

happign



IGN

Sommaire

happign, pourquoi ?

Flux WFS

Flux WMS

APIcarto

LIDAR

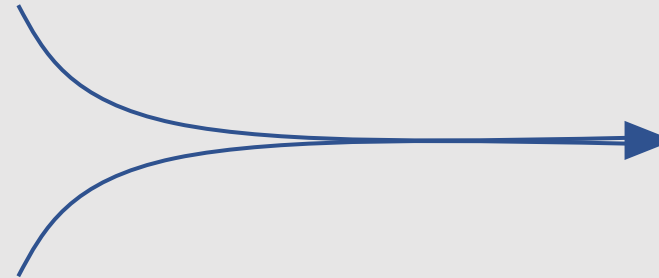
happign, pourquoi ?

PROBLEMES :

- Passage par du téléchargement direct :
 - Source de données pas toujours facile à trouver ;
 - Paquets de données parfois non cohérents avec les besoins ;
 - Mise à jour fastidieuse
- Utilisation très limitée de QGIS

BESOINS :

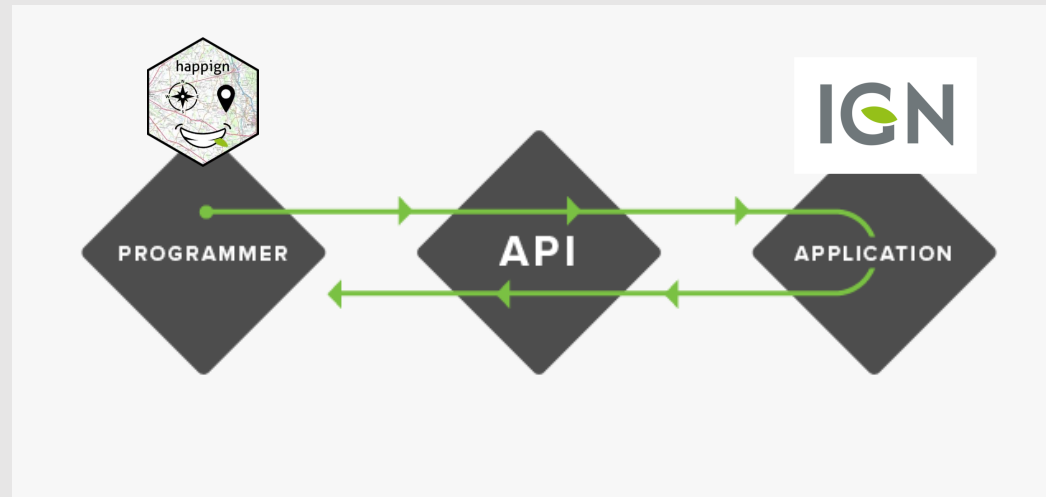
- Être plus agile avec l'utilisation des données :
 - Besoin de comparer facilement différentes données ;
 - Besoin d'utiliser les rasters autrement que pour de la visualisation ;
 - Besoin d'optimisation ;
 - Besoins de reproductibilité



Une API c'est quoi ?

API :

Une API c'est une interface de programmation qui permet de rendre disponibles les données ou les fonctionnalités d'une application existante afin que d'autres applications les utilisent



