

Package ‘basemaps’

March 4, 2024

Type Package

Title Accessing Spatial Basemaps in R

Version 0.0.6

Depends R (>= 3.5.0)

Date 2024-03-03

Description A lightweight package to access spatial basemaps from open sources such as 'Open-StreetMap', 'Carto', 'Mapbox' and others in R.

License GPL-3

Encoding UTF-8

Imports sf, slippymath, httr, curl, terra, stars, pbapply, magick, utils, grDevices, methods

Suggests raster, ggplot2, png, mapview, mapedit, testthat, covr

BugReports <https://github.com/16eagle/basemaps/issues>

RoxygenNote 7.2.3

NeedsCompilation no

Author Jakob Schwalb-Willmann [aut, cre]
(<https://orcid.org/0000-0003-2665-1509>)

Maintainer Jakob Schwalb-Willmann <dev@schwalb-willmann.de>

Repository CRAN

Date/Publication 2024-03-04 00:10:03 UTC

R topics documented:

basemap	2
cache	6
data	7
defaults	8
draw_ext	9
get_maptypes	10
plot	11

Index	13
--------------	-----------

`basemap`*Get a spatial basemap*

Description

These functions (down)load and cache a basemap of a defined extent `ext`, `map_service` and `map_type` and return it as an object of the defined class. Alternatively to defining the following arguments, [set_defaults](#) can be used to define basemap preferences once for the running session.

Usage

```
basemap(  
    ext = NULL,  
    map_service = NULL,  
    map_type = NULL,  
    map_res = NULL,  
    map_token = NULL,  
    map_dir = NULL,  
    class = "plot",  
    force = FALSE,  
    ...,  
    verbose = TRUE  
)
```

```
basemap_plot(  
    ext = NULL,  
    map_service = NULL,  
    map_type = NULL,  
    map_res = NULL,  
    map_token = NULL,  
    map_dir = NULL,  
    force = NULL,  
    ...,  
    verbose = TRUE  
)
```

```
basemap_magick(  
    ext = NULL,  
    map_service = NULL,  
    map_type = NULL,  
    map_res = NULL,  
    map_token = NULL,  
    map_dir = NULL,  
    force = NULL,  
    ...,  
    verbose = TRUE  
)
```

```
basemap_png(  
    ext = NULL,  
    map_service = NULL,  
    map_type = NULL,  
    map_res = NULL,  
    map_token = NULL,  
    map_dir = NULL,  
    force = NULL,  
    ...,  
    verbose = TRUE  
)
```

```
basemap_geotif(  
    ext = NULL,  
    map_service = NULL,  
    map_type = NULL,  
    map_res = NULL,  
    map_token = NULL,  
    map_dir = NULL,  
    force = NULL,  
    ...,  
    verbose = TRUE  
)
```

```
basemap_terra(  
    ext = NULL,  
    map_service = NULL,  
    map_type = NULL,  
    map_res = NULL,  
    map_token = NULL,  
    map_dir = NULL,  
    force = NULL,  
    ...,  
    verbose = TRUE  
)
```

```
basemap_raster(  
    ext = NULL,  
    map_service = NULL,  
    map_type = NULL,  
    map_res = NULL,  
    map_token = NULL,  
    map_dir = NULL,  
    force = NULL,  
    ...,  
    verbose = TRUE  
)
```

```
basemap_stars(  
  ext = NULL,  
  map_service = NULL,  
  map_type = NULL,  
  map_res = NULL,  
  map_token = NULL,  
  map_dir = NULL,  
  force = NULL,  
  ...,  
  verbose = TRUE  
)
```

```
basemap_ggplot(  
  ext = NULL,  
  map_service = NULL,  
  map_type = NULL,  
  map_res = NULL,  
  map_token = NULL,  
  map_dir = NULL,  
  force = NULL,  
  ...,  
  verbose = TRUE  
)
```

```
basemap_gglayer(  
  ext = NULL,  
  map_service = NULL,  
  map_type = NULL,  
  map_res = NULL,  
  map_token = NULL,  
  map_dir = NULL,  
  force = NULL,  
  ...,  
  verbose = TRUE  
)
```

```
basemap_mapview(  
  ext = NULL,  
  map_service = NULL,  
  map_type = NULL,  
  map_res = NULL,  
  map_token = NULL,  
  map_dir = NULL,  
  force = NULL,  
  ...,  
  verbose = TRUE  
)
```

