Documentation utilisateurs
API Carto module RPG

Version 1.0

SOMMAIRE

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
RPG	Registre Parcellaire Graphique, base de données géographiques servant de référence à l'instruction des aides de la politique agricole commune (PAC).

I Description générale du module RPG 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, 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 RPG d'API Carto

Le module RPG d'API Carto permet d'obtenir des informations du registre parcellaire graphique d'un millésime donné intersectant une géométrie (ponctuelle ou surfacique) et/ou correspondant à un code culture. L'API fait appel à des ressources WFS du géoportail de l'IGN.

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

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

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

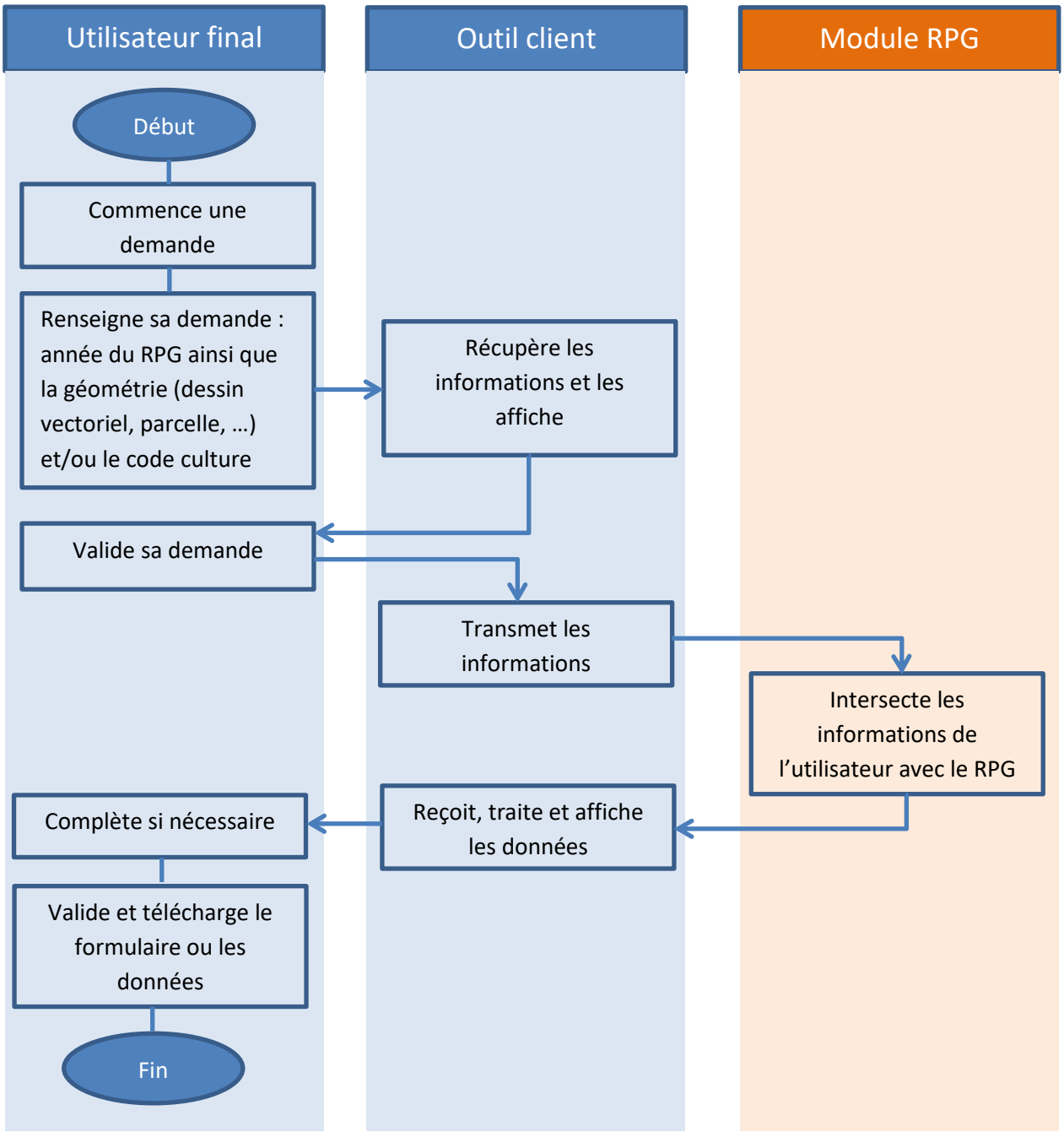

Figure 1 : cas d'utilisation générale du module RPG

Le module RPG d'API Carto récupère les données du RPG qui correspondent aux informations saisies par l'utilisateur final :

- l'année du RPG (obligatoire)
- une géométrie (facultative)
- un code culture (facultatif)

Note : il faut a minima une géométrie ou un code culture.

Les informations du RPG 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 RPG de l'API Carto est disponible à cette adresse : https://apicarto.ign.fr/api/doc/RPG.

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

Le fonctionnement du module RPG 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 RPG du millésime choisi en paramètre de la requête.

