Package 'finbif'

October 22, 2026

Title Interface for the 'Finnish Biodiversity Information Facility' API

Version 0.10.1

Description A programmatic interface to the 'Finnish Biodiversity Information Facility' ('FinBIF') API (<https://api.laji.fi>). 'FinBIF' aggregates Finnish biodiversity data from multiple sources in a single open access portal for researchers, citizen scientists, industry and government. 'FinBIF' allows users of biodiversity information to find, access, combine and visualise data on Finnish plants, animals and microorganisms. The 'finbif' package makes the publicly available data in 'FinBIF' easily accessible to programmers. Biodiversity information is available on taxonomy and taxon occurrence. Occurrence data can be filtered by taxon, time, location and other variables. The data accessed are conveniently preformatted for subsequent analyses.

License MIT + file LICENSE

URL https://github.com/luomus/finbif, https://luomus.github.io/finbif/

BugReports https://github.com/luomus/finbif/issues

VignetteBuilder knitr

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports secretbase, httr2 (>= 1.1.2), jsonlite, lutz, utils

Suggests callr, data.table, DBI, future, knitr, rmarkdown, RSQLite, testthat (>= 3.0.0), vcr (>= 0.6.0), webfakes

Config/testthat/edition 3

X-schema.org-applicationCategory Biodiversity

X-schema.org-keywords api, biodiversity, biodiversity-informatics, biodiversity-information, finbif, finbif-access, occurrences, r-package, r-programming, rstats, species, specimens, taxon, taxonomy, web-services

X-schema.org-isPartOf https://species.fi

Config/roxygen2/version 8.1.1

Config/roxygen2/markdown TRUE

NeedsCompilation no

Author Finnish Museum of Natural History - Luomus [cph] (ROR: <https://ror.org/03tcx6c30>), Finnish Biodiversity Information Facility [cph] (ROR: <https://ror.org/01c0yey93>), William K. Morris [aut, cre] (ORCID: <https://orcid.org/0000-0002-8686-4154>)

Maintainer William K. Morris <willi@mmorris.email>

Repository CRAN

Date/Publication 2026-10-22 13:30:02 UTC

Contents

finbif-package . . . 2
caching . . . 4
filters . . . 5
finbif_check_taxa . . . 10
finbif_clear_cache . . . 11
finbif_collections . . . 11
finbif_informal_groups . . . 12
finbif_last_mod . . . 13
finbif_metadata . . . 14
finbif_occurrence . . . 15
finbif_occurrence_load . . . 17
finbif_request_token . . . 20
finbif_taxa . . . 20
finbif_update_cache . . . 22
to_dwc . . . 22
variables . . . 23

Index **28**

`finbif-package` *finbif: Interface for the 'Finnish Biodiversity Information Facility' API*

Description

A programmatic interface to the 'Finnish Biodiversity Information Facility' ('FinBIF') API (https://api.laji.fi). 'FinBIF' aggregates Finnish biodiversity data from multiple sources in a single open access portal for researchers, citizen scientists, industry and government. 'FinBIF' allows users of biodiversity information to find, access, combine and visualise data on Finnish plants, animals and microorganisms. The 'finbif' package makes the publicly available data in 'FinBIF' easily accessible to programmers. Biodiversity information is available on taxonomy and taxon

occurrence. Occurrence data can be filtered by taxon, time, location and other variables. The data accessed are conveniently preformatted for subsequent analyses.

Package options

`finbif_api_url` Character. The base URL of the API to query. Default: `"https://api.laji.fi"`

`finbif_api_version` Character. The API version to use. Default: `"1"`

`finbif_allow_query` Logical. Should remote API queries by allowed? Default: `TRUE`

`finbif_hide_progress` Logical. Global option to suppress progress indicators for downloading, importing and processing FinBIF records. Default: `FALSE`

`finbif_use_async` Logical. Should downloading and processing of data be done asynchronously? Default: `TRUE`

`finbif_use_cache` Logical or Integer. If `TRUE` or a number greater than zero, then data-caching will be used. If not logical then cache will be invalidated after the number of hours indicated by the value. Default: `TRUE`

`finbif_use_cache_metadata` Logical or Integer. If `TRUE` or a number greater than zero, then metadata-caching will be used. If not logical then cache will be invalidated after the number of hours indicated by the value. Default: `TRUE`

`finbif_cache_path` Character. The path to the directory where to store cached API queries. If unset (the default) in memory caching is used.

