

Package: CatastRoNav (via r-universe)

October 25, 2026

Type Package

Title Interface to the 'INSPIRE' Services of the Cadastre of Navarre

Version 1.0.0.9000

Description Provides access to public spatial data from the Cadastre of Navarre through its 'INSPIRE' ATOM feeds, Web Feature Service and Web Map Service endpoints provided by the Government of Navarre through the Sistema de Información Territorial de Navarra ('SITNA'). Supports complete municipal dataset downloads, bounding box feature queries and georeferenced map image downloads for addresses, buildings and cadastral parcels.

License GPL-2

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

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

Depends R (>= 4.1.0)

Imports CatastRo (>= 1.1.0), cli, dplyr, httr2 (>= 1.0.0), lifecycle, mapSpain (>= 1.0.0), rappdirs (>= 0.3.0), sf (>= 1.0.0), terra, tools, utils, xml2

Suggests ggplot2, knitr, quarto, testthat (>= 3.3.0), tibble, tidyterra, withr

VignetteBuilder quarto

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

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

Config/testthat/parallel true

Copyright See file COPYRIGHTS

Encoding UTF-8

X-schema.org-isPartOf <https://ropenspain.es/>

X-schema.org-keywords catastro, maps, navarra, r, r-package, r-stats, ropenspain, spain

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libpng-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

Date/Publication 2026-10-23 07:51:23 UTC

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

RemoteRef HEAD

RemoteSha 6c9652cf3d4614535af9509ecacfe5ac5fadedea

Contents

catnav_atom_get_address	2
catnav_atom_get_address_db_all	4
catnav_atom_get_buildings	5
catnav_atom_get_buildings_db_all	7
catnav_atom_get_parcel s	8
catnav_atom_get_parcel s _db_all	9
catnav_atom_search_munic	11
catnav_clear_cache	12
catnav_set_cache_dir	13
catnav_wfs_get_address_bbox	15
catnav_wfs_get_buildings_bbox	17
catnav_wfs_get_parcel s _bbox	18
catnav_wms_get_layer	20
Index	23

catnav_atom_get_address

ATOM INSPIRE: download all addresses for a municipality

Description

Downloads spatial data for all addresses in a municipality using the ATOM INSPIRE service provided by the Cadastre of Navarre.

Usage

```
catrnav_atom_get_address(
  munic,
  cache = deprecated(),
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

munic	A single municipality name, partial name or cadastral code. Accepts a character string or numeric code. Use catrnav_atom_search_munic() to search for available municipalities.
cache	[Deprecated] This argument is no longer supported because results are always cached.
update_cache	Logical. Whether to refresh the cached file. Defaults to FALSE.
cache_dir	Path to a cache directory. If NULL, uses the configured cache or a directory inside base::tempdir() . See catrnav_set_cache_dir() .
verbose	Logical. Whether to display informational messages.

Details

Empty, missing or non-scalar values of `munic` produce an error before any download is attempted. Returns NULL if no municipality matches `munic`. Municipal ZIP archives are read directly without extracting their contents to disk.

Value

An [sf](#) object. Returns [NULL](#) if the data cannot be retrieved.

Source

SITNA – Catastro de Navarra

See Also

[catrnav_atom_get_address_db_all\(\)](#) lists available municipal downloads. [catrnav_wfs_get_address_bbox\(\)](#) retrieves addresses within a bounding box instead of downloading a complete municipal dataset.

[catrnav_set_cache_dir\(\)](#) configures the download cache and [catrnav_clear_cache\(\)](#) removes cached downloads.

Query ATOM INSPIRE services: [catrnav_atom_get_address_db_all\(\)](#), [catrnav_atom_get_buildings\(\)](#), [catrnav_atom_get_buildings_db_all\(\)](#), [catrnav_atom_get_parcel\(\)](#), [catrnav_atom_get_parcel_db_all\(\)](#), [catrnav_atom_search_munic\(\)](#)

Work with cadastral addresses: [catrnav_atom_get_address_db_all\(\)](#), [catrnav_wfs_get_address_bbox\(\)](#), [catrnav_wms_get_layer\(\)](#)

Examples

```
s <- catnav_atom_get_address("Tudela")

library(ggplot2)

ggplot(s) +
  geom_sf() +
  labs(
    title = "Addresses",
    subtitle = "Tudela"
  )
```

```
catnav_atom_get_address_db_all
```

ATOM INSPIRE: list address download URLs

Description

Creates a [tibble](#) of URLs provided by the Cadastre of Navarre ATOM INSPIRE service for downloading addresses by municipality.

Usage

```
catnav_atom_get_address_db_all(
  cache = deprecated(),
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

cache	[Deprecated] This argument is no longer supported because results are always cached.
update_cache	Logical. Whether to refresh the cached file. Defaults to FALSE.
cache_dir	Path to a cache directory. If NULL, uses the configured cache or a directory inside base::tempdir() . See catnav_set_cache_dir() .
verbose	Logical. Whether to display informational messages.