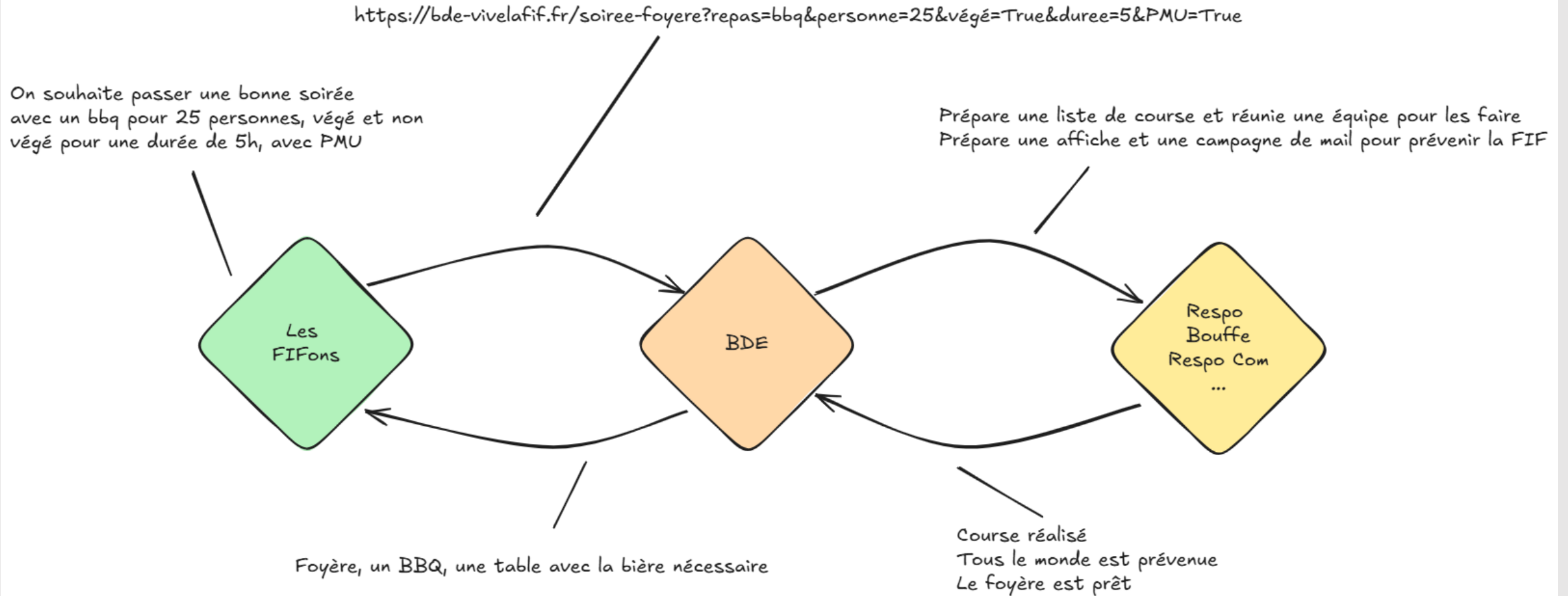
<https://geo.api.gouv.fr/communes?codePostal=78000>

Adresse

Chemin
d'accès

requête

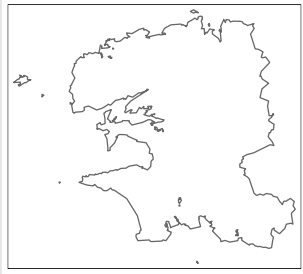
Une API c'est quoi ?



Structure des flux IGN



**Zone
Géographique**



Package sf

**Type de
flux**

WFS (vecteur)

WMS (raster)

WMTS (raster)

**get_wfs()
get_wms_raster()
get_wmts()**

Nom de couche

*LIMITES_ADMINISTRATI
VES_EXPRESS.LATEST:ca
nton*

get_layers_metadata()

Flux WFS

Manuel

[Services web "experts" | Géoservices \(ign.fr\)](#)



Récupération
des
noms
de
couches

➤ `get_layers_metadata("wfs« , "administratif")`

Name	Title	Abstract
ADMINEXPRESS-COG-CARTO.LATEST:arrondissement_municipal	ADMINEXPRESS-COG-CARTO.LATEST:arrondissement_municipal.title	édition 2023
ADMINEXPRESS-COG-CARTO.LATEST:canton	ADMINEXPRESS-COG-CARTO.LATEST:canton.title	édition 2023
ADMINEXPRESS-COG-CARTO.LATEST:chflieu_arrondissement_municipal	ADMINEXPRESS-COG-CARTO.LATEST:chflieu_arrondissement_municipal.title	édition 2023
ADMINEXPRESS-COG-CARTO.LATEST:chflieu_commune	ADMINEXPRESS-COG-CARTO.LATEST:chflieu_commune.title	édition 2023
...