`finbif_tz` Character. The timezone used by `finbif` functions that compute dates and times. Default: `Sys.timezone()`

`finbif_locale` Character. One of the supported two-letter ISO 639-1 language codes. Current supported languages are English, Finnish and Swedish. By default, the system settings are used to set this option if they are set to one of the supported languages, otherwise English is used.

`finbif_use_dwc` Logical. Should DarwinCore style field names be used? Default: `FALSE`

`finbif_use_all_collections` Logical. Should all collections be used? Default: `FALSE`

Author(s)

Maintainer: William K. Morris <willi@mmorris.email> (ORCID)

Authors:

- William K. Morris <willi@mmorris.email> (ORCID)

Other contributors:

- Finnish Museum of Natural History - Luomus (ROR) [copyright holder]
- Finnish Biodiversity Information Facility (ROR) [copyright holder]

See Also

Useful links:

- https://github.com/luomus/finbif
- https://luomus.github.io/finbif/
- Report bugs at https://github.com/luomus/finbif/issues

caching *Caching FinBIF downloads*

Description

Working with cached data from FinBIF.

Turning caching off

By default, local caching of most FinBIF API requests is turned on. Any request made using the same arguments will only request data from FinBIF in the first instance and subsequent requests will use the local cache while it exists. This will increase the speed of repeated requests and save bandwidth and computation for the FinBIF server. Caching can be turned off temporarily by setting `cache = c(FALSE, FALSE)` in the requesting function.

Setting `options(finbif_use_cache = FALSE, finbif_use_cache_metadata = FALSE)` will turn off caching for the current session.

Using filesystem caching

By default cached requests are stored in memory. This can be changed by setting the file path for the current session with `options(finbif_cache_path = "path/to/cache")`.

Using database caching

Caching can also be done using a database. Using a database for caching requires the `DBI` package and a database backend package such as `RSQLite` to be installed. To use the database for caching simply pass the connection objected created with `DBI::dbConnect` to the `finbif_cache_path` option (e.g., `db <- DBI::dbConnect(RSQLite::SQLite(), "my-db.sqlite"); options(finbif_cache_path = db)`).

Timeouts

A cache timeout can be set by using an integer (number of hours until cache is considered invalid and is cleared) instead of a logical value for the `finbif_use_cache` and `finbif_use_cache_metadata` options or the `cache` function arguments.

Clearing the cache

The cache can be reset `finbif_clear_cache()`.

Updating the cache

The cache can be updated using `finbif_update_cache()`.

filters *Filtering FinBIF records*

Description

Filters available for FinBIF records and occurrence data.

Taxa

Filters related to taxa include:

