Package ‘ohsome’

February 22, 2023

Title  An ‘ohsome API’ Client

Version  0.2.1

Description  A client that grants access to the power of the ‘ohsome API’ from R. It lets you analyze the rich data source of the ‘OpenStreetMap (OSM)’ history. You can retrieve the geometry of ‘OSM’ data at specific points in time, and you can get aggregated statistics on the evolution of ‘OSM’ elements and specify your own temporal, spatial and/or thematic filters.

License  LGPL (>= 3)


BugReports  https://github.com/GIScience/ohsome-r/issues

Depends  R (>= 2.10)

Imports  geojsonsf, httr, jsonlite, readr, sf, utils

Suggests  dplyr, ggplot2, httptest, janitor, knitr, mapview, nominatimlite, osmdata, rmarkdown, spelling, testthat (>= 3.0.0), tmaptools, tidyr

VignetteBuilder  knitr

Config/testthat/edition  3

Encoding  UTF-8

Language  en-US

LazyData  true

RoxygenNote  7.2.2

NeedsCompilation  no

Author  Heidelberg Institute for Geoinformation Technology (HeiGIT) gGmbH [cph].

Oliver Fritz [aut, cre] (https://orcid.org/0000-0001-6324-7295)

Maintainer  Oliver Fritz <oliver.fritz@heigit.org>

Repository  CRAN

Date/Publication  2023-02-22 14:50:02 UTC
ohsome_aggregate_elements

Aggregate OSM elements

Description

Creates an ohsome_query object for OSM element aggregation

Usage

ohsome_aggregate_elements(
  boundary = NULL,
  aggregation = c("count", "length", "perimeter", "area"),
  return_value = c("absolute", "density", "ratio"),
  grouping = NULL,
  time = NULL,
...
)

ohsome_elements_count(boundary = NULL, ...)

ohsome_elements_length(boundary = NULL, ...)

ohsome_elements_perimeter(boundary = NULL, ...)

ohsome_elements_area(boundary = NULL, ...)
Arguments

boundary  Bounding geometries specified by WGS84 coordinates in the order lon, lat. The geometries of sf are transformed to WGS84 if the CRS of the object is known. The following classes are supported:

- sf with (MULTI)POLYGON geometries
- sfc with (MULTI)POLYGON geometries
- sfg with (MULTI)POLYGON geometries and WGS 84 coordinates
- bbox created with sf::st_bbox() or tmaptools::bb()
- matrix created with sp::bbox() or osmdata::getbb()
- character providing textual definitions of bounding polygons, boxes or circles as allowed by the ohsome API (see ohsome API - Boundaries):
  - bboxes: WGS84 coordinates in the following format: "id1:lon1,lat1,lon2,lat2|id2:lon1,lat1,lon2,lat2|..." OR "lon1,lat1,lon2,lat2|lon1,lat1,lon2,lat2|..."
  - bcircles: WGS84 coordinates + radius in meter in the following format: "id1:lon,lat,r|id2:lon,lat,r|..." OR "lon,lat,r|lon,lat,r|...
  - bpolys: WGS84 coordinates given as a list of coordinate pairs (as for bboxes) or GeoJSON FeatureCollection. The first point has to be the same as the last point and MultiPolygons are only supported in GeoJSON.
- list of bbox, matrix or character. Bounding geometry types of all list elements must be the same. Does not work with GeoJSON FeatureCollections.

aggregation character; aggregation type:

- "count" returns the total number of elements. This is the default.
- "length" returns the total length of elements in meters.
- "perimeter" returns the total perimeter of elements in meters.
- "area" returns the total area of elements in square meters.

return_value character; the value to be returned by the ohsome API:

- "absolute" returns the absolute number, length, perimeter or area of elements. This is the default.
- "density" returns the number, length, perimeter or area (in meters!) of elements per square kilometer.
- "ratio" returns an absolute value for elements satisfying the filter argument, an absolute value2 for elements satisfying the filter2 argument, and the ratio of value2 to value.

grouping character; group type(s) for grouped aggregations (only available for queries to aggregation endpoints). The following group types are available:

- "boundary" groups the result by the given boundaries that are defined through any of the boundary query parameters.
- "key" groups the result by the given keys that are defined through the groupByKey query parameter.
- "tag" groups the result by the given tags that are defined through the groupByKey and groupByValues query parameters.
- "type" groups the result by OSM element type.
- c("boundary", "tag") groups the result by the given boundaries and tags.

Not all of these group types are accepted by all of the aggregation endpoints. Check Grouping for available group types.

time - character; time parameter of the query (see Supported time formats).

... Parameters of the request to the ohsome API endpoint.

Details

ohsome_aggregate_elements() creates an ohsome_query object for OSM element aggregation. ohsome_elements_count(), ohsome_elements_length(), ohsome_elements_perimeter() and ohsome_elements_area() are wrapper functions for specific aggregation endpoints. Boundary objects are passed via set_boundary() into ohsome_boundary().

Value

An ohsome_query object. The object can be sent to the ohsome API with ohsome_post(). It consists of the following elements:

- url: The URL of the endpoint.
- encode: The way the information is encoded and then posted to the ohsome API. Set as "form".
- body: The parameters of the query such as format, filter or bpolys.