Récupération
des
catégories

➤ `get_apikeys()`

```
c("administratif", "adresse", "agriculture", "altimetrie", "cartes",  
  "cartovecto", "clc", "economie", "environnement", "geodesie",  
  "lambert93", "ocsge", "ortho", "orthohisto", "parcellaire",  
  "satellite", "sol", "topographie", "transports")
```

Flux WFS



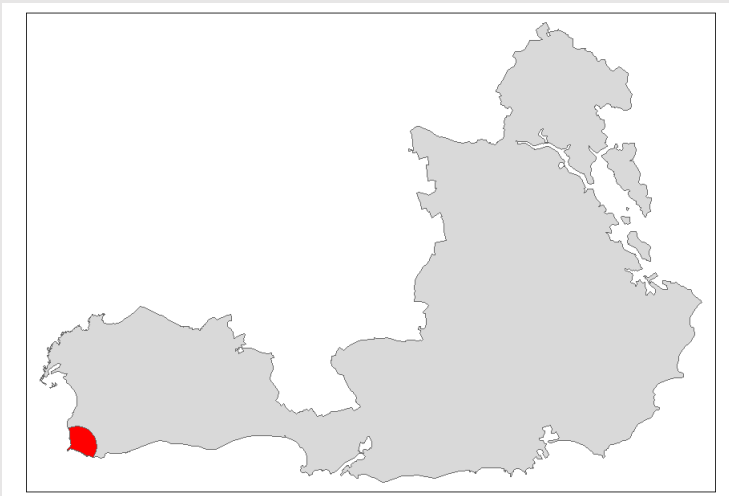
Récupération
d'une
shape

```
➤ penmarch <- read_sf(system.file("extdata/penmarch.shp", package = "happign"))
```



Récupération
des
données

```
➤ get_wfs(x = penmarch,  
          layer = "ADMINEXPRESS-COG-CARTO.LATEST:arrondissement")
```



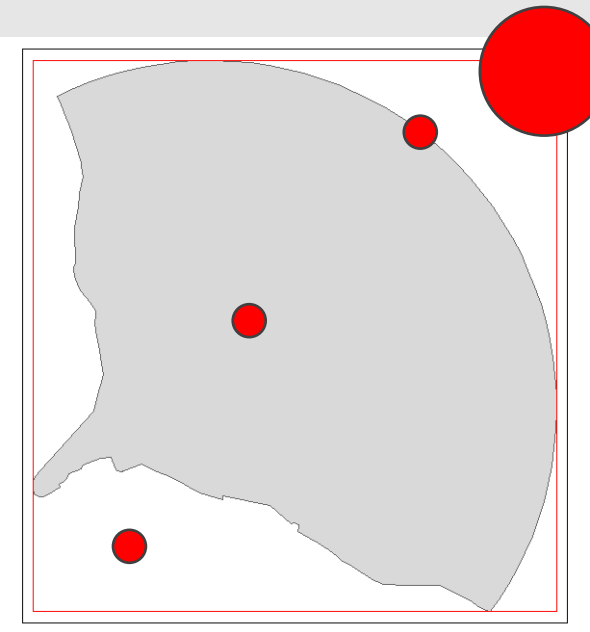
id	nom_m	nom	insee_arr	insee_dep	insee_reg	geometry
ARR_DEP_FXX_00000000099	QUIMPER	Quimper	22	29	53	MULTIPOLYGON ((-4.38082 47...