- `taxon_id` Character vector. FinBIF taxon IDs. The functions `finbif_check_taxa()` and `finbif_taxa()` can be used to search for taxon IDs.
- `taxon_name` Character vector. Filter based on taxon names (scientific or common) rather than IDs. If the specified taxa are not found in the FinBIF taxonomy then matches are attempted with the occurrence record names as originally supplied verbatim.
- `quality_controlled_det` Logical. If `TRUE` (default) use quality controlled taxonomic determinations. Or, if `FALSE` use the originally recorded taxonomic determinations.
- `subtaxa` Logical. If `TRUE` (default) return records of all taxa belonging to specified taxa. Or, if `FALSE` only return records for exact matches to the specified taxa (e.g., if a genus is specified, do not return records of the species belonging to the genus, return records of individuals identified as that genus only and not identified to a lower taxonomic level).
- `invalid_taxa` Logical. If `TRUE` (default) return records for taxa not found in the FinBIF taxonomic database as well as taxa that are in the FinBIF database. Or, if `FALSE` limit records to only those of taxa found in the FinBIF database.
- `informal_groups` Character vector. Filter by informal taxonomic groups. Only including informal groups linked to the recorded taxa in the FinBIF database. Use the function `finbif_informal_groups()` to see the informal taxonomic groups available in FinBIF.
- `informal_groups_reported` Character vector. Filter by informal taxonomic groups including groups reported directly with the record and those linked to the recorded taxa in the FinBIF database. Use the function `finbif_informal_groups()` to see the informal taxonomic groups available in FinBIF.
- `regulatory_status` Character vector. Filter by regulatory status code. Use the function `finbif_metadata()` to see regulatory statuses and codes.
- `red_list_status` Character vector. Filter by IUCN red list status code. Use the function `finbif_metadata()` to see red list statuses and codes.
- `primary_habitat` Character vector or named list of character vectors. Filter by primary habitat code. Use the function `finbif_metadata()` to see habitat (sub)types and codes for taxa in the FinBIF database. Habitat type/subtypes can be refined further by indicating habitat qualifiers with a named list of character vectors where the names are habitat (sub)type codes and the elements of the character vector are the habitat qualifier codes. Use the function `finbif_metadata()` to see habitat qualifiers and codes. The records returned will be of taxa whose primary habitat is considered to be the (sub)habitat/habitat qualifier combination supplied.

- primary_secondary_habitat Character or named list of character vectors. As above, except the records returned will be of taxa whose primary or secondary habitat is considered to be the combination supplied.
- finnish_occurrence_status Character vector. Filter by Finnish occurrence status of taxa. Use finbif_metadata() to see the possible occurrence statuses of taxa.
- finnish_occurrence_status_neg Character vector. Negation of the above. Selecting a status will filter out rather than include records with the selected status.
- finnish Logical. If TRUE, limit records to taxa thought to occur in Finland. Or if FALSE limit to taxa not thought to occur in Finland. If unspecified (default) return records of all taxa.
- invasive Logical. If TRUE, limit records to invasive taxa. Or if FALSE limit to non-invasive taxa. If unspecified (default) return records of invasive and non-invasive taxa.
- taxon_rank Character vector. Filter by taxonomic rank. Use finbif_metadata() to see the taxonomic ranks available. Records returned will be limited to the specified ranks and not include records of lower taxonomic levels.

Location

Filters related to location of record include:

- locality Character vector. Filter by name of locality. Will first try to match strings to the countries, bio-provinces, and municipalities (see below) in FinBIF, if none of these locality types match exactly then will return records with verbatim locality matches in the original records.
- country Character vector. Filter by country. Use finbif_metadata() to see country names and ISO codes (2 and 3 character) used in FinBIF.
- region Character vector. Filter by region. Use finbif_metadata() to see region names and codes.
- bio_province Character vector. Filter by bio-province. Use finbif_metadata() to see bio-province names and codes.
- finnish_municipality Character vector. Filter by municipality. Use finbif_metadata() to see Finnish municipality names.
- location_tag Character vector Filter by tags associated with a location (e.g., "farmland").
- bird_assoc_area Character vector. Filter by BirdLife Finland association area. Use finbif_metadata() to see association names and codes.
- coordinates Coordinates. A character vector or list of coordinate data. Must be length 3 to 4 (e.g., list(lat = c(60.4, 61), lon = c(22, 22.5), system = "wgs84", ratio = 1). The first element is minimum and maximum latitude and the second minimum and maximum longitude (or can be minimums only). The third element is the coordinate system; either one of "wgs84", "euref" or "ykj". The optional fourth element is a positive value less than 1. When 1, the coverage area of the returned records will be completely within the box bound by the coordinates values. Values less than 1 requires the returned record's coverage to overlap with the bounding box in that proportion. When using the system "ykj" the coordinates will be coerced to integers with units inferred from the number of integer digits (7 digits equals kms, 6 equals 10kms, etc.,). If coordinate maximums are not specified they will be assumed to be one unit above the minimums (e.g., c(666, 333, "ykj") is equivalent to list(c(6660000, 6670000), c(3330000, 3340000), "ykj")).