See Also

ohsome API Endpoints - Elements Aggregation

Examples

# Count of breweries in Franconia
ohsome_aggregate_elements(
    mapview::franconia,
    aggregation = "count",
    filter = "craft=brewery",
    time = "2022-01-01"
)

ohsome_elements_count(
    mapview::franconia,
    filter = "craft=brewery",
    time = "2022-01-01"
)

# Monthly counts of breweries in Franconia from 2012 to 2022
ohsome_elements_count(
    mapview::franconia,
    filter = "craft=brewery",
    time = "2012/2022/P1M"
# Count of breweries per district of Franconia
```
ohsome_elements_count(
    mapview::franconia,
    filter = "craft=brewery",
    grouping = "boundary",
    time = "2022-01-01"
)
```

# Number of breweries per square kilometer
```
ohsome_elements_count(
    mapview::franconia,
    filter = "craft=brewery",
    return_value = "density",
    time = "2022-01-01"
)
```

# Proportion of breweries that are microbreweries
```
ohsome_elements_count(
    mapview::franconia,
    filter = "craft=brewery",
    filter2 = "craft=brewery and microbrewery=yes",
    return_value = "ratio",
    time = "2022-01-01"
)
```

# Total length of highway elements in Franconia
```
ohsome_elements_length(
    mapview::franconia,
    filter = "highway=* and geometry:line",
    time = "2022-01-01"
)
```

---

**ohsome_api_url**  
**ohsome API URL**

**Description**

The base URL of the ohsome API with path to current major version.

**Format**

A list:

- base: character; base URL
- version: character; path to current major API version
ohsome_boundary

Create an ohsome_boundary object

Description

Creates an ohsome_boundary object from various classes of input geometries. The ohsome_boundary object is used to set the bpolys, bboxes or bcircles parameter of an ohsome_query object.

Usage

```
ohsome_boundary(boundary, ...)
```

## S3 method for class 'quotesingle.Var'
ohsome_boundary(quotesingle.Var)

## S3 method for class 'character'
ohsome_boundary(quotesingle.Var)

## S3 method for class 'sf'
ohsome_boundary(quotesingle.Var, digits = 6, ...)

## S3 method for class 'sfc'
ohsome_boundary(quotesingle.Var, ...)

## S3 method for class 'sfg'
ohsome_boundary(quotesingle.Var, ...)

## S3 method for class 'bbox'
ohsome_boundary(quotesingle.Var, ...)

## S3 method for class 'matrix'
ohsome_boundary(quotesingle.Var, ...)

## S3 method for class 'list'
ohsome_boundary(quotesingle.Var, ...)

Arguments

boundary  Bounding geometries specified by WGS84 coordinates in the order lon, lat. The geometries of sf are transformed to WGS84 if the CRS of the object is known. The following classes are supported:

- sf with (MULTI)POLYGON geometries
- sfc with (MULTI)POLYGON geometries
- sfg with (MULTI)POLYGON geometries and WGS 84 coordinates
- bbox created with sf::st_bbox() or tmaptools::bb()
- matrix created with sp::bbox() or osmdata::getbb()
character providing textual definitions of bounding polygons, boxes or circles as allowed by the ohsome API (see ohsome API - Boundaries):

- bboxes: WGS84 coordinates in the following format: "id1:lon1,lat1,lon2,lat2|id2:lon1,lat1,lon2,lat2|..." OR "lon1,lat1,lon2,lat2|lon1,lat1,lon2,lat2|..."
- bcircles: WGS84 coordinates + radius in meter in the following format: "id1:lon,lat,r|id2:lon,lat,r|..." OR "lon,lat,r|lon,lat,r|..."
- bpolys: WGS84 coordinates given as a list of coordinate pairs (as for bboxes) or GeoJSON FeatureCollection. The first point has to be the same as the last point and MultiPolygons are only supported in GeoJSON.

- list of bbox, matrix or character. Bounding geometry types of all list elements must be the same. Does not work with GeoJSON FeatureCollections.

Additional arguments other than digits are ignored.

digits integer; number of decimal places of coordinates in the resulting GeoJSON when converting sf to GeoJSON (defaults to 6).

Value

An ohsome_boundary object which contains the following elements:

- boundary: the boundary in textual format
- type of the boundary (bpolys, bcircles, or bboxes).

Examples

# Definition of a bounding circle (lon,lat,radius in meters)
ohsome_boundary("8.6528,49.3683,1000")

# Definition of two named bounding circles
ohsome_boundary("Circle 1:8.6528,49.3683,1000|Circle 2:8.7294,49.4376,1000")

# Definition of two named circles with a character vector
ohsome_boundary(c("Circle 1:8.6528,49.3683,1000", "Circle 2:8.7294,49.4376,1000"))

# Use franconia from the mapview package as bounding polygons
ohsome_boundary(mapview::franconia, digits = 4)

# Use the bounding box of franconia
ohsome_boundary(sf::st_bbox(mapview::franconia))

# Get bounding box of the city of Berlin from OSM
## Not run:
ohsome_boundary(osmdata::getbb("Berlin"))

## End(Not run)
ohsome_contributions_count

Count OSM contributions

Description

Creates an ohsome_query object for OSM contributions count

Usage

ohsome_contributions_count(
  boundary = NULL,
  latest = FALSE,
  return_value = c("absolute", "density"),
  time = NULL,
  ...
)

Arguments

boundary  Bounding geometries specified by WGS84 coordinates in the order lon, lat. The geometries of sf are transformed to WGS84 if the CRS of the object is known. The following classes are supported:
   • sf with (MULTI)POLYGON geometries
   • sfc with (MULTI)POLYGON geometries
   • sfg with (MULTI)POLYGON geometries and WGS 84 coordinates
   • bbox created with sf::st_bbox() or tmaptools::bb()
   • matrix created with sp::bbox() or osmdata::bbox()
   • character providing textual definitions of bounding polygons, boxes or circles as allowed by the ohsome API (see ohsome API - Boundaries):
     - bboxes: WGS84 coordinates in the following format: "id1:lon1,lat1,lon2,lat2|id2:lon1,lat1,lon2,lat2|...
     - bcircles: WGS4 coordinates + radius in meter in the following format: "id1:lon,lat,r|id2:lon,lat,r|...
     - bpolys: WGS84 coordinates given as a list of coordinate pairs (as for bboxes) or GeoJSON FeatureCollection. The first point has to be the same as the last point and MultiPolygons are only supported in GeoJSON.
**Details**

ohsome_contributions_count() creates an ohsome_query object for OSM element aggregation. Boundary objects are passed via set_boundary() into ohsome_boundary().