Arguments

<code>ext</code>	extent to be covered by the basemap as any spatial class supported by <code>st_bbox</code> .
<code>map_service</code>	character, a map service, either "osm", "carto" or "mapbox". Default is "osm".
<code>map_type</code>	character, a map type, e.g. "streets". For a full list of available map types, see get_maptypes .
<code>map_res</code>	numeric, resolution of base map in range from 0 to 1.
<code>map_token</code>	character, authentication token for services that require registration, which are "osm_stamen", "osm_stadia", "osm_thunderforest" and "mapbox". Register at https://stadiamaps.com/ (for stamen and stadia), https://www.thunderforest.com/ and/or https://www.mapbox.com/ to get tokens. Ignored for all other map services.
<code>map_dir</code>	character, cache directory where downloaded basemap tiles will be stored. By default, a temporary directory is used, which is destroyed when the session is terminated.
<code>class</code>	character, output class, either either plot (default), magick, png, geotif or if suggested packages are installed, terra, raster, stars, ggplot, gglayer or mapview.
<code>force</code>	logical, whether to force download over cached files or not. Default is FALSE.
<code>...</code>	additional arguments, including <ul style="list-style-type: none"> <code>browse</code>, logical, for <code>class = "png"</code> and interactive sessions only. Whether to open the png file in the system's default PNG viewer or not. Default is TRUE. <code>col</code>, character vector of colours for transforming single-layer basemaps into RGB, if <code>class = "png"</code> or <code>class = "magick"</code>. Default is <code>topo.colors(25)</code>. <code>dpi</code>, numeric vector of length 1 or 2 specifying the resolution of the image in DPI (dots per inch) for x and y (in that order) - it is recycled to length 2.
<code>verbose</code>	logical, if TRUE, messages and progress information are displayed on the console (default).

Value

A basemap of the defined class in Web/Pseudo Mercator Projection (EPSG: 3857)

Note

See [get_maptypes](#) for available map services and their sources.

The use of the map services "osm_thunderforest" and "mapbox" require registration to obtain an API token/key which can be supplied to `map_token`. Register at <https://www.thunderforest.com/> and/or <https://www.mapbox.com/> to get a token.

Examples

```
library(basemaps)

# example extent
```

```

data(ext)

# view all available maps
get_maotypes()

# set defaults for the basemap
set_defaults(map_service = "osm", map_type = "terrain_bg")
# for "osm_stamen", "osm_stadia", osm "thunderforest" and "mapbox" maps, you need a API token.
# Register for free at stadiamaps.com, thunderforest.com and mapbox.com to get tokens.

## Not run:
# load and return basemap map as raster (default)
map <- basemap(ext)

# or explicitly as different classes such as:
basemap_magick(ext)
basemap_raster()
basemap_stars()

# or as files:
basemap_geotif()
basemap_png()

# or as plots:
basemap_plot(ext)
basemap_mapview()

# including ggplot2:
basemap_ggplot(ext)

# or as ggplot2 layer:
library(ggplot2)
ggplot() +
  basemap_gglayer(ext) +
  scale_fill_identity() +
  coord_sf()

# or, when combined with an sf vector object,
# make sure to use Web/Pseudo Mercator (EPSG 3857), as this is
# the CRS in which all basemaps are returned (see "Value"):
library(sf)
ext <- st_transform(ext, crs = st_crs(3857))
ggplot() +
  basemap_gglayer(ext) +
  geom_sf(data = ext, color = "red", fill = "transparent") +
  coord_sf() +
  scale_fill_identity()

## End(Not run)

```

Description

This function flushes the basemaps cache and thereby removes all previously queried and/or composited products from the map directories (temporary or user-defined using the argument `map_dir`) used during the current session.

Usage

```
flush_cache()
```

Value

None.

Examples

```
library(basemaps)
flush_cache()
```

data	<i>Example extent</i>
------	-----------------------

Description

The example datasets contain the sf objects `ext` and `ext_eur` that can be used to call [basemap](#) and the associated functions.

Usage

```
data(ext)
```

```
data(ext_eur)
```

Format

sf object

An object of class sf (inherits from `data.frame`) with 1 rows and 3 columns.

defaults	<i>Set, get and reset basemaps defaults</i>
----------	---

Description

These functions set, get or reset the defaults of all map arguments passed to [basemap](#) and associated functions.

Usage

```
set_defaults(
    ext = NULL,
    map_service = NULL,
    map_type = NULL,
    map_res = NULL,
    map_token = NULL,
    map_dir = NULL
)

get_defaults()

reset_defaults()
```

Arguments

ext	extent to be covered by the basemap as any spatial class supported by <code>st_bbox</code> .
map_service	character, a map service, either "osm", "carto" or "mapbox". Default is "osm".
map_type	character, a map type, e.g. "streets". For a full list of available map types, see get_maptypes .
map_res	numeric, resolution of base map in range from 0 to 1.
map_token	character, authentication token for services that require registration, which are "osm_stamen", "osm_stadia", "osm_thunderforest" and "mapbox". Register at https://stadiamaps.com/ (for stamen and stadia), https://www.thunderforest.com/ and/or https://www.mapbox.com/ to get tokens. Ignored for all other map services.
map_dir	character, cache directory where downloaded basemap tiles will be stored. By default, a temporary directory is used, which is destroyed when the session is terminated.

Value

For `get_defaults`, a list of defaults, otherwise none.

