Package: CatastRoNav (via r-universe)

July 9, 2024

Title Interface to the 'INSPIRE' services of 'Catastro de Navarra'

Version 0.1.0

Description Access public spatial data available under the 'INSPIRE' directive. Tools for downloading references, buildings and addresses of properties on Navarre (Spain).

License CC BY 4.0

URL https://ropenspain.github.io/CatastRoNav/, https://github.com/rOpenSpain/CatastRoNav

BugReports https://github.com/rOpenSpain/CatastRoNav/issues

Depends R (>= 3.6)

Imports rappdirs, sf (>= 1.0.0), stringi, tibble, xml2

Suggests CatastRo, dplyr, ggplot2, knitr, mapSpain, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/Needs/website ropenspain/rostemplate, devtools, sessioninfo, remotes, sfheaders, rapidjsonr, jsonify, geometries, tidyverse, magick

Config/testthat/edition 3

Config/testthat/parallel true

Copyright Data Source: SITNA – Catastro de Navarra <https://geoportal.navarra.es/es/inspire>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://ropenspain.r-universe.dev

RemoteUrl https://github.com/rOpenSpain/CatastRoNav

RemoteRef HEAD

RemoteSha 3394cc07222642a56beeb943c791779588026cb7

Contents

`catrnav_atom_get_parcels`

ATOM INSPIRE: Download all the cadastral parcels of a municipality

---

Description

Get the spatial data of all the cadastral parcels belonging to a single municipality using the INSPIRE ATOM service.

Usage

```
catrnav_atom_get_parcels(
  munic,
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

`munic`	Municipality to extract, It can be a part of a string or the cadastral code. See catrnav_atom_get_parcels_db_all.
`cache`	A logical whether to do caching. Default is `TRUE`. See **About caching** section on `catrnav_set_cache_dir()`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source file.
`cache_dir`	A path to a cache directory. On missing value the function would store the cached files on a temporary dir (See `base::tempdir()`).
`verbose`	Logical, displays information. Useful for debugging, default is `FALSE`.

Value

A `sf` object.

References

SITNA – Catastro de Navarra

See Also

Other ATOM: `catrnav_atom_get_parcels_db_all()`

Other parcels: `catrnav_atom_get_parcels_db_all()`, `catrnav_wfs_get_parcels_bbox()`

Examples

```
s <- catrnav_atom_get_parcels("Iruña")

library(ggplot2)

ggplot(s) +
  geom_sf() +
  labs(
    title = "Cadastral Zoning",
    subtitle = "Pamplona / Iruña"
  )
```

---

catrnav_atom_get_parcels_db_all

ATOM INSPIRE: Reference database for ATOM cadastral parcels

Description

Create a database containing the urls provided in the INSPIRE ATOM service for extracting Cadastral Parcels.

Usage

```
catrnav_atom_get_parcels_db_all(
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

cache	A logical whether to do caching. Default is TRUE. See **About caching** section on `catrnav_set_cache_dir()`.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.

cache_dir	A path to a cache directory. On missing value the function would store the cached files on a temporary dir (See base::tempdir()).
verbose	Logical, displays information. Useful for debugging, default is FALSE.

Value

A tibble with the information requested:

- munic: Name of the municipality.
- url: url for downloading information of the corresponding municipality.
- date: Reference date of the data.

Source

SITNA – Catastro de Navarra

See Also

Other ATOM: catrnav_atom_get_parcels()

Other parcels: catrnav_atom_get_parcels(), catrnav_wfs_get_parcels_bbox()

Examples

```
catrnav_atom_get_parcels_db_all()
```

---

catrnav_clear_cache *Clear your* **CatastRoNav** *cache dir*

Description

Use this function with caution. This function would clear your cached data and configuration, specifically:

- Deletes the **CatastRoNav** config directory (rappdirs::user_config_dir("CatastRoNav", "R")).
- Deletes the cache_dir directory.
- Deletes the values on stored on Sys.getenv("CATASTRONAV_CACHE_DIR").