**Value**

An ohsome_query object. The object can be sent to the ohsome API with ohsome_post(). It consists of the following elements:

- **url**: The URL of the endpoint.
- **encode**: The way the information is encoded and then posted to the ohsome API. Set as "form".
- **body**: The parameters of the query such as format, filter or bpolys.

**See Also**

ohsome API Endpoints - Contributions Aggregation

**Examples**

```r
# Monthly counts of contributions to man-made objects around "Null Island"
ohsome_contributions_count("0,0,10", filter = "man_made=*", time = "2010/2020/P1Y")

# Monthly counts of latest contributions to man-made objects around "Null Island"
ohsome_contributions_count("0,0,10",
  latest = TRUE,
  filter = "man_made=*",
  time = "2010/2020/P1Y"
)
```
ohsome_extract_contributions

Description
Available ohsome API endpoints with their parameters

Format
A list of ohsome API endpoints.

ohsome_extract_contributions
Extract OSM contributions

Description
Creates an ohsome_query object for OSM contribution extraction

Usage

```r
ohsome_extract_contributions(
  boundary = NULL,
  geometryType = c("centroid", "bbox", "geometry"),
  latest = FALSE,
  time = NULL,
  properties = NULL,
  clipGeometry = TRUE,
  ...
)
```

```r
ohsome_contributions_bbox(boundary = NULL, ...)
```

```r
ohsome_contributions_centroid(boundary = NULL, ...)
```

```r
ohsome_contributions_geometry(boundary = NULL, ...)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boundary</td>
<td>Bounding geometries specified by WGS84 coordinates in the order lon, lat. The geometries of sf are transformed to WGS84 if the CRS of the object is known. The following classes are supported:</td>
</tr>
<tr>
<td></td>
<td>• sf with (MULTI)POLYGON geometries</td>
</tr>
<tr>
<td></td>
<td>• sfc with (MULTI)POLYGON geometries</td>
</tr>
<tr>
<td></td>
<td>• sfg with (MULTI)POLYGON geometries and WGS 84 coordinates</td>
</tr>
</tbody>
</table>
• bbox created with `sf::st_bbox()` or `tmaptools::bb()`
• matrix created with `sp::bbox()` or `osmdata::getbb()`
• character providing textual definitions of bounding polygons, boxes or circles as allowed by the ohsome API (see ohsome API - Boundaries):
  – `bbox`: WGS84 coordinates in the following format: "id1:lon1,lat1,lon2,lat2|id2:lon1,lat1,lon2,lat2|..."
  – `bcircles`: WGS84 coordinates + radius in meter in the following format: "id1:lon,lat,r|id2:lon,lat,r|...
  – `bpolys`: WGS84 coordinates given as a list of coordinate pairs (as for `bbox`) or GeoJSON FeatureCollection. The first point has to be the same as the last point and MultiPolygons are only supported in GeoJSON.
• list of bbox, matrix or character. Bounding geometry types of all list elements must be the same. Does not work with GeoJSON FeatureCollections.

```
geometryType character; type of geometry to be extracted:
  • "centroid",
  • "bbox" (bounding boxes), or
  • "geometry"
```

Caveat: Node elements are omitted from results in queries for bounding boxes.

```
latest logical; if TRUE, request only the latest contributions provided to each OSM element.
```

```
time character; time parameter of the query (see Supported time formats).
```

```
properties character; properties to be extracted with the contributions:
  • "tags", and/or
  • "metadata" (i.e. @changesetId, @lastEdit, @osmType, @version), and/or
  • "contributionTypes" (i.e. @creation, @tagChange, @deletion, and @geometryChange)
```

Multiple values can be provided as comma-separated character or as character vector. This defaults to NULL (provides @contributionChangesetId, @osmId and @timestamp).

```
clipGeometry logical; specifies whether the returned geometries should be clipped to the query’s spatial boundary
```

```
... Parameters of the request to the ohsome API endpoint.
```

Details

`ohsome_extract_contributions()` creates an ohsome_query object for OSM contribution extraction. `ohsome_contributions_bbox()`, `ohsome_contributions_centroid()` and `ohsome_contributions_geometry()` are wrapper functions for specific contributions extraction endpoints. Boundary objects are passed via `set_boundary()` into `ohsome_boundary()`.
Value

An `ohsome_query` object. The object can be sent to the `ohsome` API with `ohsome_post()`. It consists of the following elements:

- `url`: The URL of the endpoint.
- `encode`: The way the information is encoded and then posted to the `ohsome` API. Set as "form".
- `body`: The parameters of the query such as `format`, `filter` or `bpolys`.

See Also

`ohsome API Endpoints – Contributions Extraction`

Examples

```r
# Extract contributions to man-made objects around "Null Island" with metadata:
ohsome_contributions_geometry(
  "0,0,10",
  filter = "man_made=*",
  time = c("2021-01-01", "2022-01-01"),
  properties = "metadata"
)
```

`ohsome_extract_elements`

*Extract OSM elements*

Description

Create an `ohsome_query` object for OSM element extraction

Usage

```r
ohsome_extract_elements(
  boundary = NULL,
  geometryType = c("centroid", "bbox", "geometry"),
  time = NULL,
  properties = NULL,
  clipGeometry = TRUE,
  ...
)
```