Examples

```
library(basemaps)
data(ext)

# set defaults for the basemap
set_defaults(ext = ext, map_service = "osm", map_type = "terrain_bg")

# get defaults
get_defaults()

## Not run:
# load and return basemap map as raster (default)
map <- basemap()

## End(Not run)

# reset defaults
reset_defaults()
```

draw_ext

Draw extent

Description

This function lets you draw an extent on an interactive map. It is a simple wrapper around `mapedit::drawFeatures()` written by Tim Appelhans et al.

Usage

```
draw_ext()
```

Value

An sf object

Examples

```
## Not run:
library(basemaps)

# draw extent interactively
ext <- draw_ext()

# set defaults for the basemap
set_defaults(ext = ext, map_service = "osm", map_type = "terrain_bg")
# for mapbox maps, you need a map_token. Register for free at mapbox.com to get a token

# load and return basemap map as raster (default)
```

```
map <- basemap()

## End(Not run)
```

get_maptypes

Get all supported map types

Description

This function returns every supported map type that can be used as input to the `map_type` argument of `set_defaults`, `basemap` or associated functions.

Usage

```
get_maptypes(map_service = NULL)
```

Arguments

`map_service` character, optional, either "osm", "osm_stamen", "osm_stadia", "osm_thunderforest", "carto", "mapbox" or "esri". Otherwise, a list of map types for both services is returned.

Value

A character vector of supported map types

Source

"osm": Open Street Map contributors (<https://www.openstreetmap.org/copyright>), Open Topo Map (<https://opentopomap.org/>), Martin Tesar (<http://mtbmap.cz/>)

"osm_stamen": Stamen (<https://maps.stamen.com/>) via Stadia Maps (<https://stadiamaps.com/>), Open Street Map contributors (<https://www.openstreetmap.org/copyright>)

"osm_stadia": Stadia Maps (<https://stadiamaps.com/>), Open Street Map contributors (<https://www.openstreetmap.org/copyright>)

"osm_thunderforest": Thunderforest (<https://www.thunderforest.com/>), Open Street Map contributors (<https://www.openstreetmap.org/copyright>)

"carto": Carto (<https://carto.com/>)

"mapbox": Mapbox (<https://www.mapbox.com>)

"esri": Esri (<https://www.esri.com/en-us/home>)

See Also

[basemap](#)

Examples

```
# for all services
get_maotypes()

# for osm only
get_maotypes("osm")
# or
get_maotypes()$osm

# for mapbox only
get_maotypes("mapbox")
# or
get_maotypes()$mapbox

# same for all other map services
```

plot

Plot raster objects using ggplot

Description

This function plots objects of class `SpatRaster`, `RasterLayer`, `RasterBrick` or `RasterStack` as `ggplot2`. It is used internally by `basemap*` functions that return `ggplot` plots.

Usage

```
gg_raster(r, r_type = "RGB", gglayer = F, ...)
```

Arguments

<code>r</code>	raster of class <code>SpatRaster</code> , <code>RasterLayer</code> , <code>RasterBrick</code> or <code>RasterStack</code> .
<code>r_type</code>	character, either "gradient" or "discrete".
<code>gglayer</code>	logical, if FALSE (default), a <code>ggplot2</code> plot is returned, if TRUE, a <code>ggplot2</code> layer is returned.
<code>...</code>	additional arguments, including <ul style="list-style-type: none"> • <code>maxpixels</code>, numeric, maximum number of pixels to be plotted (default: 500000) • <code>alpha</code>, numeric between 0 and 1, alpha value of the plotted data (transparency). • <code>maxColorValue</code>, numeric, the value to use as colour maximum. • <code>interpolate</code>, logical, whether to smooth the plot (default is TRUE).

Value

A `ggplot2` object

Examples

```
library(basemaps)

# example extent
data(ext)

## Not run:
# raster object: Brick
map <- basemap_raster(ext)

# plotting RasterBrick
gg_raster(map, r_type = "RGB")

## End(Not run)
```

Index

* datasets

data, [7](#)

basemap, [2](#), [7](#), [8](#), [10](#)

basemap_geotif (basemap), [2](#)

basemap_gglayer (basemap), [2](#)

basemap_ggplot (basemap), [2](#)

basemap_magick (basemap), [2](#)

basemap_mapview (basemap), [2](#)

basemap_plot (basemap), [2](#)

basemap_png (basemap), [2](#)

basemap_raster (basemap), [2](#)

basemap_stars (basemap), [2](#)

basemap_terra (basemap), [2](#)

cache, [6](#)

data, [7](#)

defaults, [8](#)

draw_ext, [9](#)

ext (data), [7](#)

ext_eur (data), [7](#)

flush_cache (cache), [7](#)

get_defaults (defaults), [8](#)

get_maptypes, [5](#), [8](#), [10](#)

gg_raster (plot), [11](#)

plot, [11](#)

reset_defaults (defaults), [8](#)

set_defaults, [2](#), [10](#)

set_defaults (defaults), [8](#)