- coordinates_center Coordinates. A character vector or list of coordinate data. Must be of length 3. The first two elements are latitude and longitude and third is the coordinate system (currently only "wgs84" is implemented). Records returned will be those for which the center point exactly matches that which is specified.
- coordinates_cell_{1k|10k|50k|100k} Coordinates. A vector of coordinate data (lat, lon). Filter by grid cell at scale *. Where * is 1, 10, 50 or 100. The coordinates specify the southeast corner of the cell. Coordinates system is "ykj".
- coordinates_cell_{1k|10k|50k|100k}_center Coordinates. As above, except coordinates indicate center of grid cell.
- coordinates_source Character. Filter by source of coordinates. Currently accepted values are "reported_value" (coordinates were recorded at time of observation) and "finnish_municipality" (coordinates were derived and observer only recorded Finnish municipality).
- coordinates_uncertainty_max Integer. Filter by maximum uncertainty of coordinates (i.e., coordinates_uncertainty_max = 100 will return records that are accurate to 100m).

Time

Filters related to time of record include:

- date_range_ymd Dates. A vector of one to two Date objects (begin and end dates) or objects that are coercible to the Date. When supplying dates as strings, the day or month-and-day can be omitted (e.g., "2001-04" or "2001"). Note however, that when omitting day, only "-" is allowed to separate year and month, and months must be in two-digit/leading zero form. If the begin or end dates are partial date strings they will be interpreted as the first or last day of the month or year (e.g., c(2001, 2003) is equivalent to c("2001-01-01", "2003-12-12")). If a single date is supplied as a partial date string then all records that fall within that month or year will be returned (e.g., c("2001-01") is equivalent to c("2001-01-01", "2001-01-31")). Use empty strings for the begin or end date to specify open-ended date ranges (e.g., c("2000-01-01", "") for all dates from the turn of the century).
- date_range_ym Dates. As above, but days (if supplied) will be ignored.
- date_range_d Integer vector. Filter by day of the year (e.g., 1 to 366). If begin or end date is omitted then it is interpreted as the first or last day of the year.
- date_range_md Character vector. Filter by month and day of the year (e.g., "01-01" to "12-31"). If begin or end date is omitted then it is interpreted as the first or last day of the year.
- {first|last}_import_date_{min|max} Date. Filter by date record was imported/modified. Either a Date object or object that is coercible to the Date class, or the number of seconds since 1970-01-01 00:00.00 UTC (the so-called UNIX epoch). Note that this means that specifying a year, such as 2019, without a month and day will be interpreted as 2019 seconds after midnight on Jan 1, 1970 and not the year 2019.

Quality

Filters related to quality of record:

- quality_issues. Character. Filter by the presence of record quality issues. One of "without_issues", "with_issues" or "both". Issues include any quality issues with the record, the event, or the document. The default is "without_issues" unless filtering by record, event or document ID or record annotation status.

- `requires_verification` Logical. Show only records requiring verification (`TRUE`) or not requiring verification (`FALSE`).
- `collection_quality` Character vector. Filter by one or more collection quality types. Must be one of `"professional"`, `"hobbyist"` or `"amateur"`.
- `record_reliability` Character vector. Filter by the reliability of the record. Must be one or more of `"reliable"`, `"unassessed` or `"unreliable"`. Default is `c("reliable", "unassessed")`.
- `record_quality` Character vector. Filter by the quality of the record. Must be one or more of `"expert_verified"`, `"community_verified"`, `"unassessed"`, `"uncertain"`, or `"erroneous"`.