```r
ohsome_elements_bbox(boundary = NULL, ...)
```

```r
ohsome_elements_centroid(boundary = NULL, ...)
```

```r
ohsome_elements_geometry(boundary = NULL, ...)
```
ohsome_extract_elements

Arguments

boundary

Bounding geometries specified by WGS84 coordinates in the order lon, lat. The geometries of sf are transformed to WGS84 if the CRS of the object is known. The following classes are supported:

• sf with (MULTI)POLYGON geometries
• sfc with (MULTI)POLYGON geometries
• sfg with (MULTI)POLYGON geometries and WGS 84 coordinates
• bbox created with sf::st_bbox() or tmaptools::bb()
• matrix created with sp::bbox() or osmdata::getbb()
• character providing textual definitions of bounding polygons, boxes or circles as allowed by the ohsome API (see ohsome API - Boundaries):
  – bboxes: WGS84 coordinates in the following format: "id1:lon1,lat1,lon2,lat2|id2:lon1,lat1,lon2,lat2|...
  OR "lon1,lat1,lon2, lat2|lon1,lat1,lon2, lat2|...
  – bcircles: WGS84 coordinates + radius in meter in the following format: "id1:lon, lat, r|id2:lon, lat, r|...
    OR "lon, lat, r|lon, lat, r|...
  – bpolys: WGS84 coordinates given as a list of coordinate pairs (as for bboxes) or GeoJSON FeatureCollection. The first point has to be the same as the last point and MultiPolygons are only supported in GeoJSON.
• list of bbox, matrix or character. Bounding geometry types of all list elements must be the same. Does not work with GeoJSON FeatureCollections.

geometryType

categorical; type of geometry to be extracted:

• "centroid",
• "bboxes" (bounding boxes), or
• "geometry"

Caveat: Node elements are omitted from results in queries for bounding boxes.

time

categorical; time parameter of the query (see Supported time formats).

properties

categorical; properties to be extracted with the features:

• "tags", and/or
• "metadata" (i.e. @changesetId, @lastEdit, @osmType, and @version)

Multiple values can be provided as comma-separated character or as character vector. This defaults to NULL (provides @osmId).

clipGeometry

logical; specifies whether the returned geometries should be clipped to the query's spatial boundary

Details

ohsome_extract_elements() creates an ohsome_query object for OSM element extraction. ohsome_elements_bbox(), ohsome_elements_centroid() and ohsome_elements_geometry() are wrapper functions for specific elements extraction endpoints. Boundary objects are passed via set_boundary() into ohsome_boundary().
Value

An ohsome_query object. The object can be sent to the ohsome API with ohsome_post(). It consists of the following elements:

- **url**: The URL of the endpoint.
- **encode**: The way the information is encoded and then posted to the ohsome API. Set as "form".
- **body**: The parameters of the query such as format, filter or bpolys.

See Also

ohsome API Endpoints – Elements Extraction

Examples

# Extract geometries, metadata and tags of man-made objects around "Null Island":
```r
ohsome_elements_geometry("0,0,10",
  filter = "man_made=*",
  time = "2022-01-01",
  properties = c("metadata", "tags")
)
```

ohsome_extract_elementsFullHistory

*Extract OSM elements’ full history*

Description

Creates an ohsome_query object for the extraction of OSM elements’ full history

Usage

```r
ohsome_extract_elementsFullHistory(boundary = NULL, 
  geometryType = c("centroid", "bbox", "geometry"), 
  time = NULL, 
  properties = NULL, 
  clipGeometry = TRUE, 
  ...
)
```

ohsome_elementsFullHistory_bbox(boundary = NULL, ...)

ohsome_elementsFullHistory_centroid(boundary = NULL, ...)

ohsome_elementsFullHistory_geometry(boundary = NULL, ...)
Arguments

boundary Bounding geometries specified by WGS84 coordinates in the order lon, lat. The geometries of sf are transformed to WGS84 if the CRS of the object is known. The following classes are supported:
- sf with (MULTI)POLYGON geometries
- sfrc with (MULTI)POLYGON geometries
- sfg with (MULTI)POLYGON geometries and WGS 84 coordinates
- bbox created with sf::st_bbox() or tmaptools::bb()
- matrix created with sp::bbox() or osmdata::getbb()
- character providing textual definitions of bounding polygons, boxes or circles as allowed by the ohsome API (see ohsome API - Boundaries):
  - bboxes: WGS84 coordinates in the following format: "id1:lon1,lat1,lon2,lat2|id2:lon1,lat1,lon2,lat2|..."
  - bcircles: WGS84 coordinates + radius in meter in the following format: "id1:lon,lat,r|id2:lon,lat,r|...
  - bpolys: WGS84 coordinates given as a list of coordinate pairs (as for bboxes) or GeoJSON FeatureCollection. The first point has to be the same as the last point and MultiPolygons are only supported in GeoJSON.
- list of bbox, matrix or character. Bounding geometry types of all list elements must be the same. Does not work with GeoJSON FeatureCollections.

geometryType character; type of geometry to be extracted:
- "centroid",
- "bboxes" (bounding boxes), or
- "geometry"

Caveat: Node elements are omitted from results in queries for bounding boxes.

time character; time parameter of the query (see Supported time formats).

properties character; properties to be extracted with the features:
- "tags", and/or
- "metadata" (i.e. @changesetId, @lastEdit, @osmType, and @version)

Multiple values can be provided as comma-separated character or as character vector. This defaults to NULL (provides @osmId).

clipGeometry logical; specifies whether the returned geometries should be clipped to the query's spatial boundary

... Parameters of the request to the ohsome API endpoint.

Details

ohsome_extract_elementsFullHistory() creates an ohsome_query object for OSM element full history extraction. ohsome_elementsFullHistory_bbox(), ohsome_elementsFullHistory_centroid() and ohsome_elementsFullHistory_geometry() are wrapper functions for specific elementsFullHistory extraction endpoints. Boundary objects are passed via set_boundary() into ohsome_boundary().
Value

An ohsome_query object. The object can be sent to the ohsome API with ohsome_post(). It consists of the following elements:

- **url**: The URL of the endpoint.
- **encode**: The way the information is encoded and then posted to the ohsome API. Set as "form".
- **body**: The parameters of the query such as format, filter or bpolys.

See Also

ohsome API Endpoints – Elements Full History Extraction

Examples