Flux WFS : paramètres optionnels

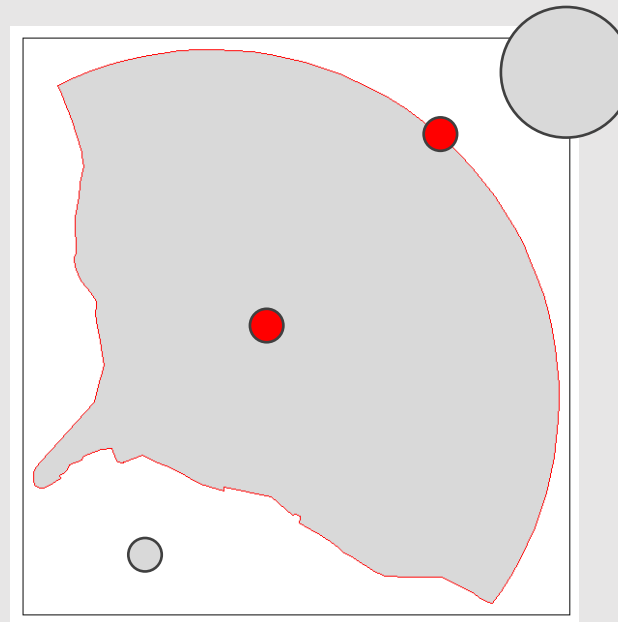
Filtre
spatial

```
➤ get_wfs(x = "penmarch",  
          layer = "ADMINEXPRESS-COG-CARTO.LATEST:canton",  
          spatial_filter = "bbox")
```

