

Package ‘robis’

July 24, 2026

Title Ocean Biodiversity Information System (OBIS) Client

Description Client for the Ocean Biodiversity Information System (<<https://obis.org>>).

Version 2.12.0

URL <https://github.com/iobis/robis>

BugReports <https://github.com/iobis/robis/issues>

Depends R (>= 3.1.3)

Imports httr, dplyr, jsonlite, leaflet, ggplot2, tidyr, tibble,
mapedit, sf, rlang, purrr, stringr, curl, data.table,
tidyselect, xml2, glue, memoise, stats, methods

License MIT + file LICENSE

Suggests testthat, knitr, rmarkdown, spelling, httpcache

VignetteBuilder knitr

Language en-US

Encoding UTF-8

RoxygenNote 8.0.0

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Pieter Provoost [cre, aut] (ORCID:
<<https://orcid.org/0000-0002-4236-0384>>),
Samuel Bosch [aut] (ORCID: <<https://orcid.org/0000-0002-2514-0283>>),
Ward Appeltans [ctb] (ORCID: <<https://orcid.org/0000-0002-3237-4547>>),
Stephen Formel [ctb] (ORCID: <<https://orcid.org/0000-0001-7418-1244>>),
OBIS [cph]

Maintainer Pieter Provoost <pieterprovoost@gmail.com>

Repository CRAN

Date/Publication 2026-07-24 18:30:02 UTC

Contents

area	2
checklist	3
dataset	4
dna_records	5
generate_citation	6
get_geometry	6
map_ggplot	7
map_leaflet	7
measurements	8
node	8
occurrence	9
taxon	11
unnest_extension	11
Index	12

area	<i>Fetch a list of areas</i>
------	------------------------------

Description

Fetch a list of areas

Usage

area(verbose = FALSE)

Arguments

verbose logical. Optional parameter to enable verbose logging (default = FALSE).

Value

The areas.

Examples

areas <- area(verbose = FALSE)

checklist*Create a checklist*

Description

This function returns all taxa observed for a given set of filters. Users can, for example, return all taxa from a particular Phylum or from a certain geographic region.

Usage

```
checklist(scientificname = NULL, taxonid = NULL, datasetid = NULL,  
  nodeid = NULL, instituteid = NULL, areaid = NULL, startdate = NULL, enddate = NULL,  
  startdepth = NULL, enddepth = NULL, geometry = NULL, redlist = NULL,  
  hab = NULL, wrims = NULL, dropped = NULL, flags = NULL, exclude = NULL,  
  verbose = FALSE)
```

Arguments

scientificname	the scientific name.
taxonid	the taxon identifier (WoRMS AphiaID).
datasetid	the dataset identifier.
nodeid	the OBIS node identifier.
instituteid	the OBIS institute identifier.
areaaid	the OBIS area identifier.
startdate	the earliest date on which occurrence took place.
enddate	the latest date on which the occurrence took place.
startdepth	the minimum depth below the sea surface.
enddepth	the maximum depth below the sea surface.
geometry	a WKT geometry string.
redlist	include only IUCN Red List species.
hab	include only IOC-UNESCO HAB species.
wrims	include only WRiMS species.
dropped	only include dropped records (TRUE), exclude dropped records (NULL) or include dropped records (include).
flags	quality flags which need to be set.
exclude	quality flags to be excluded from the results.
verbose	logical. Optional parameter to enable verbose logging (default = FALSE).

Value

A checklist (tibble), with the records that met the filter.

Note

When using a geometry, the function will fail if the geometry is too complex. One alternative is to use a bounding box or a simplified version of the geometry.

Examples

```
taxa <- checklist(scientificname = "Tellinidae")
taxa <- checklist(geometry = "POLYGON ((2.3 51.8, 2.3 51.6, 2.6 51.6, 2.6 51.8, 2.3 51.8))")
taxa <- checklist(areaid = 10181)
```

dataset

*Create a list of datasets.***Description**

Create a list of datasets.