```r
# Extract full history of building geometries around Heidelberg main station:
ohsome_elementsFullHistory_geometry(
  boundary = "8.67542,49.40347,1000",
  time = "2012,2022",
  filter = "building=* and geometry:polygon",
  clipGeometry = FALSE
)
```

---

**ohsome_get_metadata**  
*GET metadata from ohsome API*

Description

Returns parsed metadata from ohsome API

Usage

```
ohsome_get_metadata(quiet = FALSE)
```

Arguments

- **quiet**: logical; suppresses message on data attribution, API version and temporal extent.

Details

ohsome_get_metadata() sends a GET request to the metadata endpoint of ohsome API and parses the response. The parsed metadata is silently returned.
**Value**

An `ohsome_metadata` object. This is a named list with the attributes `date`, `status code` (of the GET request) and the following list elements:

- `attribution`: character; URL and text of OSM data copyrights and attribution
- `apiVersion`: character; Version of the ohsome API
- `timeout`: numeric; limit of the processing time in seconds
- `extractRegion`:
  - `spatialExtent`: sfc_POLYGON; spatial boundary of the OSM data in the underlying OSHDB
  - `temporalExtent`: vector of ISO 8601 character; start and end of the temporal extent of OSM data in the underlying OSHDB
  - `replicationSequenceNumber`: numeric; precise state of the OSM data contained in the underlying OSHDB, expressed as the id of the last applied (hourly) diff file from Planet OSM

**See Also**

`ohsome API Endpoints – Metadata`

**Examples**

```r
## Not run:
ohsome_get_metadata()
## End(Not run)
```

**Description**

Metadata of the ohsome API that is requested on loading the package

**Format**

An `ohsome_metadata` object. This is a named list with the attributes `date`, `status_code` (of the GET request) and the following list elements:

- `attribution`: character; URL and text of OSM data copyrights and attribution
- `apiVersion`: numeric_version; Version of the ohsome API
- `timeout`: numeric; limit of the processing time in seconds
- `extractRegion`:
  - `spatialExtent`: sfc_POLYGON; spatial boundary of the OSM data in the underlying OSHDB
Parse content from an ohsome API response

Description

Extracts and parses the content from an ohsome API response.

Usage

```r
ohsome_parse(
  response,
  returnclass = c("default", "sf", "data.frame", "list", "character"),
  omit_empty = TRUE
)

ohsome_sf(response, omit_empty = TRUE)

ohsome_df(response, omit_empty = TRUE)
```

Arguments

- `response`: An `ohsome_response` object
- `returnclass`: character; one of the following:
  - "default" returns `sf` if the `ohsome_response` contains GeoJSON, or else a `data.frame`.
  - "sf" returns `sf` if the `ohsome_response` contains GeoJSON, else issues a warning and returns a `data.frame`.
  - "data.frame" returns a `data.frame`.
  - "list" returns a list.
  - "character" returns the ohsome API response body as text (JSON or semicolon-separated values)
- `omit_empty`: logical; omit features with empty geometries (only if `returnclass = "sf"`)

Details

`ohsome_parse()` parses an `ohsome_response` object into an object of the specified class. By default, this is an `sf` object if the ohsome API response contains GeoJSON data or a `data.frame` if it does not. `ohsome_sf()` and `ohsome_df()` wrapper functions for specific return classes.
Value

An sf object, a data.frame, a list or a character

Examples

```r
## Not run:
# Create and send a query to ohsome API
r <- ohsome_query("elements/centroid", filter = "amenity=*") |>
  set_boundary(osmdata::getbb("Heidelberg")) |>
  set_time("2021") |>
  set_properties("metadata") |>
  ohsome_post(parse = FALSE)

# Parse response to object of default class (here: sf)
ohsome_parse(r)

# Parse response to data.frame
ohsome_df(r)

# Parse response to sf
ohsome_sf(r)

## End(Not run)
```

### Description

Sends an `ohsome_query` object as a POST request to the ohsome API and returns the response.

### Usage

```r
ohsome_post(
  query,  # An ohsome_query object constructed with ohsome_query() or any of its wrapper functions
  parse = TRUE,  # logical; if TRUE, parse the ohsome API response with ohsome_parse()
  validate = TRUE,
  strict = validate,
  additional_identifiers = NULL,
  ...
)
```

### Arguments

- `query`  
  - An `ohsome_query` object constructed with `ohsome_query()` or any of its wrapper functions

- `parse`  
  - logical; if TRUE, parse the ohsome API response with `ohsome_parse()`
ohsome_query

validate logical; if TRUE, issues warning for invalid endpoint or invalid/missing query parameters.

strict logical; if TRUE, throws error on invalid query. Overrules validate argument TRUE.

additional_identifiers vector coercible to character; optional user agent identifiers in addition to "ohsome-r/{version}".

Value

An ohsome_response object if parse = FALSE, else an sf object, a data.frame, a list or a character

See Also

ohsome API documentation

Examples

```
## Not run:
# Get bounding box of the city of Berlin
bbberlin <- osmdata::getbb("Berlin")