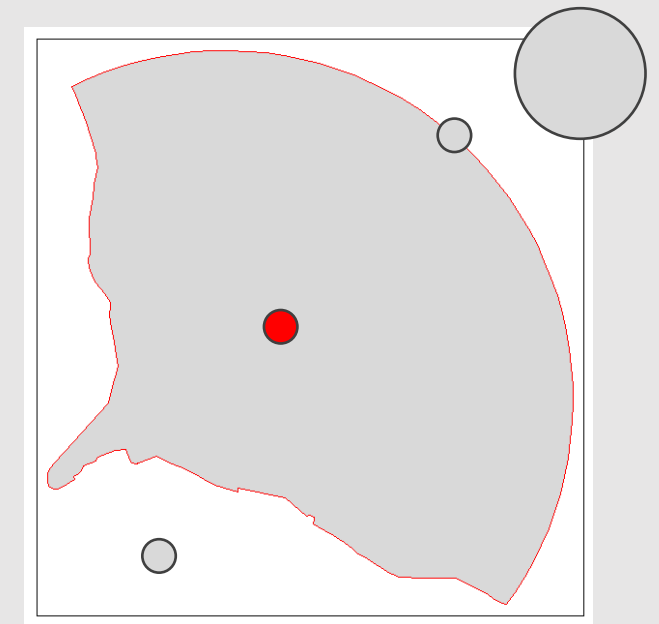
`spatial_filter = "bbox"`



`spatial_filter = "intersects"`



`spatial_filter = "within"`

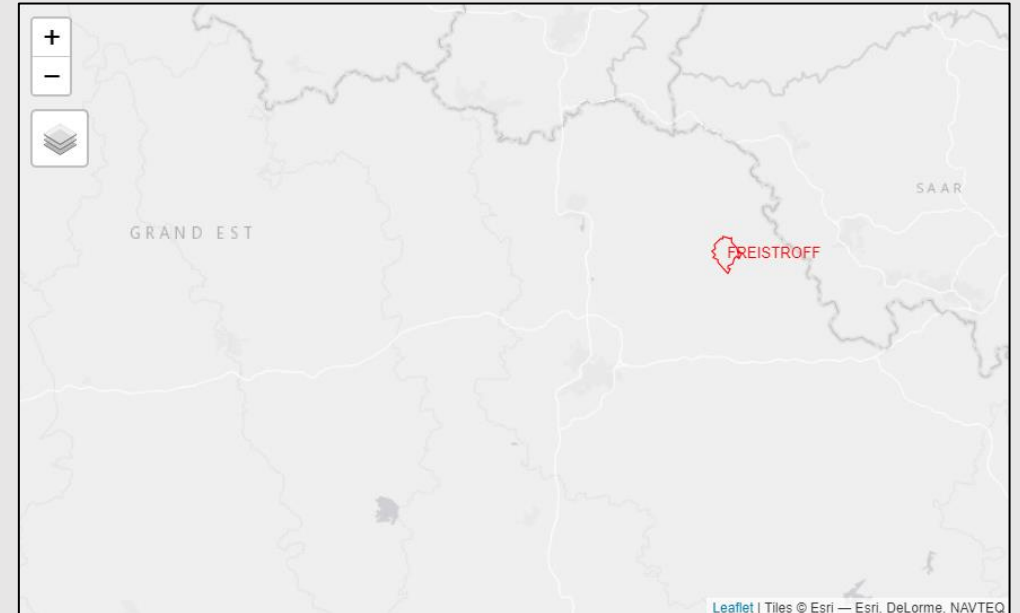


Flux WFS : paramètres optionnels

[Filtre ECQL](#)

```
➤ get_wfs(x = NULL,  
  layer = "LIMITES_ADMINISTRATIVES_EXPRESS.LATEST:commune",  
  ecql_filter = "nom_m LIKE 'F%I%F'")
```

```
ecql_filter = "nom_m LIKE 'F%I%F%' AND population > 1000")
```



Utilitaires : `get_wfs_attributes("administratif", "LIMITES_ADMINISTRATIVES_EXPRESS.LATEST:commune")`

`=> c("id", "nom", "nom_m", "insee_com", "statut", "population", "insee_can", "insee_arr", "insee_dep", "insee_reg", "siren_epci")`

Flux WFS : paramètres optionnels

Session
interactive

➤ `get_wfs(shape = penmarch,
interactive = TRUE)`

1: administratif	2: adresse	3: agriculture	4: altimétrie
5: cartes	6: cartovecto	7: clc	8: économie
9: environnement	10: géodésie	11: lambert93	12: ocsge
13: ortho	14: orthohisto	15: parcellaire	16: satellite
17: sol	18: topographie	19: transports	

➤ **Sélection : 4**

1: ELEVATION.CONTOUR.LINE:courbe
2: ELEVATIONGRIDCOVERAGE.HIGHRES.QUALITY:source_fra

➤ **Sélection : 1**

id	altitu de	nat_topo	importance	geometry
ISOHYPSE0120680000000311	10	1	0	LINESTRING (-4.223159 47.79...
ISOHYPSE0120680000000467	5	2	0	LINESTRING (-4.36668 47.800...
ISOHYPSE0120680000000540	5	2	0	LINESTRING (-4.373909 47.80...
ISOHYPSE0120680000000660	0	1	1	LINESTRING (-4.222427 47.79...
ISOHYPSE0120680000000665	5	1	0	LINESTRING (-4.222553 47.79...
ISOHYPSE0120680000000724	15	1	0	LINESTRING (-4.227225 47.82...
ISOHYPSE0120680000000741	5	2	0	LINESTRING (-4.364432 47.79...
ISOHYPSE0120680000000963	5	2	0	LINESTRING (-4.363637 47.80.

Flux WMS

Récupération
des
noms
de
couches

➤ `get_layers_metadata("altimetrie", "wms")`

name	title	abstract
ELEVATION.CONTOUR.LINE	Courbes de niveau	Le produit « Courbes de niveau » est un modèle numérique de terrain sous forme de courbes de ...
ELEVATION.ELEVATIONGRIDCOVERAGE	MNT BDAlti V1	BDAlti V1 au pas de 25m
ELEVATION.ELEVATIONGRIDCOVERAGE.HIGHRES	Modèle Numérique de Terrain issu du RGEALTI	Modèle Numérique de Terrain issu du RGEALTI
ELEVATION.ELEVATIONGRIDCOVERAGE.HIGHRES.MNS	Modèle numérique de surface	Modèle numérique de surface (MNS)
ELEVATION.ELEVATIONGRIDCOVERAGE.HIGHRES.MNS.SHADOW	Estompage appliqué au Modèle Numérique de Surface	Modèle Numérique de Surface (MNS) avec un style par défaut

Récupération
des
catégories

➤ `get_apikeys()`

```
c("administratif" "adresse" "agriculture" "altimetrie" "cartes"
  "cartovecto" "clc" "economie" "environnement" "geodesie"
  "lambert93" "ocsge" "ortho" "orthohisto" "parcellaire"
  "satellite" "sol" "topographie" "transports")
```

