Package: azuremapsr (via r-universe)

May 27, 2026

Title Interface to the 'Azure Maps' API

Version 0.0.2.9000

Description Provides a wrapper for the Microsoft 'Azure Maps' REST APIs <https://learn.microsoft.com/en-us/rest/api/maps/route?view=rest-maps-2025-01-01>, enabling users to access mapping and geospatial services directly from R. This package simplifies authenticating, building, and sending requests for services like route directions. It handles conversions between R objects (such as 'sf' objects) and the GeoJSON+JSON format required by the API, making it easier to integrate 'Azure Maps' into R-based data analysis workflows.

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/juanfonsecaLS1/azuremapsr

BugReports https://github.com/juanfonsecaLS1/azuremapsr/issues

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports geojsonsf (>= 2.0.3), httr2 (>= 1.2.1), jsonlite (>= 2.0.0), lubridate (>= 1.9.4), purrr (>= 1.1.0), rlist (>= 0.4.6.2), sf (>= 1.0.21), stringr (>= 1.5.1)

Depends R (>= 4.1.0)

LazyData true

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://juanfonsecals1.r-universe.dev

Date/Publication 2025-09-29 14:03:37 UTC

RemoteUrl https://github.com/juanfonsecals1/azuremapsr

RemoteRef HEAD

RemoteSha 1026a551c3b90e814d79b8f41c7da4fd1e9da810

Contents

`check_params` *check conformity of parameters for JSON section*

Description

check conformity of parameters for JSON section

Usage

```
check_params(test_params, template_params, tz)
```

Arguments

`test_params`	list of parameters from input
`template_params`	list of parameters hardcoded in package
`tz`	timezone from input

Value

No return value, called for side effects

Examples

```
## Not run:
check_params(params,template_params,"UTC")

## End(Not run)
```

get_azuremaps_token *Get Azure Maps API Authentication Token*

Description

Retrieves the Azure Maps API token from the environment.

Usage

```
get_azuremaps_token()
```

Value

A character string containing the Azure Maps API token.

Examples

```
## Not run:
get_azuremaps_token()

## End(Not run)
```

get_point *Create sfc Object from Coordinates or sf POINT Object*

Description

Converts a pair of coordinates (numeric vector), a matrix of coordinates, or an sf/sfc POINT object into an sfc object for use in GeoJSON bodies. Only POINT geometries are supported. The output is always in EPSG:4326.

Usage

```
get_point(x, multiple = FALSE)

## Default S3 method:
get_point(x, multiple = FALSE)

## S3 method for class 'numeric'
get_point(x, multiple = FALSE)

## S3 method for class 'matrix'
get_point(x, multiple = FALSE)

## S3 method for class 'sf'
get_point(x, multiple = FALSE)
```

```
## S3 method for class 'sfc'
get_point(x, multiple = FALSE)
```

Arguments

x	A numeric vector of length 2, a matrix with two columns (coordinates), or an sf/sfc object of POINT type.
multiple	Logical; if TRUE, allows handling of multiple features (e.g., when input is an sfc or sf object with more than one POINT). Default is FALSE.

Value

An sfc object with coordinates in EPSG:4326.

Examples

```
get_point(c(-122.201399, 47.608678))
get_point(
  matrix(
    c(-122.201399, 47.608678, -122.202, 47.609),
    ncol = 2,
    byrow = TRUE
  ),
  multiple = TRUE
)
library(sf)
pt <- st_sf(
  geometry = st_sfc(st_point(c(-122.201399, 47.608678)), crs = 4326)
)
get_point(pt)
```

---

get_routes *Extract and Combine Routes from an 'Azure Maps' Response*

Description

This function takes a successful response object from the 'Azure Maps' API, extracts the main route and any alternative routes, and combines them into a single `sf` object.

Usage

```
get_routes(resp)
```

Arguments

resp	An `httr2_response` object, typically from a successful call to `req_route_directions`.

Value

An sf object containing the combined main and alternative routes. If the request was not successful (status code is not 200), the function will stop with an error.

Examples

```
## Not run:
# Assuming 'response' is a successful response from req_route_directions
all_routes_sf <- get_routes(response)
plot(sf::st_geometry(all_routes_sf))

## End(Not run)
```

---

json_to_sf *Convert 'Azure Maps' JSON Response to an sf Object*

Description

This function processes a JSON response body from the Azure Maps API, extracts the route information, and converts it into a spatial (sf) object.

Usage

```
json_to_sf(body, main_route = TRUE, linestring = TRUE)
```

Arguments

body	A list, typically the parsed JSON response from an httr2 request.
main_route	A logical value. If TRUE (the default), only the main route is processed. If FALSE, alternative routes are processed instead.
linestring	A logical value. If TRUE (the default), it filters for LineString geometries (the route path).