# Query for cinema geometries within bounding box
q <- ohsome_elements_geometry(bbberlin, filter = "amenity=cinema")

# Send query to ohsome API and return sf by default
ohsome_post(q)

# Send query to ohsome API and return data.frame
ohsome_post(q, returnclass = "data.frame")

# Send query and return unparsed response
ohsome_post(q, parse = FALSE)

## End(Not run)
```

ohsome_query Create an ohsome_query object

Description

Creates an ohsome_query object specifying the ohsome API endpoint and the request parameters.
ohsome_query

Usage

ohsome_query(endpoint, boundary = NULL, grouping = NULL, ..., validate = FALSE)

Arguments

endpoint  The path to the ohsome API endpoint. Either a single string (e.g. "elements/count") or a vector of character in the right order (e.g. c("elements", "count")).

boundary Bounding geometries specified by WGS84 coordinates in the order lon,lat. The geometries of sf are transformed to WGS84 if the CRS of the object is known. The following classes are supported:

• sf with (MULTI)POLYGON geometries
• sfc with (MULTI)POLYGON geometries
• sfg with (MULTI)POLYGON geometries and WGS 84 coordinates
• bbox created with sf::st_bbox() or tmaptools::bb()
• matrix created with sp::bbox() or osmdata::getbb()
• character providing textual definitions of bounding polygons, boxes or circles as allowed by the ohsome API (see ohsome API - Boundaries):
  • bboxes: WGS84 coordinates in the following format: "id1:lon1,lat1,lon2,lat2|id2:lon1,lat1,lon2,lat2|..."
  • bcircles: WGS84 coordinates + radius in meter in the following format: "id1:lon,lat,r|id2:lon,lat,r|..." OR "lon,lat,r|lon,lat,r|..."
  • bpolys: WGS84 coordinates given as a list of coordinate pairs (as for bboxes) or GeoJSON FeatureCollection. The first point has to be the same as the last point and MultiPolygons are only supported in GeoJSON.
• list of bbox, matrix or character. Bounding geometry types of all list elements must be the same. Does not work with GeoJSON FeatureCollections.

grouping character; group type(s) for grouped aggregations (only available for queries to aggregation endpoints). The following group types are available:

• "boundary" groups the result by the given boundaries that are defined through any of the boundary query parameters.
• "key" groups the result by the given keys that are defined through the groupByKey query parameter.
• "tag" groups the result by the given tags that are defined through the groupByKey and groupByValues query parameters.
• "type" groups the result by OSM element type.
• c("boundary", "tag") groups the result by the given boundaries and tags.

Not all of these group types are accepted by all of the aggregation endpoints. Check Grouping for available group types.

... Parameters of the request to the ohsome API endpoint.

validate logical; if TRUE, issues warning for invalid endpoint or invalid/missing query parameters.
Value

An ohsome_query object. The object can be sent to the ohsome API with ohsome_post(). It consists of the following elements:

- **url**: The URL of the endpoint.
- **encode**: The way the information is encoded and then posted to the ohsome API. Set as "form".
- **body**: The parameters of the query such as format, filter or bpolys.

See Also

ohsome API documentation

Examples

```r
# Extract building geometries with manually set bboxes parameter
ohsome_query(
  "elements/geometry",
  bboxes = "8.6,49.36,8.75,49.44",
  time = "2022-01-01",
  filter = "building=*"
)

# Extract building geometries using a boundary object:
ohsome_query(
  "elements/geometry",
  boundary = "8.6,49.36,8.75,49.44",
  time = "2022-01-01",
  filter = "building=*"
)
```

ohsome_temporalExtent  ohsome API temporal extent

Description

Temporal extent of the OSM data in the underlying OSHDB

Format

A vector of POSIXct
ohsome_users_count  Count OSM users

Description

Create an ohsome_query object for OSM users count

Usage