Flux WMS

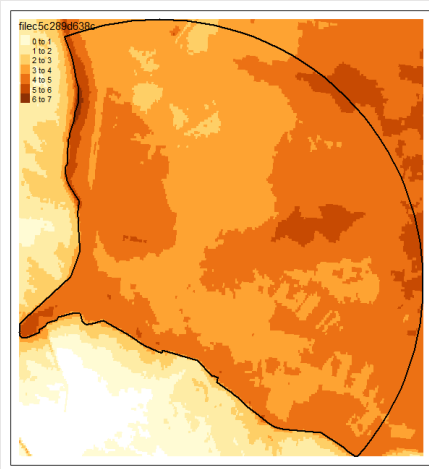
Récupération
d'une
shape

```
➤ penmarch <- read_sf(system.file("extdata/penmarch.shp", package = "happign"))
```



Récupération
des
données

```
➤ get_wms_raster(x = penmarch,  
  apikey = "altimetrie" ,  
  layer = "ELEVATION.ELEVATIONGRIDCOVERAGE.HIGHRES",  
  res = 1)
```



Remarques :

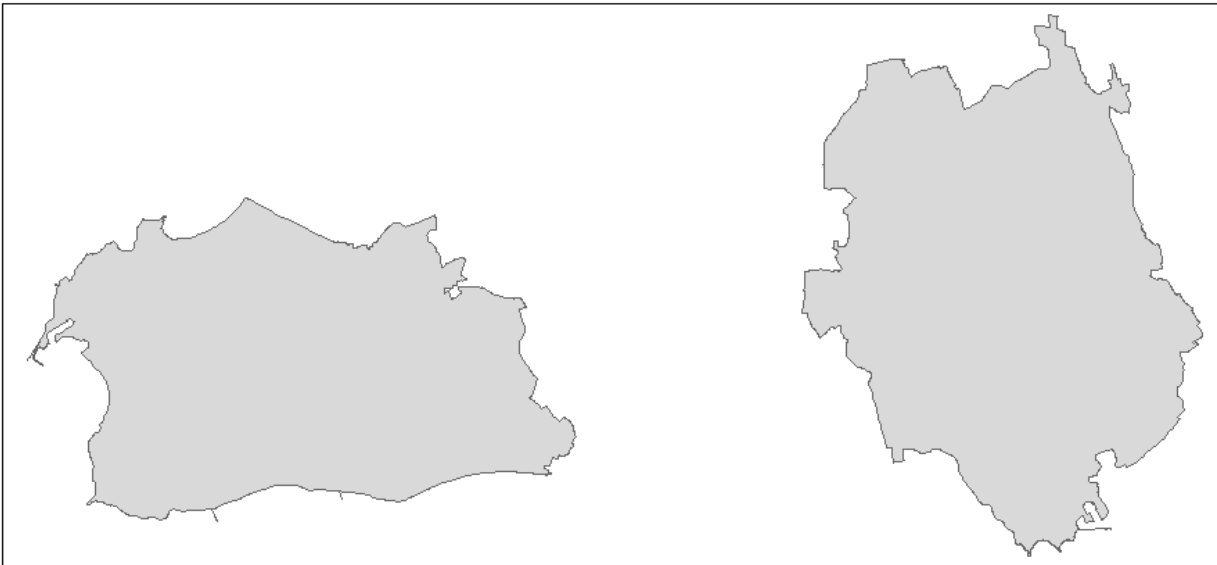
- Le paramètre *res* doit être choisie au regard de la résolution réelle du la donnée
- Les rasters sont toujours téléchargés en local, par défaut dans un dossier temporaire

API Carto

Les API Carto sont mise à disposition par l'IGN pour faciliter le remplissage de formulaire. Elles ont l'avantages de ne pas nécessairement avoir besoin d'une shape en entrée. happign permet de se connecter à ces APIs à l'aide des fonctions ***get_apicarto_****. Toutes ces fonctions sont vectorisées, ainsi chaque paramètre peut prendre plusieurs valeurs :

get_apicarto_cadastre : parcelles cadastrales, communes, feuille, division et localisant à partir du code insee, code département ou d'une shape.

