

Package: bingmapr (via r-universe)

September 18, 2024

Type Package

Title Get Maps from Bing Maps Static Map API

Version 0.1.0.9000

Description Get static maps using aerial, bird's eye and other imagery from Bing Maps.

License MIT + file LICENSE

URL <https://github.com/elipousson/bingmapr>,
<https://elipousson.github.io/bingmapr/>

BugReports <https://github.com/elipousson/bingmapr/issues>

Imports cli (>= 2.5.0), httr2, magick, rlang (>= 1.0.0), sf, utils

Encoding UTF-8

Language en

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

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

RemoteUrl <https://github.com/elipousson/bingmapr>

RemoteRef HEAD

RemoteSha 62295c8f68760dc38d9f3a51a9236e487f18a10d

Contents

bingmap_image_ggplot	2
bing_maps_api_key	3
bing_static_map	4

Index	7
--------------	----------

bingmap_image_ggplot *Get map image and plot with magick::image_ggplot()*

Description

Wrapper for `magick::image_ggplot()` where all ... parameters are passed to `get_map_image()`.

Usage

```
bingmap_image_ggplot(..., interpolate = FALSE)
```

Arguments

... Arguments passed on to `get_map_image`

location A `sf`, `sfc`, or `bbox` object (centroid point is used for location) or numeric vector in format `c(latitude, longitude)`. Defaults to `NULL`.

query String with query for location. query is ignored if a location is provided. Defaults to `NULL`.

imagery String with imagery type, Default: 'BirdsEye' Supported values include:

- Aerial: Aerial imagery.
- AerialWithLabels: Aerial imagery with a road overlay.
- AerialWithLabelsOnDemand: Aerial imagery with on-demand road overlay.
- Streetside: Street-level imagery.
- BirdsEye: Birds Eye (oblique-angle) imagery.
- BirdsEyeWithLabels: Birds Eye (oblique-angle) imagery with a road overlay.
- Road: Roads without additional imagery.
- CanvasDark: A dark version of the road maps.
- CanvasLight: A lighter version of the road maps which also has some of the details such as hill shading disabled.
- CanvasGray: A grayscale version of the road maps

width,height,mapsize Width and height in pixels or use `mapsize` to provide a vector of `c(width, height)`. If `mapsize` is provided, width and height are ignored. Default: 600px width, 400px height, `mapsize` is `NULL`.

zoom Numeric vector between 0 and 20 for imagery other than Bird's Eye maps or 18 to 22 for Bird's Eye maps. Default: 18

orientation Orientation as a character string ("N", "E", "S", "W") or length 1 numeric vector (0,90,180,270). Other numeric orientations (from -360 to 720) are matched to the closest value, e.g. 35 to 0 or 75 to 90. Default: 0

nudge Numeric vector in the format, `c(meters to shift latitude, meters to shift longitude)`, e.g. `c(100, 0)` to shift center 100 meters in latitude. Defaults to `NULL`.

key Bing Maps API Key, Default: `Sys.getenv("BING_MAPS_API_KEY")`

	check	If TRUE, check the map metadata which returns an error if the image is unavailable. Default FALSE.
	strip	drop image comments and metadata
interpolate		passed to ggplot2::annotation_raster

bing_maps_api_key *Install a Bing Maps API Key in Your '.Renviron' File for Repeated Use*

Description

This function will add your CENSUS API key to your .Renviron file so it can be called securely without being stored in your code. After you have installed your key, it can be called any time with `get_bing_maps_api_key()`.

Usage

```
bing_maps_api_key(  
  key,  
  overwrite = FALSE,  
  install = FALSE,  
  default = "BING_MAPS_API_KEY"  
)  
  
get_bing_maps_api_key(  
  key = NULL,  
  default = "BING_MAPS_API_KEY",  
  call = caller_env()  
)
```

Arguments

key	The API key provided to you from the Bing Maps Dev Center formatted in quotes. A key can be acquired at https://www.bingmapsportal.com/
overwrite	If this is set to TRUE, it will overwrite an existing BING_MAPS_API_KEY that you already have in your .Renviron file.
install	if TRUE, will install the key in your .Renviron file for use in future sessions. Defaults to FALSE

bing_static_map *Get and plot Bing Static Maps*

Description

See the documentation on Bing Static Maps for reference: <https://docs.microsoft.com/en-us/bingmaps/rest-services/imagery/get-a-static-map>

