sélection via les flux WFS BDTOPO, sélection des parcelles et récupération des géométries via le module RPG d'API Carto, saisie avec des outils de dessin vectoriel etc.

Quels que soient les choix de modélisation de l'outil client, la géométrie devra être prétraitée en amont de l'appel à l'API pour assurer qu'elle soit conforme au format attendu, c'est-à-dire :

- valide au sens OGC,
- de type Polygon ou MultiPolygon (ou Point le cas échéant),
- définie en projection WGS84,
- au format JSON.

ii. Cas des demandes sur plusieurs géométries

Si la modélisation de l'outil client permet à l'utilisateur final de faire une demande portant sur plusieurs géométries distinctes, l'outil client devra également gérer :

- soit le prétraitement des géométries (union par exemple) pour qu'elles soient conformes en amont de l'appel à l'API ;
- soit l'enchaînement des requêtes envoyées à l'API (un appel par géométrie), ainsi que la récupération et le stockage des résultats renvoyés par chaque requête, puis le traitement de l'ensemble des résultats renvoyés pour les mettre en forme selon les besoins de l'utilisateur final.

II.3 Paramétrage des requêtes avec plusieurs valeurs d'un même paramètre

Si l'outil client souhaite récupérer les résultats correspondant à **plusieurs valeurs d'un même paramètre**, il devra enchaîner plusieurs requêtes (une par valeur), récupérer et stocker les résultats

renvoyés par chaque requête, puis traiter l'ensemble des résultats renvoyés par l'API pour les mettre en forme selon les besoins de l'utilisateur final.

II.4 Requêtes avec plus de 1000 réponses

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

Pour une requête avec plus de 1000 objets répondant aux critères, il est possible récupérer les objets en faisant des appels successifs à l'API en implémentant le paramètre `_start`.

Le paramétrage des requêtes devra être fait en amont par le client qui devra aussi assurer le collationnement des résultats en aval.

Exemple :

```
curl -X GET "https://apicarto.ign.fr/api/cadastre/parcalle?code_insee=94067" -H "accept: application/json"
curl -X GET "https://apicarto.ign.fr/api/cadastre/parcalle?code_insee=94067&_start=1000" -H "accept: application/json"
curl -X GET "https://apicarto.ign.fr/api/cadastre/parcalle?code_insee=94067&_start=2000" -H "accept: application/json"
...
```

II.5 Détection des résultats vides

i. Erreur de la requête

Dans le cas où la requête contient une erreur, l'outil client recevra un code d'erreur 400, et la réponse contiendra le message de l'erreur rencontrée dans `msg`.

Code 400, Error: Bad Request

```
{
  "code": 400,
  "message": {
    "geom": {
      "location": "query",
      "param": "geom",
      "value": "{\"type\":\"MultiPolygon\",\"coordinates\":[[[-1.6993786,48.1113366],[-1.6994647,48.1113416],[-1.6994613,48.1113573],[-1.6993639,48.111803],[-1.6992707,48.112222],[-1.6990176,48.1120599],[-1.6989945,48.1120573],[-1.6991084,48.111617],[-1.6991262,48.1115482],[-1.6993407],
      \"msg\": \"Parse error on line 1:\\n...1115482],[-1.6993407\\n-----^\\nExpecting ',', ']', got 'EOF'\"
    }
  }
}
```

Attention, ces erreurs portent bien sur le format de la requête et non pas sur son bien-fondé. Si une requête est réalisée sur une couche avec un paramètre non disponible (ex.code_insee ne correspondant à aucune commune) mais respectant la syntaxe attendue, l'outil client recevra un code de succès et mais aucun objet ne correspondant à la requête, le nombre d'objet renvoyé sera nul.

```
{
  "type": "FeatureCollection",
  "features": [],
  "totalFeatures": 0,
  "numberMatched": 0,
  "numberReturned": 0,
}
```

```
"timeStamp": "2020-04-02T11:11:43.575Z",  
"crs": null  
}
```

Dans le cas où la connexion avec l'API rencontre une erreur, l'outil client recevra un code d'erreur 500, et la réponse contiendra le **message** de l'erreur rencontrée.

Code 500, Error: Internal Server Error

```
{  
  "type": "error",  
  "message": ".....",  
  "featureType": "....."  
}
```

ii. Résultat de la requête vide

Dans le cas où la requête ne renvoie aucun résultat, sans toutefois rencontrer d'erreur, l'outil client recevra un code « succès » mais pourra détecter l'absence de résultat par un **totalFeatures = 0**.

Code 200, Success

Exemple de résultat vide renvoyé par une requête :

```
{  
  "type": "FeatureCollection",  
  "features": [],  
  "totalFeatures": 0,  
  "numberMatched": 0,  
  "numberReturned": 0,  
  "timeStamp": "2020-04-02T11:11:43.575Z",  
  "crs": null  
}
```

iii. Attributs non renseignés

Certains attributs peuvent apparaître vides ou nuls dans la réponse, signifiant qu'ils sont non renseignés dans les données cadastrales requêtées.