Value

A [tibble](#) with the following columns. Returns [NULL](#) if the data cannot be retrieved.

- `munic`: Municipality name and cadastral code.
- `url`: ATOM URL for the corresponding municipality.
- `date`: Reference timestamp of the data in UTC.

Source

SITNA – Catastro de Navarra

See Also

`catrnav_atom_get_address()` downloads addresses for a municipality listed in this index.

`catrnav_atom_search_munic()` finds municipality names and cadastral codes.

Query ATOM INSPIRE services: `catrnav_atom_get_address()`, `catrnav_atom_get_buildings()`, `catrnav_atom_get_buildings_db_all()`, `catrnav_atom_get_parcel()`, `catrnav_atom_get_parcel_db_all()`, `catrnav_atom_search_munic()`

Browse ATOM municipal indexes: `catrnav_atom_get_buildings_db_all()`, `catrnav_atom_get_parcel_db_all()`

Work with cadastral addresses: `catrnav_atom_get_address()`, `catrnav_wfs_get_address_bbox()`, `catrnav_wms_get_layer()`

Examples

```
catrnav_atom_get_address_db_all()
```

```
catrnav_atom_get_buildings
```

ATOM INSPIRE: download all buildings for a municipality

Description

Downloads spatial data for all buildings in a municipality using the ATOM INSPIRE service provided by the Cadastre of Navarre.

Usage

```
catrnav_atom_get_buildings(
    munic,
    cache = deprecated(),
    update_cache = FALSE,
    cache_dir = NULL,
    verbose = FALSE
)
```

Arguments

<code>munic</code>	A single municipality name, partial name or cadastral code. Accepts a character string or numeric code. Use <code>catrnav_atom_search_munic()</code> to search for available municipalities.
<code>cache</code>	[Deprecated] This argument is no longer supported because results are always cached.

update_cache	Logical. Whether to refresh the cached file. Defaults to FALSE.
cache_dir	Path to a cache directory. If NULL, uses the configured cache or a directory inside <code>base::tempdir()</code> . See <code>catnav_set_cache_dir()</code> .
verbose	Logical. Whether to display informational messages.

Details

Empty, missing or non-scalar values of `munic` produce an error before any download is attempted. Returns NULL if no municipality matches `munic`. Municipal ZIP archives are read directly without extracting their contents to disk.

Value

An `sf` object. Returns `NULL` if the data cannot be retrieved.

Source

SITNA – Catastro de Navarra

See Also

`catnav_atom_get_buildings_db_all()` lists available municipal downloads. `catnav_wfs_get_buildings_bbox()` retrieves buildings within a bounding box instead of downloading a complete municipal dataset.

`catnav_set_cache_dir()` configures the download cache and `catnav_clear_cache()` removes cached downloads.

Query ATOM INSPIRE services: `catnav_atom_get_address()`, `catnav_atom_get_address_db_all()`, `catnav_atom_get_buildings_db_all()`, `catnav_atom_get_parcel()`, `catnav_atom_get_parcel_db_all()`, `catnav_atom_search_munic()`

Work with cadastral buildings: `catnav_atom_get_buildings_db_all()`, `catnav_wfs_get_buildings_bbox()`, `catnav_wms_get_layer()`

Examples

```
s <- catnav_atom_get_buildings("Iru<U+00F1>a")

library(ggplot2)

ggplot(s) +
  geom_sf() +
  labs(
    title = "Buildings",
    subtitle = "Pamplona / Iru<U+00F1>a"
  )
```

catrnav_atom_get_buildings_db_all

ATOM INSPIRE: list building download URLs

Description

Creates a [tibble](#) of URLs provided by the Cadastre of Navarre ATOM INSPIRE service for downloading buildings by municipality.

Usage

```
catrnav_atom_get_buildings_db_all(
  cache = deprecated(),
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

cache	[Deprecated] This argument is no longer supported because results are always cached.
update_cache	Logical. Whether to refresh the cached file. Defaults to FALSE.
cache_dir	Path to a cache directory. If NULL, uses the configured cache or a directory inside base::tempdir() . See catrnav_set_cache_dir() .
verbose	Logical. Whether to display informational messages.

Value

A [tibble](#) with the following columns. Returns [NULL](#) if the data cannot be retrieved.

- `munic`: Municipality name and cadastral code.
- `url`: ATOM URL for the corresponding municipality.
- `date`: Reference timestamp of the data in UTC.

Source

SITNA – Catastro de Navarra

See Also

[catrnav_atom_get_buildings\(\)](#) downloads buildings for a municipality listed in this index.

[catrnav_atom_search_munic\(\)](#) finds municipality names and cadastral codes.

Query ATOM INSPIRE services: [catrnav_atom_get_address\(\)](#), [catrnav_atom_get_address_db_all\(\)](#), [catrnav_atom_get_buildings\(\)](#), [catrnav_atom_get_parcel\(\)](#), [catrnav_atom_get_parcel_db_all\(\)](#), [catrnav_atom_search_munic\(\)](#)

Browse ATOM municipal indexes: `catrnav_atom_get_address_db_all()`, `catrnav_atom_get_parcel_db_all()`

Work with cadastral buildings: `catrnav_atom_get_buildings()`, `catrnav_wfs_get_buildings_bbox()`, `catrnav_wms_get_layer()`

Examples