Usage

```
req_bingmapr(  
  location = NULL,  
  query = NULL,  
  imagery = "BirdsEye",  
  width = 600,  
  height = 400,  
  mapsize = NULL,  
  zoom = 18,  
  orientation = 0,  
  nudge = NULL,  
  key = Sys.getenv("BING_MAPS_API_KEY"),  
  check = FALSE,  
  .perform = TRUE,  
  ...  
)
```

```
get_request_url(...)
```

```
get_map_image(  
  location = NULL,  
  query = NULL,  
  imagery = "BirdsEye",  
  width = 600,  
  height = 400,  
  mapsize = NULL,  
  zoom = 18,  
  orientation = 0,  
  nudge = NULL,  
  key = Sys.getenv("BING_MAPS_API_KEY"),  
  check = TRUE,  
  strip = TRUE  
)
```

```
get_map_meta(  
  location = NULL,  
  query = NULL,  
  imagery = "BirdsEye",
```

```

width = 600,
height = 400,
mapsize = NULL,
zoom = 18,
orientation = 0,
nudge = NULL,
key = Sys.getenv("BING_MAPS_API_KEY"),
bbox = FALSE
)

```

Arguments

location	A sf, sfc, or bbox object (centroid point is used for location) or numeric vector in format c(latitude, longitude). Defaults to NULL.
query	String with query for location. query is ignored if a location is provided. Defaults to NULL.
imagery	String with imagery type, Default: 'BirdsEye' Supported values include: <ul style="list-style-type: none"> • Aerial: Aerial imagery. • AerialWithLabels: Aerial imagery with a road overlay. • AerialWithLabelsOnDemand: Aerial imagery with on-demand road overlay. • Streetside: Street-level imagery. • BirdsEye: Birds Eye (oblique-angle) imagery. • BirdsEyeWithLabels: Birds Eye (oblique-angle) imagery with a road overlay. • Road: Roads without additional imagery. • CanvasDark: A dark version of the road maps. • CanvasLight: A lighter version of the road maps which also has some of the details such as hill shading disabled. • CanvasGray: A grayscale version of the road maps
width, height, mapsize	Width and height in pixels or use mapsize to provide a vector of c(width, height). If mapsize is provided, width and height are ignored. Default: 600px width, 400px height, mapsize is NULL.
zoom	Numeric vector between 0 and 20 for imagery other than Bird's Eye maps or 18 to 22 for Bird's Eye maps. Default: 18
orientation	Orientation as a character string ("N", "E", "S", "W") or length 1 numeric vector (0,90,180,270). Other numeric orientations (from -360 to 720) are matched to the closest value, e.g. 35 to 0 or 75 to 90. Default: 0
nudge	Numeric vector in the format, c(meters to shift latitude, meters to shift longitude), e.g. c(100, 0) to shift center 100 meters in latitude. Defaults to NULL.
key	Bing Maps API Key, Default: Sys.getenv("BING_MAPS_API_KEY")
check	If TRUE, check the map metadata which returns an error if the image is unavailable. Default FALSE.

<code>.perform</code>	If TRUE, return results from <code>httr2::req_perform()</code> . If FALSE, return request.
<code>...</code>	Additional parameters passed to <code>httr2::req_url_query()</code>
<code>strip</code>	drop image comments and metadata
<code>bbox</code>	If TRUE, return a <code>bbox</code> class object based on the bounding box values from the map metadata. If FALSE (default), return the full JSON metadata. (<code>get_map_meta()</code> only)

Details

Get API key from <https://www.bingmapsportal.com/>

Value

`get_request_url` returns the request URL for the Static Map API

`get_map_image` returns an image from `magick::image_read`

`get_map_meta` returns the JSON with the map metadata or a `bbox` for the map area

Index

bing_maps_api_key, 3
bing_static_map, 4
bingmap_image_ggplot, 2

get_bing_maps_api_key
 (bing_maps_api_key), 3
get_map_image, 2
get_map_image(bing_static_map), 4
get_map_image(), 2
get_map_meta(bing_static_map), 4
get_map_meta(), 6
get_request_url(bing_static_map), 4
ggplot2::annotation_raster, 3

httr2::req_perform(), 6
httr2::req_url_query(), 6

magick::image_ggplot(), 2

req_bingmapr(bing_static_map), 4