Value

An sf object containing the spatial features from the route response, or NULL if no valid features are found.

Examples

```
## Not run:
# Assuming 'resp' is an httr2 response object from req_route_directions
body <- httr2::resp_body_json(resp)
route_sf <- json_to_sf(body)
plot(sf::st_geometry(route_sf))

## End(Not run)
```

POSTbody_builder_directions_geojson

Build GeoJSON Body for Route Directions

Description

Constructs the GeoJSON part of the request body for the Azure Maps Route Directions API. This includes the origin, destination, and any waypoints.

Usage

```
POSTbody_builder_directions_geojson(origin, destination, waypoints = NULL)
```

Arguments

origin	A numeric vector of coordinates (longitude, latitude) or an sf object representing the starting point.
destination	A numeric vector of coordinates (longitude, latitude) or an sf object representing the end point.
waypoints	Optional. A numeric vector, a matrix of coordinates, or an sf object with POINT geometries for intermediate stops.

Value

A list formatted as a GeoJSON FeatureCollection, ready to be included in the API request body.

Examples

```
## Not run:
origin <- c(-122.201399, 47.608678)
destination <- c(-122.201669, 47.615076)
waypoints <- c(-122.20687, 47.612002)
geojson_part <- POSTbody_builder_directions_geojson(origin, destination, waypoints)

## End(Not run)
```

POSTbody_builder_directions_json

Build JSON Parameter Body for Route Directions

Description

Constructs the JSON part of the request body containing routing parameters for the Azure Maps Route Directions API.

Usage

```
POSTbody_builder_directions_json(params, tz)
```

Arguments

params	A list of routing parameters, such as travelMode, routeType, departAt, etc.
tz	A string specifying the timezone for any date-time parameters.

Value

A list of routing parameters, with values formatted and unboxed as required for the JSON request.

Examples

```
## Not run:
params <- list(
  travelMode = "car",
  routeType = "fastest"
)
json_part <- POSTbody_builder_directions_json(params, "UTC")

## End(Not run)
```

---

req_route_directions *Get Route Directions from 'Azure Maps'*

Description

Requests route directions from 'Azure Maps' API using origin, destination, waypoints, and route parameters.

Usage

```
req_route_directions(
  origin,
  destination,
  waypoints = NULL,
  params,
  tz = Sys.timezone(),
  api_key = get_azuremaps_token(),
  api_version = "2025-01-01"
)
```

Arguments

origin	A numeric vector of length 2 with origin coordinates (longitude, latitude), or an sf object with a single POINT geometry.
destination	A numeric vector of length 2 with destination coordinates (longitude, latitude), or an sf object with a single POINT geometry.
waypoints	Optional. A numeric vector, a matrix of coordinates, or an sf object with POINT geometries representing intermediate stops.
params	A list of route parameters (e.g., optimizeRoute, routeOutputOptions, maxRouteCount, travelMode). See the API documentation
tz	A string specifying the timezone. Defaults to the system's timezone.
api_key	The 'Azure Maps' API key. Defaults to the value retrieved by get_azuremaps_token().
api_version	The API version to use. Defaults to "2025-01-01".

Value

An httr2_response object from the 'Azure Maps' API.

Examples

```
## Not run:
origin <- c(-122.201399, 47.608678)
destination <- c(-122.201669, 47.615076)
waypoints <- c(-122.20687, 47.612002)

params <- list(
  optimizeRoute = "fastestWithTraffic",
  routeOutputOptions = "routePath",
  maxRouteCount = 3,
  travelMode = "driving"
)

response <- req_route_directions(origin, destination, waypoints, params)

## End(Not run)
```

---

sample_response *A sample response obtained from the* get_route_directions *function*

Description

A sample response obtained from the get_route_directions function

Format

a httr2 response object

References

Sample of the call based on the API documentation in https://learn.microsoft.com/en-us/rest/api/maps/route/post-route-directions?view=rest-maps-2025-01-01&tabs=HTTP#examples

---

`sample_response_leeds` *A sample response obtained from the* `get_route_directions` *function*

Description

A sample response obtained from the `get_route_directions` function

Format

a `httr2` response object

`set_azuremaps_token` *Set Azure Maps API Authentication Token*

Description

Saves an authentication token for the Azure Maps API in the environment.

Usage

```
set_azuremaps_token(token)
```

Arguments

`token`	A character string containing the Azure Maps API token.

Value

Logical TRUE if the token is correctly set.

Examples

```
## Not run:
set_azuremaps_token("your_token_here")

## End(Not run)
```

Index