```
catrnav_atom_get_buildings_db_all()
```

```
catrnav_atom_get_parcel
```

ATOM INSPIRE: download all cadastral parcels for a municipality

Description

Downloads spatial data for all cadastral parcels in a municipality using the ATOM INSPIRE service provided by the Cadastre of Navarre.

Usage

```
catrnav_atom_get_parcel(
    munic,
    cache = deprecated(),
    update_cache = FALSE,
    cache_dir = NULL,
    verbose = FALSE
)
```

Arguments

munic	A single municipality name, partial name or cadastral code. Accepts a character string or numeric code. Use <code>catrnav_atom_search_munic()</code> to search for available municipalities.
cache	[Deprecated] This argument is no longer supported because results are always cached.
update_cache	Logical. Whether to refresh the cached file. Defaults to FALSE.
cache_dir	Path to a cache directory. If NULL, uses the configured cache or a directory inside <code>base::tempdir()</code> . See <code>catrnav_set_cache_dir()</code> .
verbose	Logical. Whether to display informational messages.

Details

Empty, missing or non-scalar values of `munic` produce an error before any download is attempted. Returns NULL if no municipality matches `munic`. Municipal ZIP archives are read directly without extracting their contents to disk.

Value

An [sf](#) object. Returns [NULL](#) if the data cannot be retrieved.

Source

SITNA – Catastro de Navarra

See Also

[catnav_atom_get_parcel_db_all\(\)](#) lists available municipal downloads. [catnav_wfs_get_parcel_bbox\(\)](#) retrieves cadastral parcels within a bounding box instead of downloading a complete municipal dataset.

[catnav_set_cache_dir\(\)](#) configures the download cache and [catnav_clear_cache\(\)](#) removes cached downloads.

Query ATOM INSPIRE services: [catnav_atom_get_address\(\)](#), [catnav_atom_get_address_db_all\(\)](#), [catnav_atom_get_buildings\(\)](#), [catnav_atom_get_buildings_db_all\(\)](#), [catnav_atom_get_parcel_db_all\(\)](#), [catnav_atom_search_munic\(\)](#)

Work with cadastral parcels: [catnav_atom_get_parcel_db_all\(\)](#), [catnav_wfs_get_parcel_bbox\(\)](#), [catnav_wms_get_layer\(\)](#)

Examples

```
s <- catnav_atom_get_parcel("Iru<U+00F1>a")

library(ggplot2)

ggplot(s) +
  geom_sf() +
  labs(
    title = "Cadastral Zoning",
    subtitle = "Pamplona / Iru<U+00F1>a"
  )
```

`catnav_atom_get_parcel_db_all`*ATOM INSPIRE: list cadastral parcel download URLs*

Description

Creates a [tibble](#) of URLs provided by the Cadastre of Navarra ATOM INSPIRE service for downloading cadastral parcels by municipality.

Usage

```
catrnav_atom_get_parcel_db_all(
  cache = deprecated(),
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

cache	[Deprecated] This argument is no longer supported because results are always cached.
update_cache	Logical. Whether to refresh the cached file. Defaults to FALSE.
cache_dir	Path to a cache directory. If NULL, uses the configured cache or a directory inside <code>base::tempdir()</code> . See <code>catrnav_set_cache_dir()</code> .
verbose	Logical. Whether to display informational messages.

Value

A [tibble](#) with the following columns. Returns [NULL](#) if the data cannot be retrieved.

- `munic`: Municipality name and cadastral code.
- `url`: ATOM URL for the corresponding municipality.
- `date`: Reference timestamp of the data in UTC.

Source

[SITNA – Catastro de Navarra](#)

See Also

`catrnav_atom_get_parcel_db_all()` downloads cadastral parcels for a municipality listed in this index.

`catrnav_atom_search_munic()` finds municipality names and cadastral codes.