➤ `get_apicarto_cadastre(c("29158", "29165"), type = "commune")`



API Carto

get_apicarto_codes_postaux : code insee et nom d'une commune à partir du code postal

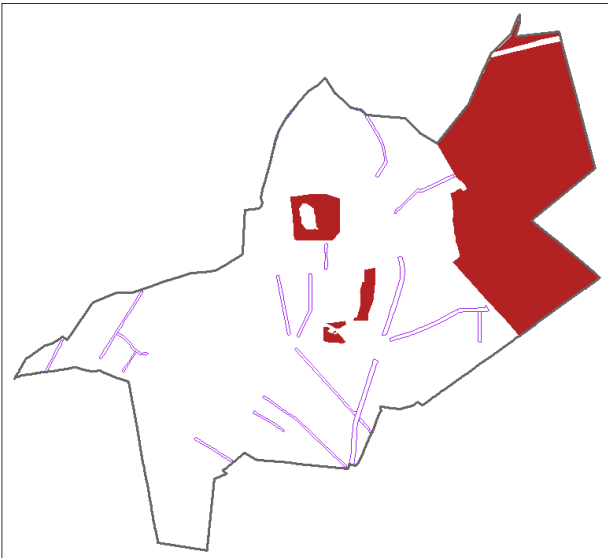
➤ `get_apicarto_codes_postaux(c("29760", "29260"))`

codePostal	codeCommune	nomCommune	libelleAcheminement
29760	29158	Penmarch	PENMARCH
29260	29055	Le Folgoët	LE FOLGOET
29260	29093	Kernilis	KERNILIS
29260	29094	Kernouës	KERNOUES
29260	29100	Lanarvily	LANARVILY
29260	29124	Lesneven	LESNEVEN
29260	29126	Loc-Brévalaire	LOC BREVALAIRE
29260	29179	Ploudaniel	PLOUDANIEL
29260	29198	Plouider	PLOUIDER
29260	29248	Saint-Frégant	SAINT-FREGANT
29260	29255	Saint-Méen	SAINT-MEEN
29260	29288	Trégarantec	TREGARANTEC

API Carto

get_apicarto_gpu: API connectée au géoportail de l'urbanisme permettant de récupérer divers documents d'urbanisme, prescriptions et servitudes d'utilités publiques (Cône de vue, Alignement d'arbre, Canalisations de transport de gaz, Espace boisé classé, Carrières, Périmètres monuments historique, ...)

- `prescriptions <- get_apicarto_gpu("DU_93014", ressource = c("prescription-surf", "prescription-lin"))`
- `ebc <- prescriptions[prescriptions$libelle == "EBC",]`
- `align_arbre <- prescriptions[prescriptions$libelle == "Alignement d'arbre",]`
- `com <- get_apicarto_cadastre("93014", "commune")`



Alignement d'arbre

Espace boisé classé

API Carto

`get_apicarto_rpg`: Registre parcellaire graphique

- `penmarch <- get_apicarto_cadastre("29158", type = "commune")`
- `rpg <- get_apicarto_rpg(penmarch, 2020, dTolerance = 10)`

Remarque : Les APIcarto ne supportent pas les géométries trop complexes. Le paramètre `dTolerance`, présent dans chaque fonction `get_apicarto_*` permet de simplifier la shape à la volée.



Idées d'utilisation



A découvrir sur la vignette : [happign for forester](#)

- Calcul du MNH ;
- Détection automatique des sites écologique ;
- Récupération des cartes historique ;
- Calcul du pourcentage d'essence avec la bdforet ;
- Récupération des images satellites ;
- Calcul d'indice à partir des images satellites ;
- Récupération du cadastre,
- Calcul d'isochrone et d'isodistance.

Autres :

- Vignette [API carto](#)

Liens utiles

Site happign : <https://paul-carteron.github.io/happign/index.html>

Répertoire github : <https://github.com/paul-carteron/happign>

Géoservice web expert : <https://geoservices.ign.fr/services-web-experts>

Documentation géoservice : <https://geoservices.ign.fr/documentation>

Documentation API carto : <https://apicarto.ign.fr/api/doc/>



Merci pour votre attention !