Usage

```
dataset(scientificname = NULL, taxonid = NULL, datasetid = NULL,
        nodeid = NULL, instituteid = NULL, areaid = NULL, startdate = NULL,
        enddate = NULL, startdepth = NULL, enddepth = NULL, geometry = NULL,
        redlist = NULL, hab = NULL, wrims = NULL, hasextensions = NULL,
        exclude = NULL, keyword = NULL, verbose = FALSE)
```

Arguments

scientificname	the scientific name.
taxonid	the taxon identifier (WoRMS AphiaID).
datasetid	the dataset identifier.
nodeid	the OBIS node identifier.
instituteid	the OBIS institute identifier.
areaid	the OBIS area identifier.
startdate	the earliest date on which occurrence took place.
enddate	the latest date on which the occurrence took place.
startdepth	the minimum depth below the sea surface.
enddepth	the maximum depth below the sea surface.
geometry	a WKT geometry string.
redlist	include only IUCN Red List species.
hab	include only IOC-UNESCO HAB species.
wrims	include only WRiMS species.
hasextensions	which extensions need to be present (e.g. MeasurementOrFact, DNADerived-Data).

exclude	quality flags to be excluded from the results.
keyword	Keyword(s) to search for in dataset metadata. When keyword is used, no other filter parameters may be specified. The search uses Elasticsearch simple_query_string syntax. The following symbolic operators are supported: • +: Equivalent to AND, requiring all terms. Any whitespace is also an implicit AND. For example, coral+reef is equivalent to coral reef. • Quotes ("..."): Used for exact phrase matching. For example, "coral reef" is stricter than the unquoted version. • : Equivalent to OR. Returns records where any listed term is present. • -: Equivalent to NOT. Excludes records containing the specified term. • *: Wildcard matching is supported only at the end of a word. For example, star* matches "starfish" and sea* matches "seastar". However, *star* returns no results, and leading wildcards like *star may appear to work in some cases but should not be relied upon. • Grouping with parentheses: (coral kelp) -fish returns results that include coral OR kelp, but NOT fish. Removing the parentheses would change the meaning — it would return results that include coral, or kelp only when fish is also present (i.e., coral OR (kelp -fish)).
verbose	logical. Optional parameter to enable verbose logging (default = FALSE).

Value

A tibble of matching datasets.

Examples

```
## Not run:
dataset(scientificname = "Tellinidae")
dataset(areaid = 10181)
dataset(keyword = '(coral | kelp) -fish')

## End(Not run)
```

`dna_records`

Extract DNA records from occurrence data with a dna column.

Description

Extract DNA records from occurrence data with a dna column.

Usage

```
dna_records(df, fields = "id")
```

Arguments

- df the occurrence dataframe.
- fields columns from the occurrence dataframe to include.

Value

The DNA records.

generate_citation	<i>Generate a citation from metadata elements.</i>
-------------------	--

Description

Generate a citation from metadata elements.

Usage

```
generate_citation(title, published, url, contacts)
```

Arguments

- title the dataset title.
- published the dataset published date.
- url the dataset url.
- contacts the dataset contacts as a dataframe.

Value

A citation string.

get_geometry	<i>Get a WKT geometry by drawing on a map.</i>
--------------	--

Description

Get a WKT geometry by drawing on a map.

Usage

```
get_geometry(provider_tiles = "Esri.WorldGrayCanvas")
```

Arguments

- provider_tiles the base map provider.

Value

A WKT representation of a geometry.

map_ggplot	Create a ggplot2 map.
------------	-----------------------

Description

Create a ggplot2 map.

Usage

```
map_ggplot(data, color = "#ff3399")
```

Arguments

data	the occurrences from occurrence().
color	color to be used for the dots.

Value

A ggplot object.

map_leaflet	Create a leaflet map.
-------------	-----------------------

Description

Create a leaflet map.

Usage

```
map_leaflet(data, color = "#ff3399",  
  provider_tiles = "Esri.WorldGrayCanvas", popup = function(x) { x["id"] },  
  antarctic = FALSE)
```

Arguments

data	the occurrences from occurrence().
color	color to be used for the dots.
provider_tiles	the base map provider.
popup	function generating the popup content.
antarctic	use antarctic polar stereographic projection.

Value

A leaflet object.

measurements	<i>Extract measurements or facts from occurrence data with a mof column.</i>
--------------	--

Description

Extract measurements or facts from occurrence data with a mof column.

Usage

```
measurements(df, fields = "id")
```

Arguments

df	the occurrence dataframe.
fields	columns from the occurrence dataframe to include.

Value

The measurements.

node	<i>Fetch a list of nodes</i>
------	------------------------------

Description

Fetch a list of nodes

Usage

```
node(verbose = FALSE)
```

Arguments

verbose	logical. Optional parameter to enable verbose logging (default = FALSE).
---------	--

Value

The nodes

Examples

```
nodes <- node()
```

occurrence	<i>Find occurrence records</i>
------------	--------------------------------

Description

Download occurrence records from OBIS. A set of filters can be used to delimit which data will be downloaded. Note that scientificname can be a species name or any other taxon name (e.g. a family name).