Query ATOM INSPIRE services: `catrnav_atom_get_address()`, `catrnav_atom_get_address_db_all()`, `catrnav_atom_get_buildings()`, `catrnav_atom_get_buildings_db_all()`, `catrnav_atom_get_parcel_db_all()`, `catrnav_atom_search_munic()`

Browse ATOM municipal indexes: `catrnav_atom_get_address_db_all()`, `catrnav_atom_get_buildings_db_all()`

Work with cadastral parcels: `catrnav_atom_get_parcel_db_all()`, `catrnav_wfs_get_parcel_bbox()`, `catrnav_wms_get_layer()`

Examples

```
catrnav_atom_get_parcel_db_all()
```

catrnav_atom_search_munic

ATOM INSPIRE: search for municipalities

Description

Searches for a municipality by name or cadastral code in the Cadastre of Navarre ATOM index.

Usage

```
catrnav_atom_search_munic(
  munic,
  cache = deprecated(),
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

munic	A single municipality name, partial name or cadastral code. Accepts a character string or numeric code. Use catrnav_atom_search_munic() to search for available municipalities.
cache	[Deprecated] This argument is no longer supported because results are always cached.
update_cache	Logical. Whether to refresh the cached file. Defaults to FALSE.
cache_dir	Path to a cache directory. If NULL, uses the configured cache or a directory inside base::tempdir() . See catrnav_set_cache_dir() .
verbose	Logical. Whether to display informational messages.

Value

A [tibble](#) with the municipality name and cadastral code. Returns [NULL](#) if no match is found or the data cannot be retrieved.

See Also

[catrnav_atom_get_address_db_all\(\)](#) provides the index used by this search. [catrnav_atom_get_buildings_db_all\(\)](#) and [catrnav_atom_get_parcel_db_all\(\)](#) list downloads for the other data types. Use the selected name or cadastral code with [catrnav_atom_get_address\(\)](#), [catrnav_atom_get_buildings\(\)](#) or [catrnav_atom_get_parcel_db_all\(\)](#) to download a complete municipal dataset.

Query ATOM INSPIRE services: [catrnav_atom_get_address\(\)](#), [catrnav_atom_get_address_db_all\(\)](#), [catrnav_atom_get_buildings\(\)](#), [catrnav_atom_get_buildings_db_all\(\)](#), [catrnav_atom_get_parcel_db_all\(\)](#), [catrnav_atom_get_parcel_db_all\(\)](#)

Examples

```
catrnav_atom_search_munic("Pamplona")

# Search using a numeric cadastral code.
catrnav_atom_search_munic(201)
```

catrnav_clear_cache	<i>Clear your CatastRoNav cache directory</i>
---------------------	--

Description

Use this function with caution. It clears cached data and configuration, specifically:

- Deletes the **CatastRoNav** configuration directory when `config = TRUE` (`tools::R_user_dir("CatastRoNav", "config")`).
- Deletes the `cache_dir` directory and its contents when `cached_data = TRUE`.
- Clears the `CATASTRONAV_CACHE_DIR` environment variable.

Usage

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

Arguments

<code>config</code>	A logical value indicating whether to delete the CatastRoNav configuration directory.
<code>cached_data</code>	If <code>TRUE</code> , deletes your <code>cache_dir</code> and all its contents.
<code>verbose</code>	Logical. Whether to display informational messages.

Details

With `config = TRUE` and `cached_data = TRUE`, this function resets the cache state as if you had never used **CatastRoNav**.

Value

`NULL`, invisibly. This function is called for its side effects.

See Also

[catrnav_detect_cache_dir\(\)](#) identifies the active cache path before deletion. [catrnav_set_cache_dir\(\)](#) configures a new cache afterward.

Manage the local cache: [catrnav_set_cache_dir\(\)](#)

Examples

```
# Caution! This modifies your current state.
## Not run:
my_cache <- catrnav_detect_cache_dir()

example_cache <- file.path(tempdir(), "example", "cache")
catrnav_set_cache_dir(example_cache, verbose = FALSE)

catrnav_clear_cache(verbose = TRUE)

# Restore the initial cache.
catrnav_set_cache_dir(my_cache)
identical(my_cache, catrnav_detect_cache_dir())

## End(Not run)
```

catrnav_set_cache_dir *Set your **CatastRoNav** cache directory*

Description

Configures the cache directory used by **CatastRoNav**. Use [Sys.getenv\(\)](#) with "CATASTRONAV_CACHE_DIR" or [catrnav_detect_cache_dir\(\)](#) to inspect the current path.

Usage

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

catrnav_detect_cache_dir()
```

Arguments

cache_dir	Path to a cache directory. If NULL or FALSE, the function stores cached files in a temporary directory. See base::tempdir() .
overwrite	A logical value indicating whether to overwrite an existing CATASTRONAV_CACHE_DIR value.
install	Logical. Whether to store the path locally for use in future sessions. Defaults to FALSE.
verbose	Logical. Whether to display informational messages.