Les différences de structure des données entre les versions 1 et 2 du RPG conduisent à proposer deux sous modules, un pour la version 1 (de 2010 à 2013) et un pour la version 2 (à partir de 2015).

Les requêtes peuvent se faire en POST ou en GET.

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

Le module RPG récupère parmi les données RPG disponibles les objets correspondant aux critères (année et code culture) 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 du RPG sont interrogeables par millésime. Il existe deux modules, un pour les millésimes correspondant à la version 1 du RPG (années 2010 à 2013) et un pour les millésimes correspondant à la version 2 à partir de 2015. L'année doit être précisée dans la requête.
Le paramètre *apikey* est disponible pour les deux modules (Cf. § I.5.a ci-dessous).
Il y a deux paramètres disponibles pour interroger le RPG d'un millésime dont un au moins doit être précisé :

- Une géométrie (ponctuelle ou surfacique) (Cf. § I.5.c ci-dessous)
- *Code_cult* : filtre les réponses renvoyées par le code culture saisi (Cf. § I.5.d ci-dessous)

Deux paramètres optionnels supplémentaires permettent de restreindre la réponse (Cf. § I.5.e 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.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 RPG** des millésimes utilisés :

RPG 2010
RPG 2011
RPG 2012
RPG 2013
RPG 2014:ilots anonymes (à venir)
RPG 2015:parcelles graphiques
RPG 2016:parcelles graphiques
RPG 2017 parcelles graphiques
RPG 2018 parcelles graphiques
RPG 2019 parcelles graphiques

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 Millésime

L'indication du millésime du RPG à utiliser est **obligatoire**. Sa valeur à renseigner dans le champ *annee* doit être de 2010 à 2013 pour la version 1 du RPG et à partir de 2015 pour la version 2 du RPG.
Note : pour raison technique liée à la ressource, le millésime 2014 ne fonctionne pas correctement actuellement. Un correctif sera apporté.

I.5.c Requête par géométrie

Tous les millésimes du RPG (à partir de 2010) peuvent être interrogés par géométrie.

Ce paramètre peut être laissé vide. Toutefois, pour être valide, une requête doit inclure au moins un filtrage sur la géométrie ou le code culture.

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":"MutiPolygon",
  "coordinates":[[[[2.4175293,48.8478826],[2.4175268,48.8478081],[ 2.4173646,4
8.8478107],[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 RPG 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.d Requête sur le code culture

Tous les millésimes du RPG (à partir de 2010) peuvent être interrogés par code culture.

Ce paramètre peut être laissé vide. Toutefois, pour être valide, une requête doit inclure au moins un filtrage sur la géométrie ou le code culture.

Il est possible de filtrer la sélection sur le code culture. Ce code culture doit correspondre aux spécifications du millésime requêté.

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

I.5.e 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.f 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": "rpg_2013.725984",
   "geometry": {
    "type": "MultiPolygon",
    "coordinates": […]
 },
   "geometry_name": "the_geom",
   "properties": {
    "num_ilot": "014-725851",
    "commune": "14720",
    "forme_juri": "EARL",
    "surf_decla": "138.08",
    "dep_rattac": "014",
    "surf_graph": "4.29",
    "surf_cultu": "4.29",
    "code_cultu": "18",
    "nom_cultu": "PRAIRIES PERMANENTES",
    "bbox": [
     -0.29430389,
     48.95340533,
     -0.2893747,
     48.95561391
    ],
```

```
      "code_insee": null
    }
   },
  { …
  },
"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 RPG

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

II.1 Paramétrage du millésime

Le module RPG d'API Carto ne requête qu'un millésime à la fois. Pour obtenir des informations sur deux millésimes différents il faut enchaîner les requêtes et traiter le résultat côté client.
Les changements intervenus dans la structure des données dans la version 2 des spécifications du produit, à partir du millésime 2015, entraine l'utilisation de deux sous-modules spécifiques.

II.2 Géométrie en entrée de la requête

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 Cadastre 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'enchainement 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 codes cultures

Si l'outil client souhaite récupérer les résultats correspondant à **plusieurs codes cultures**, il devra enchaîner plusieurs requêtes (une par code culture), 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.

Pour le RPG version 1 la valeur à saisir correspond au nombre contenu dans l'attribut **CODE_CULTU de la classe ILOTS_ANONYMES** avec une valeur allant de 01 à 28 (le 0 est obligatoire pour les valeurs inférieures à 10). La liste des codes cultures peut être consultée dans le descriptif de contenu du RPG version 1.0.

Pour le RPG version 2 la valeur à saisir correspond trigramme contenu dans l'attribut **CODE_CULTU de la classe PARCELLES_GRAPHIQUES** en respectant la casse (majuscule). La liste des codes cultures peut être consultée dans le descriptif de contenu du RPG version 2.0.

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/rpg/v1?annee=2013&code_cultu=02" -H "accept: application/json"
curl -X GET "https://apicarto.ign.fr/api/rpg/v1?annee=2013&code_cultu=02&_start=1000" -H "accept: application/json"
curl -X GET "https://apicarto.ign.fr/api/rpg/v1?annee=2013&code_cultu=02&_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. /rpg/v1 ?code_cultu=PRL) 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 du millésime RPG requêté.