

Package ‘basemaps’

July 24, 2026

Type Package
Title Accessing Spatial Basemaps in R
Version 0.1.1
Depends R (>= 3.5.0)
Date 2026-07-23
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, magick, terra, pbapply, utils, grDevices, methods
Suggests stars, raster, ggplot2, png, mapview, mapedit, testthat, covr
BugReports <https://github.com/16eagle/basemaps/issues>
URL <https://jakob.schwalb-willmann.de/basemaps/>
Config/roxygen2/version 8.0.0
NeedsCompilation no
Author Jakob Schwalb-Willmann [aut, cre] (ORCID: <https://orcid.org/0000-0003-2665-1509>)
Maintainer Jakob Schwalb-Willmann <dev@schwalb-willmann.de>
Repository CRAN
Date/Publication 2026-07-24 08:00:02 UTC

Contents

basemap	2
cache	7
data	7
defaults	8
draw_ext	9
maptypes	10
plot	12

Index**14**

basemap	<i>Get a spatial basemap</i>
---------	------------------------------

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

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", "maptiler" and "mapbox". Register at https://stadiamaps.com/ (for stamen and stadia), https://www.thunderforest.com/ , https://www.maptiler.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.
class	character, output class, either either plot (default), magick, png, geotif or if suggested packages are installed, terra, raster, stars, ggplot, gglayer or mapview.
force	logical, whether to force download over cached files or not. Default is FALSE.
...	additional arguments, including • browse, logical, for class = "png" and interactive sessions only. Whether to open the png file in the system's default PNG viewer or not. Default is TRUE. • col, character vector of colours for transforming single-layer basemaps into RGB, if class = "png" or class = "magick". Default is <code>topo.colors(25)</code>. • dpi, 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. • etc. (see <code>?gg_raster</code> for valid arguments when using class = "gglayer" or class = "ggplot", including <code>maxpixels</code> to control resolution of ggplot outputs
verbose	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_maptypes()

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

## Not run:
# load and plot basemap (default)
map <- basemap(ext)

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

# or as files:
basemap_geotif(ext)
basemap_png(ext)

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

# 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)
```

cache	<i>Flush basemaps cache</i>
-------	-----------------------------

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", "maptiler" and "mapbox". Register at https://stadiamaps.com/ (for stamen and stadia), https://www.thunderforest.com/ , https://www.maptiler.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)
```

maptypes

Get supported map types or add new map types

Description

These functions manage map types.

Usage

```
get_maptypes(map_service = NULL, as_df = FALSE, url_cols = FALSE)

add_maptypes(
  map_service,
  map_type,
  url_endpoint,
  url_xy = "xy",
  url_file_format = ".png",
  url_map_token = as.character(NA),
  auth_error_code = 401,
  http_version = 0,
  url_website = as.character(NA)
)

save_maptypes(file)

load_maptypes(file)
```

Arguments

map_service	character. - for <code>get_maptypes()</code>: optional, either "osm", "osm_stamen", "osm_stadia", "osm_thunderforest", "carto", "mapbox", "esri" or "maptiler". Otherwise, a list of map types for both services is returned. - for <code>add_maptypes()</code>: character, name of map service the map type to add belongs to
as_df	logical, whether to return a data.frame instead of a list (defaults to FALSE).
url_cols	logical, whether to include the endpoint URL and auxiliary URL columns (defaults to FALSE). Ignored, if as_df = FALSE.
map_type	character, name of map type to add.
url_endpoint	character, endpoint URL for map service and type to add.

<code>url_xy</code>	character, either "xy" or "yx" defining the order the map service to add is expecting tile subscripts in (defaults to "xy").
<code>url_file_format</code>	character, file format the endpoint to add is using (defaults to ".png").
<code>url_map_token</code>	character, optional, request query string used by the endpoint to add for transmitting authentication tokens if required (defaults to NA). Run <code>get_maptypes(as_df = T, url_cols = T)\$url_map_token</code> for examples.
<code>auth_error_code</code>	numeric, http error code the endpoint to add uses if authentication failed (defaults to 401).
<code>http_version</code>	numeric, optional, HTTP version used by the map service (defaults to 0 for auto setting).
<code>url_website</code>	character, optional, website URL for the service to add the user is directed to for registering (defaults to NA).
<code>file</code>	character, file name ending on ".csv" that the map types table is saved to or loaded from.

Details

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

`add_maptypes()` adds custom user-defined map types.

`save_maptypes()` saves the current map types table (as printed by `get_maptypes(as_df = T, url_cols = T)`) to a file.

`load_maptypes()` loads a map types table from file.

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/>)

"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>)

"maptiler": Esri (<https://www.maptiler.com>)

See Also[basemap](#)**Examples**

```
# for all services, as list (default:)
get_maptypes()

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

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

# same for all other map services

# or as data.frame:
get_maptypes(as_df = TRUE)

# or as data.frame including all url columns (endpoints):
get_maptypes(as_df = TRUE, url_cols = TRUE)

# add a custom map service and type yourself:
add_maptypes(
  map_service = "someservice", map_type = "terrain",
  url_endpoint = "https://tile.someservice.org")

# control further aspects of a custom map service and type:
add_maptypes(
  map_service = "someservice", map_type = "terrain",
  url_endpoint = "https://tile.someservice.org",
  url_xy = "xy", # order in which this service expects tile x and y id
  url_file_format = ".png",
  url_map_token = "?authtoken=", # query params for auth token
  auth_error_code = 401,
  url_website = "https://someservice.org"
)
```

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 • <code>maxpixels</code>, numeric, maximum number of pixels to be plotted (default: number of pixels in <code>r</code>). Use a value lower than <code>ncell(r)</code> to lower resolution for faster plotting. • <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:
# terra raster object
map <- basemap_terra(ext)

# plotting raster as ggplot using the with fill aesthetic
gg_raster(map, r_type = "RGB")

# or as gg layer using the with fill aesthetic
ggplot() + gg_raster(map, r_type = "RGB", gglayer = T) + scale_fill_identity()

## End(Not run)
```

Index

- * **datasets**
 - data, [7](#)
- add_maptypes (maptypes), [10](#)
- basemap, [2](#), [7](#), [8](#), [11](#), [12](#)
- 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, [7](#)
- 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](#)
- get_maptypes (maptypes), [10](#)
- gg_raster (plot), [12](#)
- load_maptypes (maptypes), [10](#)
- maptypes, [10](#)
- plot, [12](#)
- reset_defaults (defaults), [8](#)
- save_maptypes (maptypes), [10](#)
- set_defaults, [2](#), [11](#)
- set_defaults (defaults), [8](#)