Details

By default, when no `cache_dir` is set, **CatastRoNav** uses a directory inside `base::tempdir()`. Files in this directory are temporary and are removed when the R session ends. To persist a cache across R sessions, use `catrnav_set_cache_dir(cache_dir, install = TRUE)`. This writes the chosen path to a configuration file under `tools::R_user_dir()` with "CatastRoNav" and "config".

Value

`catrnav_set_cache_dir()` returns a [character](#) string containing the cache path, invisibly. This function is primarily called for its side effects.

`catrnav_detect_cache_dir()` returns a [character](#) string containing the cache path used in the current session.

Caching strategies

Source files are always cached after download. **CatastRoNav** implements the following caching options:

- For occasional use, rely on the default `tempdir()`-based cache without installing a persistent path.
- Modify the cache for a single session by setting `catrnav_set_cache_dir(cache_dir = "a/path/here")`.
- For reproducible workflows, install a persistent cache with `catrnav_set_cache_dir(cache_dir = "a/path/here", install = TRUE)`. This cache is kept across R sessions.
- To cache specific files elsewhere, use the `cache_dir` argument in the corresponding function.

Cached files can occasionally become corrupt. In that case, download the data again by setting `update_cache = TRUE` in an ATOM or WMS function. ATOM downloads check the file size with a HEAD request and report downloads larger than 20 MB before fetching the body. Failed ATOM updates preserve the previous cached file. WFS queries reuse their cached response until the cache is cleared.

ATOM indexes are stored in databases, municipal downloads in `atom_ad`, `atom_bu` or `atom_cp` and WFS responses in `wfs_inspire_cache`.

The ATOM cache argument is deprecated and no longer changes caching. Use a temporary `cache_dir` when downloads should last only for a session.

HTTP settings

ATOM downloads and WFS queries use the `catastronav_timeout` and `catastronav_ssl_verify` options. If unset, the `CATASTRONAV_TIMEOUT` and `CATASTRONAV_SSL_VERIFY` environment variables are used, followed by the `catastro_timeout` and `catastro_ssl_verify` options. The defaults are 300 seconds and enabled SSL verification. WFS queries apply these settings only for the request and restore the previous **CatastRo** options. WMS request settings are passed to `mapSpain::esp_get_tiles()` through the options argument of `catrnav_wms_get_layer()`.

If a download fails, use `verbose = TRUE` to inspect the request and `catrnav_detect_cache_dir()` to identify the active cache path.

Note

The configuration location has moved from `rappdirs::user_config_dir("CatastRoNav", "R")` to `tools::R_user_dir()` with "CatastRoNav" and "config". Existing configuration files are migrated automatically. A migration message is shown only once.

See Also

`tools::R_user_dir()` determines the persistent configuration directory. `base::tempdir()` provides the default temporary cache directory. `catnav_atom_get_address()`, `catnav_atom_get_buildings()` and `catnav_atom_get_parcelles()` cache municipal downloads. `catnav_wfs_get_address_bbox()`, `catnav_wfs_get_buildings_bbox()` and `catnav_wfs_get_parcelles_bbox()` cache spatial queries. `catnav_wms_get_layer()` caches map images.

Manage the local cache: `catnav_clear_cache()`

Examples

```
# Caution! This modifies your current state.
## Not run:
my_cache <- catnav_detect_cache_dir()

example_cache <- file.path(tempdir(), "example", "cache")
catnav_set_cache_dir(example_cache)

catnav_detect_cache_dir()

# Restore the initial cache.
catnav_set_cache_dir(my_cache)
identical(my_cache, catnav_detect_cache_dir())

## End(Not run)

catnav_detect_cache_dir()
```

```
catnav_wfs_get_address_bbox
```

WFS INSPIRE: retrieve addresses

Description

Retrieves spatial address data from the Cadastre of Navarre WFS INSPIRE service. `catnav_wfs_get_address_bbox()` retrieves features within the supplied bounding box. See **Bounding box**.

Usage

```
catnav_wfs_get_address_bbox(x, srs = 4326, verbose = FALSE, count = NULL)
```

Arguments

x	Input defining the query area. See Bounding box . It can be: - A numeric vector of length 4 with the coordinates that define the bounding box: c(xmin, ymin, xmax, ymax). - An sf or sfc object from sf.
srs	The CRS to use for the query. Defaults to 4326. See Bounding box .
verbose	Logical. Whether to display informational messages.
count	A positive whole number specifying the maximum number of features to return. If NULL, the service default applies.

Details

Responses are cached in the `wfs_inspire_cache` subdirectory of the `catnav_set_cache_dir()` cache. Repeated queries reuse the cached file. Clear the cache with `catnav_clear_cache()` to download fresh results.

Value

An [sf](#) object. Returns `NULL` if the data cannot be retrieved.

API limits

The service returns a maximum of 5,000 features by default. Use `count` to request a smaller result.