Usage

```
catrnav_clear_cache(config = FALSE, cached_data = TRUE, verbose = FALSE)
```

Arguments

config	if TRUE, will delete the configuration folder of **CatastRoNav**.
cached_data	If this is set to TRUE, it will delete your cache_dir and all its content.
verbose	Logical, displays information. Useful for debugging, default is FALSE.

Details

This is an overkill function that is intended to reset your status as it you would never have installed and/or used **CatastRoNav**.

Value

Invisible. This function is called for its side effects.

See Also

Other cache utilities: `catrnav_set_cache_dir()`

Examples

```
# Don't run this! It would modify your current state
## Not run:
catrnav_clear_cache(verbose = TRUE)

## End(Not run)

Sys.getenv("CATASTRONAV_CACHE_DIR")
```

---

catrnav_set_cache_dir *Set your* **CatastRoNav** *cache dir*

Description

`catrnav_set_cache_dir()` will store your `cache_dir` path on your local machine and would load it for future sessions.

Alternatively, you can store the `cache_dir` manually with the following options:

- Run `Sys.setenv(CATASTRONAV_CACHE_DIR = "cache_dir")`. You would need to run this command on each session (Similar to `install = FALSE`).
- Write this line on your .Renviron file: `CATASTRONAV_CACHE_DIR = "value_for_cache_dir"` (same behavior than `install = TRUE`). This would store your `cache_dir` permanently.

`catrnav_detect_cache_dir()` detects and returns the path to your current `cache_dir`.

Usage

```
catrnav_set_cache_dir(
  cache_dir,
  overwrite = FALSE,
  install = FALSE,
  verbose = TRUE
)

catrnav_detect_cache_dir(...)
```

Arguments

cache_dir	A path to a cache directory. On missing value the function would store the cached files on a temporary dir (See base::tempdir()).
overwrite	If this is set to TRUE, it will overwrite an existing CATASTRONAV_CACHE_DIR that you already have in local machine.
install	if TRUE, will install the key in your local machine for use in future sessions. Defaults to FALSE. If cache_dir is FALSE this parameter is set to FALSE automatically.
verbose	Logical, displays information. Useful for debugging, default is FALSE.
...	Ignored

Value

catrnav_set_cache_dir() is called for its side effects, and returns an (invisible) character with the path to your cache_dir.

catrnav_detect_cache_dir() returns the path to the cache_dir used in this session

About caching

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

See Also

rappdirs::user_config_dir()

Other cache utilities: catrnav_clear_cache()

Examples

```
# Don't run this! It would modify your current state
## Not run:
catrnav_set_cache_dir(verbose = TRUE)

## End(Not run)


catrnav_detect_cache_dir()
```

`catrnav_wfs_get_address_bbox` *Download addresses of Navarre in spatial format*

Description

Get the spatial data of addresses by bounding box.

Usage

```
catrnav_wfs_get_address_bbox(x, srs, verbose = FALSE, count = NULL)
```

Arguments

`x`	See **Details**. It could be: • A numeric vector of length 4 with the coordinates that defines the bounding box: `c(xmin, ymin, xmax, ymax)`. • A `sf/sfc` object, as provided by the **sf** package.
`srs`	SRS/CRS to use on the query. See **Details**.
`verbose`	Logical, displays information. Useful for debugging, default is `FALSE`.
`count`	integer, indicating the maximum number of features to return. The default value `NULL` does not pass this parameter to the query, and the maximum number of features would be determined by the default value of the API service (5,000 in this case).

Details

When `x` is a numeric vector, make sure that the `srs` matches the coordinate values. Additionally, when the `srs` correspond to a geographic reference system (4326, 4258), the function queries the bounding box on EPSG:25830 - ETRS89 / UTM zone 30N. The result is provided always in the SRS provided in `srs`.

When `x` is a `sf` object, the value `srs` is ignored. The query is performed using EPSG:25830 (ETRS89 / UTM zone 30N) and the spatial object is projected back to the SRS of the initial object.

Value

A `sf` object.