Usage

```
occurrence(scientificname = NULL, taxonid = NULL, datasetid = NULL,
  nodeid = NULL, instituteid = NULL, areaid = NULL, startdate = NULL, enddate = NULL,
  startdepth = NULL, enddepth = NULL, geometry = NULL,
  measurementtype = NULL, measurementtypeid = NULL, measurementvalue = NULL,
  measurementvalueid = NULL, measurementunit = NULL, measurementunitid = NULL,
  redlist = NULL, hab = NULL, wrims = NULL, extensions = NULL, hasextensions = NULL,
  mof = NULL, dna = NULL, absence = NULL, event = NULL, dropped = NULL,
  flags = NULL, exclude = NULL, fields = NULL, qcfields = NULL, q=NULL, verbose = FALSE)
```

Arguments

scientificname	the scientific name.
taxonid	the taxon identifier (WoRMS AphiaID).
datasetid	the dataset identifier.
nodeid	the OBIS node identifier.
instituteid	the OBIS institute identifier.
areaid	the OBIS area identifier.
startdate	the earliest date on which occurrence took place.
enddate	the latest date on which the occurrence took place.
startdepth	the minimum depth below the sea surface.
enddepth	the maximum depth below the sea surface.
geometry	a WKT geometry string.
measurementtype	the measurement type to be included in the measurements data.
measurementtypeid	the measurement type ID to be included in the measurements data.
measurementvalue	the measurement value to be included in the measurements data.
measurementvalueid	the measurement value ID to be included in the measurements data.
measurementunit	the measurement unit to be included in the measurements data.

measurementunitid	the measurement unit ID to be included in the measurements data.
redlist	include only IUCN Red List species.
hab	include only IOC-UNESCO HAB species.
wrims	include only WRiMS species.
extensions	which extensions to include (e.g. MeasurementOrFact, DNADerivedData, default = NULL).
hasextensions	which extensions need to be present (e.g. MeasurementOrFact, DNADerivedData, default = NULL).
mof	include measurements data (default = NULL).
dna	include DNA data (default = NULL).
absence	only include absence records (TRUE), exclude absence records (NULL) or include absence records (include).
event	only include pure event records (TRUE), exclude pure event records (NULL) or include event records (include).
dropped	only include dropped records (TRUE), exclude dropped records (NULL) or include dropped records (include).
flags	quality flags which need to be set.
exclude	quality flags to be excluded from the results.
fields	fields to be included in the results.
qcfields	include lists of missing and invalid fields (default = NULL).
q	text search query.
verbose	logical. Optional parameter to enable verbose logging (default = FALSE).

Value

The occurrence records in a tibble format. The number of columns will vary according to the number of information associated with the records. For more information about the type of data available with OBIS occurrences, see the [OBIS manual](#).

Note

When using a geometry, the function will fail if the geometry is too complex. One alternative is to use a bounding box or a simplified version of the geometry.

Examples

```
records <- occurrence(scientificname = "Abra sibogai")
records <- occurrence(taxonid = 141438, startdate = as.Date("2007-10-10"))
```

taxon	<i>Get taxon by taxon ID.</i>
-------	-------------------------------

Description

Get taxon by taxon ID.

Usage

```
taxon(taxonid, verbose = FALSE)
```

Arguments

taxonid	the taxon identifier (WoRMS AphiaID).
verbose	logical. Optional parameter to enable verbose logging (default = FALSE).

Value

Tibble containing taxon records.

Examples

```
taxon(c(141433, 141434))
```

unnest_extension	<i>Extract extension records from occurrence data with nested extension column.</i>
------------------	---

Description

Extract extension records from occurrence data with nested extension column.

Usage

```
unnest_extension(df, extension, fields = "id")
```

Arguments

df	the occurrence dataframe.
extension	the extension type (e.g. MeasurementOrFact, DNADerivedData).
fields	columns from the occurrence dataframe to include.

Value

The extension records.

Index

area, [2](#)

checklist, [3](#)

dataset, [4](#)

dna_records, [5](#)

generate_citation, [6](#)

get_geometry, [6](#)

map_ggplot, [7](#)

map_leaflet, [7](#)

measurements, [8](#)

node, [8](#)

occurrence, [9](#)

taxon, [11](#)

unnest_extension, [11](#)