Bounding box

When `x` is a numeric vector, make sure that `srs` matches the coordinate values. The function queries the bounding box in [EPSG:25830](#), ETRS89 / UTM zone 30N, then transforms the result back to `srs`.

When `x` is an [sf](#) or [sfc](#) object, `srs` is ignored. The object's bounding box is used for the query and the result is transformed back to the input CRS. See [sf::st_bbox\(\)](#).

Source

[SITNA – Catastro de Navarra](#)

See Also

[catnav_atom_get_address\(\)](#) downloads all addresses for a municipality.

[catnav_wms_get_layer\(\)](#) retrieves a georeferenced map image rather than individual spatial features.

Query WFS INSPIRE services: [catnav_wfs_get_buildings_bbox\(\)](#), [catnav_wfs_get_parcel_bbox\(\)](#)

Work with cadastral addresses: [catnav_atom_get_address\(\)](#), [catnav_atom_get_address_db_all\(\)](#), [catnav_wms_get_layer\(\)](#)

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

WFS INSPIRE: retrieve buildings

Description

Retrieves spatial building data from the Cadastre of Navarre WFS INSPIRE service. [catrnav_wfs_get_buildings_bbox\(\)](#) retrieves features within the supplied bounding box. See **Bounding box**.

Usage

```
catrnav_wfs_get_buildings_bbox(x, srs = 4326, verbose = FALSE, count = NULL)
```

Arguments

x	Input defining the query area. See Bounding box . It can be: • A numeric vector of length 4 with the coordinates that define the bounding box: <code>c(xmin, ymin, xmax, ymax)</code>. • An sf or sfc object from sf.
srs	The CRS to use for the query. Defaults to 4326. See Bounding box .
verbose	Logical. Whether to display informational messages.
count	A positive whole number specifying the maximum number of features to return. If NULL, the service default applies.

Details

Responses are cached in the `wfs_inspire_cache` subdirectory of the [catrnav_set_cache_dir\(\)](#) cache. Repeated queries reuse the cached file. Clear the cache with [catrnav_clear_cache\(\)](#) to download fresh results.

Value

An [sf](#) object. Returns [NULL](#) if the data cannot be retrieved.

API limits

The service returns a maximum of 5,000 features by default. Use count to request a smaller result.

Bounding box

When `x` is a numeric vector, make sure that `srs` matches the coordinate values. The function queries the bounding box in **EPSG:25830**, ETRS89 / UTM zone 30N, then transforms the result back to `srs`.

When `x` is an `sf` or `sfc` object, `srs` is ignored. The object's bounding box is used for the query and the result is transformed back to the input CRS. See `sf::st_bbox()`.

Source

SITNA – Catastro de Navarra

See Also

`catrnav_atom_get_buildings()` downloads all buildings for a municipality.

`catrnav_wms_get_layer()` retrieves a georeferenced map image rather than individual spatial features.

Query WFS INSPIRE services: `catrnav_wfs_get_address_bbox()`, `catrnav_wfs_get_parcel_bbox()`

Work with cadastral buildings: `catrnav_atom_get_buildings()`, `catrnav_atom_get_buildings_db_all()`, `catrnav_wms_get_layer()`

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_parcel_bbox
```

WFS INSPIRE: retrieve cadastral parcels

Description

Retrieves spatial cadastral parcel data from the Cadastre of Navarra WFS INSPIRE service. `catrnav_wfs_get_parcel_bbox` retrieves features within the supplied bounding box. See **Bounding box**.

Usage

```
catnav_wfs_get_parcel_bbox(x, srs = 4326, verbose = FALSE, count = NULL)
```

Arguments

x	Input defining the query area. See Bounding box . It can be: - A numeric vector of length 4 with the coordinates that define the bounding box: <code>c(xmin, ymin, xmax, ymax)</code>. - An <code>sf</code> or <code>sfc</code> object from <code>sf</code>.
srs	The CRS to use for the query. Defaults to 4326. See Bounding box .
verbose	Logical. Whether to display informational messages.
count	A positive whole number specifying the maximum number of features to return. If NULL, the service default applies.

Details

Responses are cached in the `wfs_inspire_cache` subdirectory of the `catnav_set_cache_dir()` cache. Repeated queries reuse the cached file. Clear the cache with `catnav_clear_cache()` to download fresh results.

Value

An `sf` object. Returns `NULL` if the data cannot be retrieved.

API limits

The service returns a maximum of 5,000 features by default. Use `count` to request a smaller result.

Bounding box

When `x` is a numeric vector, make sure that `srs` matches the coordinate values. The function queries the bounding box in [EPSG:25830](#), ETRS89 / UTM zone 30N, then transforms the result back to `srs`.

When `x` is an `sf` or `sfc` object, `srs` is ignored. The object's bounding box is used for the query and the result is transformed back to the input CRS. See `sf::st_bbox()`.

Source

[SITNA – Catastro de Navarra](#)

See Also

`catnav_atom_get_parcel_bbox()` downloads all cadastral parcels for a municipality.

`catnav_wms_get_layer()` retrieves a georeferenced map image rather than individual spatial features.

Query WFS INSPIRE services: `catnav_wfs_get_address_bbox()`, `catnav_wfs_get_buildings_bbox()`

Work with cadastral parcels: `catnav_atom_get_parcel_bbox()`, `catnav_atom_get_parcel_db_all()`, `catnav_wms_get_layer()`

Examples

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

cp <- catrnav_wfs_get_parcel_bbox(downtown, srs = 4326)

library(ggplot2)

ggplot(cp) +
  geom_sf()
```