Source

SITNA – Catastro de Navarra

See Also

`sf::st_bbox()`

`CatastRo::catr_wfs_get_address_bbox()`

Examples

```
downtown <- c(-1.646812, 42.814528, -1.638036, 42.820320)

ad <- catrnav_wfs_get_address_bbox(downtown, srs = 4326)

library(ggplot2)

ggplot(ad) +
  geom_sf()
```

`catrnav_wfs_get_buildings_bbox`
Download buildings of Navarre in spatial format

Description

Get the spatial data of buildings by bounding box.

Usage

```
catrnav_wfs_get_buildings_bbox(x, srs, verbose = FALSE, count = NULL)
```

Arguments

`x`	See **Details**. It could be: • A numeric vector of length 4 with the coordinates that defines the bounding box: `c(xmin, ymin, xmax, ymax)`. • A `sf/sfc` object, as provided by the **sf** package.
`srs`	SRS/CRS to use on the query. See **Details**.
`verbose`	Logical, displays information. Useful for debugging, default is `FALSE`.
`count`	integer, indicating the maximum number of features to return. The default value `NULL` does not pass this parameter to the query, and the maximum number of features would be determined by the default value of the API service (5,000 in this case).

Details

When `x` is a numeric vector, make sure that the `srs` matches the coordinate values. Additionally, when the `srs` correspond to a geographic reference system (4326, 4258), the function queries the bounding box on EPSG:25830 - ETRS89 / UTM zone 30N. The result is provided always in the SRS provided in `srs`.

When `x` is a `sf` object, the value `srs` is ignored. The query is performed using EPSG:25830 (ETRS89 / UTM zone 30N) and the spatial object is projected back to the SRS of the initial object.

Value

A sf object.

Source

SITNA – Catastro de Navarra

See Also

sf::st_bbox()

CatastRo::catr_wfs_get_buildings_bbox()

Examples

```
downtown <- c(-1.646812, 42.814528, -1.638036, 42.820320)

bu <- catrnav_wfs_get_buildings_bbox(downtown, srs = 4326)

library(ggplot2)

ggplot(bu) +
  geom_sf()
```

---

catrnav_wfs_get_parcels_bbox

Download cadastral parcels of Navarre in spatial format

Description

Get the spatial data of cadastral parcels by bounding box.

Usage

```
catrnav_wfs_get_parcels_bbox(x, srs, verbose = FALSE, count = NULL)
```

Arguments

x	See **Details**. It could be: • A numeric vector of length 4 with the coordinates that defines the bounding box: `c(xmin, ymin, xmax, ymax)`. • A `sf/sfc` object, as provided by the **sf** package.
srs	SRS/CRS to use on the query. See **Details**.
verbose	Logical, displays information. Useful for debugging, default is `FALSE`.
count	integer, indicating the maximum number of features to return. The default value `NULL` does not pass this parameter to the query, and the maximum number of features would be determined by the default value of the API service (5,000 in this case).

Details

When `x` is a numeric vector, make sure that the `srs` matches the coordinate values. Additionally, when the `srs` correspond to a geographic reference system (4326, 4258), the function queries the bounding box on EPSG:25830 - ETRS89 / UTM zone 30N. The result is provided always in the SRS provided in `srs`.

When `x` is a `sf` object, the value `srs` is ignored. The query is performed using EPSG:25830 (ETRS89 / UTM zone 30N) and the spatial object is projected back to the SRS of the initial object.

Value

A `sf` object.

Source

SITNA – Catastro de Navarra

See Also

`sf::st_bbox()`

`CatastRo::catr_wfs_get_parcels_bbox()`

Other parcels: `catrnav_atom_get_parcels()`, `catrnav_atom_get_parcels_db_all()`

Examples

```
downtown <- c(-1.646812, 42.814528, -1.638036, 42.820320)

cp <- catrnav_wfs_get_parcels_bbox(downtown, srs = 4326)

library(ggplot2)

ggplot(cp) +
  geom_sf()
```

Index