Package 'gchartsmap'

June 4, 2025

Type Package

Title Access 'Google Charts' Map Data

Version 0.1.2

Description Connects to the 'Google Charts' geographic data resources hosted at <https://www.gstatic.com/charts/geochart/10/mapfiles/US-500_METROS.js>, allowing the user to download contents to use as a reference for related services like 'Google Trends'.

License GPL-3

URL https://github.com/odeleongt/gchartsmap

Imports httr, jsonlite, sf, tigris

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

BugReports https://github.com/odeleongt/gchartsmap/issues

NeedsCompilation no

Author Oscar de Leon [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-1344-4412>)

Maintainer Oscar de Leon <odeleon@emory.edu>

Repository CRAN

Date/Publication 2025-06-04 12:00:08 UTC

Contents

`gchart_available_areas` *Get available areas*

Description

Get a list of areas that have been downloaded

Usage

```
gchart_available_areas(cache = gchart_get_cache_path())
```

Arguments

`cache`	Path where the downloaded data is stored

`gchart_cache_dir` *Verify cache directory*

Description

Ensure that the cache directory exists

Usage

```
gchart_cache_dir(
  path = tools::R_user_dir(package = "gchartsmap", which = "cache")
)
```

Arguments

`path`	Path to verify

gchart_generate_us_areas *Generate Google Charts spatial data for US areas*

Description

This function queries 'Google Charts' resources to identify the US geographic areas used in services like Google Trends, and uses geographic data from the US Census Bureau to provide those areas with subdivisions at the county level.

Usage

```
gchart_generate_us_areas(areas = 1:1000L, limit = 1000)
```

Arguments

areas	Area codes to get. Should be integers.
limit	Maximum number of areas to look for.

Value

Returns a simple features 'data.frame' with class 'sf', representing the spatial data for all areas with a valid id between 1 and a 1000 from the Google Charts servers, using the WGS84 (epsg = 4326) coordinate reference system. You need to first run 'gchart_set_cache()' so the package knows where to store the downloaded data.

Examples

```
library(package = "gchartsmap")

# set the cache path to your system's cache path
gchartsmap::gchart_set_cache()

# GET and process area 500
gchartsmap::gchart_generate_us_areas(500L)
```

gchart_get_cache_path *Get the cache path*

Description

Get the saved cache path

Usage

```
gchart_get_cache_path(path = NULL)
```

Arguments

path	Path to use as cache

Value

Returns the path to the local cache as set in the 'R_GOOGLE_CHART_CACHE' environment variable. If that is not set, gets the system's default cache path for the package as provided by 'tools::R_user_dir()'.

Examples

```
library(package = "gchartsmap")

# set the cache path to your system's cache path
gchartsmap::gchart_set_cache()

# check the set cache
gchartsmap::gchart_get_cache_path()
```

---

gchart_get_us_areas *Get Google Charts data for US areas*

Description

Access the Google Charts geochart data for US areas

Usage

```
gchart_get_us_areas(
  areas,
  server = "https://www.gstatic.com/charts/geochart/10/mapfiles/",
  cache = gchart_get_cache_path(),
  limit = 1000
)
```

Arguments

areas	Area codes to get. Should be integers.
server	Google geochart server to access.
cache	Path to store downloaded data.
limit	Maximum number of areas to look for

Details

The function invisibly returns the file path for successful requests or the response status code for failed requests, in a character vector with the area name for each element.

gchart_process_us_areas *Process Google Charts data for US areas*

Description

Process the downloaded Google Charts geochart data for US areas

Usage

```
gchart_process_us_areas(areas, cache = gchart_get_cache_path())
```

Arguments

areas	Area codes to get. Should be integers. If not provided, all available areas are processed.
cache	Path where the downloaded data is stored.

Details

Google Charts data is served as JavaScript code that defines objects with the desired data. This function processes the locally-available Google Charts js files to generate spatial objects.

gchart_set_cache *Set up a cache directory*

Description

Ensures that the directory exists and sets the environment variable for access.

Usage

```
gchart_set_cache(
  path = tools::R_user_dir(package = "gchartsmap", which = "cache"),
  install = FALSE,
  overwrite = FALSE,
  home = "HOME"
)
```

Arguments

path	Path to use for the package cache.
install	if TRUE, will install the cache path in your .Renviron file for use in future sessions. Defaults to FALSE.
overwrite	If this is set to TRUE, it will overwrite an existing cache path that you already have in your .Renviron file.
home	Path for the .Renviron file. Defaults to "HOME".

Value

Sets and returns the path to the cache where downloaded data will be stored. Is used for the side effect of setting the ‘R_GOOGLE_CHART_CACHE‘ environment variable, and can store the path in ‘.Renviron‘ for use in future R sessions if requested.

Examples

```
library(package = "gchartsmap")

# set the cache path to your system's cache path
gchartsmap::gchart_set_cache()


# save the cache path in your home .Renviron file
gchartsmap::gchart_set_cache(install = TRUE)
```

Index