catrnav_wms_get_layer *WMS INSPIRE: download georeferenced map images*

Description

Downloads georeferenced map images from the Cadastre of Navarre WMS service. This function wraps [mapSpain::esp_get_tiles\(\)](#).

Usage

```
catrnav_wms_get_layer(
  x,
  srs = 4326,
  what = c("building", "parcel", "address"),
  styles = c("default", "ELFCadastre"),
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  crop = FALSE,
  options = NULL,
  ...
)
```

Arguments

x	Input defining the query area. See Bounding box . It can be: - A numeric vector of length 4 with the coordinates that define the bounding box: <code>c(xmin, ymin, xmax, ymax)</code>. - An sf or sfc object from sf.
srs	The CRS to use for the query. Defaults to 4326. See Bounding box .
what	WMS layer to download. See Layers and styles .
styles	Style to apply to the selected WMS layer. See Layers and styles .
update_cache	Logical. Whether to refresh the cached file. Defaults to FALSE.
cache_dir	Path to a cache directory. If NULL, uses the configured cache or a directory inside base::tempdir() . See catrnav_set_cache_dir() .

verbose	Logical. Whether to display informational messages.
crop	Logical. If TRUE, crop results to the specified x extent. If x is an sf object with one POINT, crop is set to FALSE. See terra::crop() .
options	A named list containing additional options to pass to the query.
...	Arguments passed on to mapSpain::esp_get_tiles
res	Character string or number. Only valid for WMS providers. Resolution (in pixels) of the final tile.
bbox_expand	Number. Expansion percentage of the bounding box of x.
transparent	Logical. Whether to use a transparent background, if supported.
mask	Logical. TRUE to mask the result to x. See terra::mask() .

Details

Returns NULL if the request cannot run or no image can be retrieved.

Value

A [SpatRaster](#) with three RGB or four RGBA layers. Returns [NULL](#) if the request cannot run or no image can be retrieved. See [terra::RGB\(\)](#).

Layers and styles

Layers:

The what argument selects one of the following API layers:

- "parcel": CP.CadastralParcel.
- "building": BU.Building.
- "address": AD.Address.

Styles:

The WMS service provides different styles for each layer (what argument). Available styles include:

- "parcel": "default" and "ELFCadastre".
- "building": "default".
- "address": "default".

Bounding box

When x is a numeric vector, make sure that srs matches the coordinate values. When x is an [sf](#) object, the srs value is ignored.

The query uses [EPSG:3857](#) (Web Mercator), then transforms the tile back to the SRS of x. If the tile appears distorted, provide a spatial object as x or set srs to the SRS of the requested tile. See [Examples](#).

Source

[SITNA – Catastro de Navarra](#)

See Also

- `catrnav_wfs_get_address_bbox()`, `catrnav_wfs_get_buildings_bbox()` and `catrnav_wfs_get_parcel_bboxes_bbox()` retrieve individual spatial features within a bounding box.
- `catrnav_atom_get_address()`, `catrnav_atom_get_buildings()` and `catrnav_atom_get_parcel_bboxes()` download complete municipal vector datasets.
- `catrnav_set_cache_dir()` configures where downloaded images are cached. `catrnav_clear_cache()` removes cached images.
- `mapSpain::esp_get_tiles()` downloads map tiles.
- `terra::RGB()` identifies RGB channels.
- `terra::plotRGB()` and `tidyterra::geom_spatraster_rgb()` plot RGB rasters.