```r
ohsome_users_count(
    boundary = NULL,
    return_value = c("absolute", "density"),
    grouping = NULL,
    time = NULL,
    ...
)
```

Arguments

- **boundary**: Bounding geometries specified by WGS84 coordinates in the order `lon, lat`. The geometries of `sf` are transformed to WGS84 if the CRS of the object is known. The following classes are supported:
  - `sf` with (MULTI)POLYGON geometries
  - `sfc` with (MULTI)POLYGON geometries
  - `sfg` with (MULTI)POLYGON geometries and WGS 84 coordinates
  - `bbox` created with `sf::st_bbox()` or `tmaptools::bb()`
  - `matrix` created with `sp::bbox()` or `osmdata::getbb()`
  - `character` providing textual definitions of bounding polygons, boxes or circles as allowed by the ohsome API (see [ohsome API - Boundaries](#)):
    - `bbox`: WGS84 coordinates in the following format: "id1:lon1,lat1,lon2,lat2|id2:lon1,..." or "lon1,lat1,lon2,lat2|lon1,lat1,lon2,lat2|..."
    - `bcircles`: WGS84 coordinates + radius in meter in the following format: "id1:lon,lat,r|id2:lon,lat,r|..." or "lon,lat,r|lon,lat,r|..."
    - `bpolys`: WGS84 coordinates given as a list of coordinate pairs (as for `bbox`) or GeoJSON FeatureCollection. The first point has to be the same as the last point and MultiPolygons are only supported in GeoJSON.
  - `list` of `bbox`, `matrix` or `character`. Bounding geometry types of all list elements must be the same. Does not work with GeoJSON FeatureCollections.

- **return_value**: character; the value to be returned by the ohsome API:
  - "absolute" returns the absolute number of users. This is the default.
  - "density" returns the number of users per square kilometer.

- **grouping**: character; group type(s) for grouped aggregations (only available for queries to aggregation endpoints). The following group types are available:
set_boundary

- "boundary" groups the result by the given boundaries that are defined through any of the boundary query parameters.
- "key" groups the result by the given keys that are defined through the groupByKeys query parameter.
- "tag" groups the result by the given tags that are defined through the groupByKey and groupByValues query parameters.
- "type" groups the result by OSM element type.
- c("boundary", "tag") groups the result by the given boundaries and tags.

Not all of these group types are accepted by all of the aggregation endpoints. Check Grouping for available group types.

time character; time parameter of the query (see Supported time formats).
...
Parameters of the request to the ohsome API endpoint.

Details

ohsome_users_count() creates an ohsome_query object for OSM users aggregation. Boundary objects are passed via set_boundary() into ohsome_boundary().

Value

An ohsome_query object. The object can be sent to the ohsome API with ohsome_post(). It consists of the following elements:

- url: The URL of the endpoint.
- encode: The way the information is encoded and then posted to the ohsome API. Set as "form".
- body: The parameters of the query such as format, filter or bpolys.

See Also

ohsome_api_endpoints – Users Aggregation

Examples

# Yearly count of users contributing to man-made objects around "Null Island"
ohsome_users_count("0,0,10", filter = "man_made=*", time = "2012/2022/P1Y")
set_boundary

Usage

set_boundary(query, boundary = NULL, ...)

Arguments

query An ohsome_query object constructed with ohsome_query() or any of its wrapper functions
boundary Bounding geometries specified by WGS84 coordinates in the order lon, lat. The geometries of sf are transformed to WGS84 if the CRS of the object is known. The following classes are supported:

- sf with (MULTI)POLYGON geometries
- sfc with (MULTI)POLYGON geometries
- sfg with (MULTI)POLYGON geometries and WGS 84 coordinates
- bbox created with sf::st_bbox() or tmaptools::bb()
- matrix created with sp::bbox() or osmdata::bbox()
- character providing textual definitions of bounding polygons, boxes or circles as allowed by the ohsome API (see ohsome API - Boundaries):
  - bboxes: WGS84 coordinates in the following format: "id1:lon1,lat1,lon2,lat2|id2:lon1,lat1,lon2,lat2|...", OR "lon1,lat1,lon2,lat2|lon1,lat1,lon2,lat2|...
  - bcircles: WGS84 coordinates + radius in meter in the following format: "id1:lon,lat,r|id2:lon,lat,r|...", OR "lon,lat,r|lon,lat,r|...
  - bpolys: WGS84 coordinates given as a list of coordinate pairs (as for bboxes) or GeoJSON FeatureCollection. The first point has to be the same as the last point and MultiPolygons are only supported in GeoJSON.
- list of bbox, matrix or character. Bounding geometry types of all list elements must be the same. Does not work with GeoJSON FeatureCollections.

... Additional arguments other than digits are ignored.

Details

set_boundary() adds a spatial filter to an ohsome_query object or replaces an existing one. The spatial filter of a query to the ohsome API can be defined as one or more polygons, bounding boxes or bounding circles.

Value

An ohsome_query object. The object can be sent to the ohsome API with ohsome_post(). It consists of the following elements:

- url: The URL of the endpoint.
- encode: The way the information is encoded and then posted to the ohsome API. Set as "form".
- body: The parameters of the query such as format, filter or bpolys.
See Also

ohsome API documentation

Examples

# Query without boundary definition
q <- ohsome_query(
  "elements/count/groupBy/boundary",
  filter = "building=*",
  time = "2022-01-01"
)

# Use franconia from the mapview package as bounding polygons
set_boundary(q, mapview::franconia, digits = 4)

# Use the bounding box of franconia
set_boundary(q, sf::st_bbox(mapview::franconia))

## Not run:
# Get bounding box of the city of Kigali from OSM
set_boundary(q, osmdata::getbb("Kigali"))

## End(Not run)

# Definition of two named bounding circles
set_boundary(q, c("Circle 1:8.6528,49.3683,1000", "Circle 2:8.7294,49.4376,1000"))
Arguments

query
An ohsome_query object constructed with ohsome_query() or any of its wrapper functions.

endpoint
The path to the ohsome API endpoint. Either a single string (e.g. "elements/count") or a vector of character in the right order (e.g. c("elements", "count")).

append
logical; If TRUE, the provided endpoint string is appended to the existing endpoint definition instead of replacing it. This is particularly useful if you wish to add density/ratio and/or a grouping to an existing aggregation query.

reset_format
logical; if TRUE, the format parameter of the query is updated depending on the new endpoint.

grouping
character; group type(s) for grouped aggregations (only available for queries to aggregation endpoints). The following group types are available:

- "boundary" groups the result by the given boundaries that are defined through any of the boundary query parameters.
- "key" groups the result by the given keys that are defined through the groupByKey query parameter.
- "tag" groups the result by the given tags that are defined through the groupByKey and groupByKeyValues query parameters.
- "type" groups the result by OSM element type.
- c("boundary", "tag") groups the result by the given boundaries and tags.

