

Package: mapirosa (via r-universe)

September 9, 2024

Title Access OS OpenData APIs From R

Version 0.1.6

Date 2024-03-04

Description A package to enable accessing basemap tiles from the [OS Maps API](https://osdatahub.os.uk/docs/wmts/overview), initially. Access to other OS (Ordnance Survey) Data Hub APIs may follow. The primary motivation for this project was to find a map tile service that supplies tiles in the British National Grid (27700) CRS, unlike the usual web mapping standard CRS of 3857. The name of the package is an intentional corruption of the Spanish word mariposa .

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports assertthat, dplyr (>= 1.0.0), httr2, png, purrr, rlang, sf, snakecase, stringr, terra, tidyr, usethis, xml2

Suggests janitor, osmdata, testthat (>= 3.0.0), tibble, tmap

Config/testthat/edition 3

Depends R (>= 4.1.0)

Repository https://francisbarton.r-universe.dev

RemoteUrl https://github.com/francisbarton/mapirosa

RemoteRef HEAD

RemoteSha 80f0b7af7add1b8c9ea42033fc223f1fd3a0490f

Contents

build_basemap	2
check_zoom	3
create_bbox	3
generate_png_data	4

query_features_api	5
query_maps_api	6
query_names_api	6

Index	8
--------------	----------

build_basemap	<i>Build a basemap from the Ordnance Survey Maps API</i>
---------------	--

Description

Build a basemap from the Ordnance Survey Maps API

Usage

```
build_basemap(bbox, zoom, crs = c(27700, 3857), ...)
```

Arguments

bbox	A bbox object (a length 4 vector with names xmin, ymin, xmax, ymax)
zoom	Zoom level, an integer. For <code>crs = 27700</code> , this must be between 0 and 13. For <code>crs = 3857</code> , this must be between 7 and 20. Certain combinations of zoom level and style retrieve tiles from the OS's Premium service, which can generate a financial charge once past a certain level of usage. See the OS Maps API webpages for details.
crs	CRS code (EPSG), an integer, either 27700 (British National Grid) or 3857 (standard web mapping projected coordinate system eg Google Maps, OSM).
...	Other details to pass on to <code>generate_png_data()</code>

Value

A geospatially-referenced raster... hopefully.

Examples

```
oxf <- create_bbox("Oxford", 27700)
oxford_basemap <- build_basemap(oxf, zoom = 5, style = "road", crs = 27700)

tmap::tm_shape(oxford_basemap, raster.downsample = FALSE) +
  tmap::tm_rgb(max.value = 1)
```

check_zoom	<i>Internal check for a valid combination of CRS, map style and zoom level</i>
------------	--

Description

Internal check for a valid combination of CRS, map style and zoom level

Usage

```
check_zoom(
  zoom,
  style = c("outdoor", "road", "light", "leisure"),
  crs = c(27700, 3857),
  allow_premium = FALSE
)
```

Arguments

zoom	Zoom level, an integer. For <code>crs = 27700</code> , this must be between 0 and 13. For <code>crs = 3857</code> , this must be between 7 and 20. Certain combinations of zoom level and style retrieve tiles from the OS's Premium service, which can generate a financial charge once past a certain level of usage. See the OS Maps API webpages for details.
crs	CRS code (EPSG), an integer, either 27700 (British National Grid) or 3857 (standard web mapping projected coordinate system eg Google Maps, OSM).

Value

No visible return unless an error is thrown

create_bbox	<i>Use osmdata to easily get a bbox object for a given place name</i>
-------------	---

Description

Only the first search result from osmdata gets used, however

Usage

```
create_bbox(place, crs)
```

Arguments

place	string: place name to search for
crs	integer or string: the EPSG code for the desired CRS

<code>generate_png_data</code>	<i>Retrieve PNG data (tiles) to cover a bbox, and generate a list of extents for each tile</i>
--------------------------------	--

Description

Retrieve PNG data (tiles) to cover a bbox, and generate a list of extents for each tile

Usage

```
generate_png_data(
  bbox,
  zoom,
  crs,
  style = c("outdoor", "road", "light", "leisure"),
  squarify = TRUE,
  squarify_to = c("south", "east"),
  cache_tiles = FALSE,
  allow_premium = FALSE,
  debug = FALSE
)
```