Work with cadastral addresses: `catrnav_atom_get_address()`, `catrnav_atom_get_address_db_all()`, `catrnav_wfs_get_address_bbox()`

Work with cadastral buildings: `catrnav_atom_get_buildings()`, `catrnav_atom_get_buildings_db_all()`, `catrnav_wfs_get_buildings_bbox()`

Work with cadastral parcels: `catrnav_atom_get_parcel_bboxes()`, `catrnav_atom_get_parcel_bboxes_db_all()`, `catrnav_wfs_get_parcel_bboxes_bbox()`

Examples

```
bu <- catrnav_wms_get_layer(
  c(-1.646812, 42.814528, -1.638036, 42.820320),
  srs = 4326,
  what = "building"
)

library(mapSpain)
library(ggplot2)
library(tidyterra)

ggplot() +
  geom_spatraster_rgb(data = bu)

# Download cadastral parcels.
parc <- catrnav_wms_get_layer(
  c(-1.646812, 42.814528, -1.638036, 42.820320),
  srs = 4326,
  what = "parcel"
)

ggplot() +
  geom_spatraster_rgb(data = parc)
```

Index

* addresses

- catrnav_atom_get_address, 2
- catrnav_atom_get_address_db_all, 4
- catrnav_wfs_get_address_bbox, 15
- catrnav_wms_get_layer, 20

* atom_indexes

- catrnav_atom_get_address_db_all, 4
- catrnav_atom_get_buildings_db_all, 7
- catrnav_atom_get_parcel_db_all, 9

* atom_services

- catrnav_atom_get_address, 2
- catrnav_atom_get_address_db_all, 4
- catrnav_atom_get_buildings, 5
- catrnav_atom_get_buildings_db_all, 7
- catrnav_atom_get_parcel, 8
- catrnav_atom_get_parcel_db_all, 9
- catrnav_atom_search_munic, 11

* buildings

- catrnav_atom_get_buildings, 5
- catrnav_atom_get_buildings_db_all, 7
- catrnav_wfs_get_buildings_bbox, 17
- catrnav_wms_get_layer, 20

* cache_utilities

- catrnav_clear_cache, 12
- catrnav_set_cache_dir, 13

* parcels

- catrnav_atom_get_parcel, 8
- catrnav_atom_get_parcel_db_all, 9
- catrnav_wfs_get_parcel_bbox, 18
- catrnav_wms_get_layer, 20

* search_tools

- catrnav_atom_search_munic, 11

* wfs_services

- catrnav_wfs_get_address_bbox, 15
- catrnav_wfs_get_buildings_bbox, 17
- catrnav_wfs_get_parcel_bbox, 18

* wms_services

- catrnav_wms_get_layer, 20

base::tempdir(), 3, 4, 6–8, 10, 11, 13–15, 20

- catrnav_atom_get_address, 2
- catrnav_atom_get_address(), 5–7, 9–11, 15, 16, 22
- catrnav_atom_get_address_db_all, 4
- catrnav_atom_get_address_db_all(), 3, 6–11, 16, 22
- catrnav_atom_get_buildings, 5
- catrnav_atom_get_buildings(), 3, 5, 7–11, 15, 18, 22
- catrnav_atom_get_buildings_db_all, 7
- catrnav_atom_get_buildings_db_all(), 3, 5, 6, 9–11, 18, 22
- catrnav_atom_get_parcel, 8
- catrnav_atom_get_parcel(), 3, 5–7, 10, 11, 15, 19, 22
- catrnav_atom_get_parcel_db_all, 9
- catrnav_atom_get_parcel_db_all(), 3, 5–9, 11, 19, 22
- catrnav_atom_search_munic, 11
- catrnav_atom_search_munic(), 3, 5–11
- catrnav_clear_cache, 12
- catrnav_clear_cache(), 3, 6, 9, 15–17, 19, 22
- catrnav_detect_cache_dir
(catrnav_set_cache_dir), 13
- catrnav_detect_cache_dir(), 12–14
- catrnav_set_cache_dir, 13
- catrnav_set_cache_dir(), 3, 4, 6–12, 16, 17, 19, 20, 22
- catrnav_wfs_get_address_bbox, 15
- catrnav_wfs_get_address_bbox(), 3, 5, 15, 18, 19, 22
- catrnav_wfs_get_buildings_bbox, 17
- catrnav_wfs_get_buildings_bbox(), 6, 8, 15–17, 19, 22

catrnav_wfs_get_parcel_bboxes, 18
catrnav_wfs_get_parcel_bboxes(), 9, 10,
15, 16, 18, 22
catrnav_wms_get_layer, 20
catrnav_wms_get_layer(), 3, 5, 6, 8–10,
14–16, 18, 19
character, 14

mapSpain::esp_get_tiles, 21
mapSpain::esp_get_tiles(), 14, 20, 22

NULL, 3, 4, 6, 7, 9–12, 16, 17, 19, 21

sf, 3, 6, 9, 16–21
sf::st_bbox(), 16, 18, 19
sfc, 16–20
SpatRaster, 21
Sys.getenv(), 13

tempdir(), 14
terra::crop(), 21
terra::mask(), 21
terra::plotRGB(), 22
terra::RGB(), 21, 22
tibble, 4, 7, 9–11
tidyterra::geom_spatraster_rgb(), 22
tools::R_user_dir(), 14, 15