Not all of these group types are accepted by all of the aggregation endpoints. Check Grouping for available group types.

... Additional arguments passed to set_endpoint()

Details

set_endpoint() takes an ohsome_query object and modifies the ohsome API endpoint. set_grouping() takes an ohsome_query object and modifies the endpoint path for grouped aggregations.

Value

An ohsome_query object. The object can be sent to the ohsome API with ohsome_post(). It consists of the following elements:

- url: The URL of the endpoint.
- encode: The way the information is encoded and then posted to the ohsome API. Set as "form".
- body: The parameters of the query such as format, filter or bpolys.

See Also

ohsome API Endpoints
Examples

# Query for count of elements
q <- ohsome_elements_count(
    boundary = "HD:8.5992,49.3567,8.7499,49.4371|HN:9.1638,49.113,9.2672,49.1766",
    time = "2022-01-01",
    filter = "highway=*"
)

# Modify query to aggregate length of elements instead of count
set_endpoint(q, "elements/length")

# Modify query to extract geometries instead of aggregating elements
set_endpoint(q, "elements/geometry")

# Append the endpoint path in order to group aggregation by boundary
set_endpoint(q, "groupBy/boundary", append = TRUE)

# Modify query to group aggregation by boundary
set_grouping(q, grouping = "boundary")

# Modify query to group by boundary, but keep format csv instead of geojson
set_grouping(q, grouping = "boundary", reset_format = FALSE)

# Append the endpoint path to query for element densities per boundary
set_endpoint(q, c("density", "groupBy", "boundary"), append = TRUE)

# Modify query to group aggregation by OSM element type
set_grouping(q, grouping = "type")

set_parameters

Description

Sets or modifies parameters of an existing ohsome_query object

Usage

set_parameters(query, ...)
set_time(query, time = query$body$time)
set_filter(query, filter = query$body$filter, filter2 = query$body$filter2)
set_groupByKeys(query, groupByKeys = query$body$groupByKeys)
set_groupByKey(query, groupByKey = query$body$groupByKey)
set_groupByValues(query, groupByValues = query$body$groupByValues)

set_properties(query, properties = NULL)

Arguments

query An ohsome_query object constructed with ohsome_query() or any of its wrapper functions

time Parameters of the request to the ohsome API endpoint.

filter character; filter parameter of the query (see Filter)

filter2 character; filter2 parameter of a ratio query

groupByKeys character; groupByKeys parameter of a groupBy/key query

groupByKey character; groupByKey parameter of a groupBy/tag query

groupByValues character; groupByValues parameter of a groupBy/tag query

properties character; properties to be extracted with extraction queries:

  • "tags", and/or
  • "metadata" (i.e. @changesetId, @lastEdit, @osmType, @version), and/or
  • "contributionTypes" (i.e. @creation, @tagChange, @deletion, and @geometryChange; only for contributions extraction)

Multiple values can be provided as comma-separated character or as character vector. This defaults to NULL (removes properties parameter from the query body).

Details

set_parameters() takes an ohsome_query object and an arbitrary number of named parameters as an input. It sets or modifies these parameters in the ohsome_query and returns the modified object. set_time(), set_filter(), set_groupByKeys(), set_groupByKey(), set_groupByValues() and set_properties() are wrapper functions to set specific parameters. By default, an unmodified ohsome_query object is returned. In order to remove a parameter from the query object, you can set the respective argument explicitly to NULL (e.g. set_filter(query, filter = NULL)).

Value

An ohsome_query object. The object can be sent to the ohsome API with ohsome_post(). It consists of the following elements:

  • url: The URL of the endpoint.
  • encode: The way the information is encoded and then posted to the ohsome API. Set as "form".
  • body: The parameters of the query such as format, filter or bpolys.

See Also

https://docs.ohsome.org/ohsome-api/v1/
Examples

# Query ratio grouped by boundary
q1 <- ohsome_query(
  endpoint = "elements/count/ratio/groupBy/boundary",
  boundary = "HD:8.5992,49.3567,8.7499,49.4371|HN:9.1638,49.113,9.2672,49.1766"
)

# Add time, filter and format parameters
q1 |> set_time("2021/2022/P3M") |> set_filter("building=*", filter2 = "building=* and building:levels=3") |> set_parameters(format = "csv")

# Query elements area grouped by tag
q2 <- ohsome_query(
  endpoint = "elements/area/groupBy/tag",
  boundary = "HD:8.5992,49.3567,8.7499,49.4371"
)

# Add time, filter and groupByKey parameters
q2 |> set_time("2021/2022/P3M") |> set_filter("building=*") |> set_groupByKey("building:levels")
Index

* Aggregate elements
  ohsome_aggregate_elements, 2
* Extract and parse the content from an ohsome API response
  ohsome_parse, 18
* Extract contributions
  ohsome_extract_contributions, 10
* Extract elements full History
  ohsome_extract_elementsFullHistory, 14
* Extract elements
  ohsome_extract_elements, 12
* Set endpoint
  set_endpoint, 26
* Set parameters
  set_parameters, 28

ohsome_aggregate_elements, 2
ohsome_api_url, 5
ohsome_boundary, 6
ohsome_boundary(), 4, 9, 11, 13, 15, 24
ohsome_contributions_bbox
  (ohsome_extract_contributions), 10
ohsome_contributions_centroid
  (ohsome_extract_contributions), 10
ohsome_contributions_count, 8
ohsome_contributions_geometry
  (ohsome_extract_contributions), 10
ohsome_df (ohsome_parse), 18
ohsome_elements_area
  (ohsome_aggregate_elements), 2
ohsome_elements_bbox
  (ohsome_extract_elements), 12
ohsome_elements_centroid
  (ohsome_extract_elements), 12
ohsome_elements_count
  (ohsome_aggregate_elements), 2
ohsome_elements_length
  (ohsome_aggregate_elements), 2
ohsome_elements_perimeter
  (ohsome_aggregate_elements), 2
ohsome_elementsFullHistory_bbox
  (ohsome_extract_elementsFullHistory), 14
ohsome_elementsFullHistory_centroid
  (ohsome_extract_elementsFullHistory), 14
ohsome_elementsFullHistory_count
  (ohsome_extract_elementsFullHistory), 14
ohsome_elementsFullHistory_length
  (ohsome_extract_elementsFullHistory), 14
ohsome_elementsFullHistory_perimeter
  (ohsome_extract_elementsFullHistory), 14
ohsome_endpoints, 10
ohsome_extract_contributions, 10
ohsome_extract_elements, 12
ohsome_extract_elementsFullHistory, 14
ohsome_get_metadata, 16
ohsome_metadata, 17
ohsome_parse, 18
ohsome_parse(), 19, 20
ohsome_post, 19
ohsome_post(), 4, 9, 12, 14, 16, 22, 24, 25, 27, 29
ohsome_query, 20
ohsome_query(), 19, 25, 27, 29
ohsome_sf (ohsome_parse), 18
ohsome_temporalExtent, 22
ohsome_users_count, 23
osmdata::getbb(), 3, 6, 8, 11, 13, 15, 21, 23, 25

set_boundary, 24
set_boundary(), 4, 9, 11, 13, 15, 24, 25
set_endpoint, 26
set_filter (set_parameters), 28
set_groupByKey (set_parameters), 28
set_groupByKeys (set_parameters), 28
set_groupByValues(set_parameters), 28
set_grouping(set_endpoint), 26
set_parameters, 28
set_properties(set_parameters), 28
set_time(set_parameters), 28
sf::st_bbox(), 3, 6, 8, 11, 13, 15, 21, 23, 25
sp::bbox(), 3, 6, 8, 11, 13, 15, 21, 23, 25

tmaptools::bb(), 3, 6, 8, 11, 13, 15, 21, 23, 25