Arguments

<code>bbox</code>	A bbox object (a length 4 vector with names xmin, ymin, xmax, ymax)
<code>zoom</code>	Zoom level, an integer. For <code>crs = 27700</code> , this must be between 0 and 13. For <code>crs = 3857</code> , this must be between 7 and 20. Certain combinations of zoom level and style retrieve tiles from the OS's Premium service, which can generate a financial charge once past a certain level of usage. See the OS Maps API webpages for details.
<code>crs</code>	CRS code (EPSG), an integer, either 27700 (British National Grid) or 3857 (standard web mapping projected coordinate system eg Google Maps, OSM).
<code>style</code>	Map style, a string. One of "road", "outdoor", "light", "leisure". Leisure is only available in the 27700 CRS.
<code>squarify</code>	Whether to add rows/columns to a basemap to make it square. TRUE by default. If a bbox covers a set of basemap tiles that is wider than it is high ("landscape"), <code>squarify</code> will add row(s) of tiles as necessary to make the basemap square. For a portrait basemap (higher than it is wide), <code>squarify</code> retrieves extra columns to make the basemap square.
<code>squarify_to</code>	Where <code>squarify</code> adds an odd number of columns or rows, one more will be added to one side than to the opposite side. In case you have a preference which side should receive more, you can stipulate this here. Needs a character vector. This is set to <code>c("south", "east")</code> by default. If you know that <code>squarify</code> will definitely add columns, say, you can stipulate a single string e.g. "west". If you are not sure whether it will add rows

or columns, then enter a vector of length 2, e.g. `c("north", "west")` depending on your preferences. Note that if 5 rows are added, 3 will be added to your preferred direction, and 2 to the other side; this parameter does not force all 5 rows to be added to your preferred side!

<code>cache_tiles</code>	Instead of converting PNG data from the API into a basemap on the fly, cache data as local PNG files. Not functional yet; hence set to <code>FALSE</code> .
<code>allow_premium</code>	Whether to only access tiles from zoom levels that are within the "Open-Data" tier of the OS API. These vary according to map style and CRS. See the API Technical Specification for details. Defaults to <code>FALSE</code> . Set to <code>TRUE</code> if you wish to access zoom levels within the "Premium" service tier (chargeable) <ul style="list-style-type: none"> • see https://osdatahub.os.uk/plans)
<code>debug</code>	Whether to show any errors that were received from the API. This package should handle errors gracefully in general, but if your basemap is not complete then you may wish to turn this on to see what errors there might be.

Value

a list of length 2: a list of PNG data and a list of extents for each tile

`query_features_api` *Retrieve data from the OS Features API*

Description

Function not yet completed and functional. See <https://osdatahub.os.uk/docs/wfs/technicalSpecification> for details.

Usage

```
query_features_api(
  x,
  version = c("2.0.0", "1.1.0", "1.0.0"),
  user_agent = NULL
)
```

Arguments

<code>x</code>	character. The name of a place, or a postcode for example. The API docs say 'A free string text search of OS Names, intended to be an ambiguous/fuzzy search.'
<code>version</code>	character. API version. Most recent (2.0.0) by default.
<code>user_agent</code>	character. <code>NULL</code> by default, which will use the package name.

Value

A raw response from the API endpoint - you will want to process this.

query_maps_api	<i>Build and perform a query to the OS Maps API</i>
----------------	---

Description

Build and perform a query to the OS Maps API

Usage

```
query_maps_api(x, y, zoom, style, crs, user_agent = NULL)
```

Arguments

<code>x</code>	Tile number (horizontally)
<code>y</code>	Tile number (vertically)
<code>zoom</code>	Zoom level, an integer. For <code>crs = 27700</code> , this must be between 0 and 13. For <code>crs = 3857</code> , this must be between 7 and 20. Certain combinations of zoom level and style retrieve tiles from the OS's Premium service, which can generate a financial charge once past a certain level of usage. See the OS Maps API webpages for details.
<code>crs</code>	CRS code (EPSG), an integer, either 27700 (British National Grid) or 3857 (standard web mapping projected coordinate system eg Google Maps, OSM).
<code>user_agent</code>	A User-Agent string to pass to the API

Value

Raw PNG data

query_names_api	<i>Retrieve data from the OS Names API</i>
-----------------	--

Description

Only uses 'find' method currently, not 'nearest' yet. See <https://osdatahub.os.uk/docs/names/technicalSpecification> for details.

Usage

```
query_names_api(
  x,
  local_types = NULL,
  bounds = NULL,
  max_results = 1L,
  user_agent = NULL
)
```

Arguments

<code>x</code>	character. The name of a place, or a postcode for example. The API docs say 'A free string text search of OS Names, intended to be an ambiguous/fuzzy search.'
<code>local_types</code>	character vector. Types of result to return. Examples are: 'City', 'Village'. See API docs for all options. <code>NULL</code> by default (= no filtering by type).
<code>bounds</code>	bbox. The bounding box of an area to limit search to. <code>NULL</code> by default - no filtering by area. Can be supplied as a vector of 4 numerics, in the following XMIN,YMIN,XMAX,YMAX form: <code>c(414000, 114000, 414100, 114100)</code>
<code>max_results</code>	integer. How many results to return. Can be from 1-100. Set to 1 by default.
<code>user_agent</code>	character. <code>NULL</code> by default, which will use the package name.

Value

An sfc geospatial tibble

Examples

```
query_names_api(c("Stroud", "Gloucester"))
```

Index

`build_basemap`, 2

`check_zoom`, 3

`create_bbox`, 3

`generate_png_data`, 4

`query_features_api`, 5

`query_maps_api`, 6

`query_names_api`, 6