Misc

Other filters:

- `keywords` Character vector. Filter by keywords.
- `collection` Character vector or `finbif_collections()` data.frame. Filter by collection. If a character vector can refer to collection ID, collection name (in English) or abbreviated name. Use `finbif_collections()` to see list of collections and metadata. Can also use the results of a call to `finbif_collections()` directly to filter records.
- `subcollections` Logical. If `TRUE` (default) include the subcollections of the collections specified. If `FALSE` do not include subcollections.
- `not_collection` Character vector or `finbif_collections()` data.frame. As for `collection`, but result will be the negation of the specified collections.
- `source` Character vector. Filter by information system data source. Use `finbif_metadata()` to see data source IDs names and descriptions.
- `record_basis` Character vector. Filter by basis of record. Use `finbif_metadata()` to see list of record bases.
- `superrecord_basis` Character vector. Filter by superset of record basis. One or more of `"human_observation"`, `"machine_observation"`, or `"specimen"`.
- `life_stage` Character vector. Filter by organism life stage. Use `finbif_metadata()` to see list of organism life stages.
- `sex` Character vector. Filter by organism sex and sex-related category name or code. Use `finbif_metadata()` to see list of organism sexes and sex-related categories and codes. If `"male"` or `"female"` is specified then records returned will be those with sex specified as male or female respectively and those records where the corresponding `{male|female}_abundance > 1`.
- `event_id` Character. Filter by event (list of records, etc.) ID.
- `document_id` Character. Filter by the document (collection of events) ID of occurrences.
- `record_id` Character.
- `individual_id` Character. Filter by individual (an individual organism) ID.
- `abundance_min` Integer. Filter by the minimum number of individual organisms in the record.
- `abundance_max` Integer. Filter by the maximum number of individual organisms in the record.
- `type_specimen` Logical. Filter by whether or not the record is a type specimen.

- `wild_status` Character. Filter by "wildness" status of records. One or more of `"wild"`, `"non_wild"` or `"unknown"`. Default is `c("wild", "unknown")`.
- `is_breeding_location` Logical. Filter by whether or not the occurrence is recorded at a known breeding location.
- `has_document_media` Logical. Filter by whether there is media (images, video, audio, etc.) associated with the records' document.
- `has_event_media` Logical. Filter by whether there is media (images, video, audio, etc.) associated with the records' event.
- `has_record_media` Logical. Filter by whether there is media (images, video, audio, etc.) associated with the record.
- `has_media` Logical. Filter by whether there is any media (images, video, audio, etc.) associated with the record, its document or its event.
- `event_observer_name` Character. Filter by observer name.
- `event_observer_id` Integer. Filter by observer ID.
- `restriction_reason` Character vector. Filter by reason data has security restrictions. See `finbif_metadata()` for a list of reasons data may have security restrictions.
- `restriction_level` Character vector. Filter by data restriction level. See `finbif_metadata()` for a list of the levels of data restrictions.
- `restricted` Logical. Filter records by whether any data restrictions are in place (`TRUE`) or not (`FALSE`).
- `annotated` Logical. Filter records that do (`TRUE`) or do not (`FALSE`) have annotations.
- `unidentified` Logical. Filter by whether the record has been identified to species level and linked to the FinBIF taxon database (`FALSE`) or has not been identified to species level reliably and linked to the taxon database (`TRUE`).
- `taxon_census` Character vector. Return records belonging to surveys or censuses of a given taxon or taxonomic group. Specify the taxonomic group with a FinBIF taxon ID. Use `finbif_check_taxa()` to find taxon IDs.
- `{record|event|document}_fact` Character vector. Filter by record, event or document facts. Facts are key-value pairs of the form `"<fact>=<value>"`. Value can be omitted in which case all records with any value recorded for the specified fact will be returned.
- `has_sample` Logical. Record includes a sample or samples (e.g., a DNA sample or preparation).
- `complete_list_type` Filter by complete list type. Records made during monitoring that produces taxon lists for a given group of taxa (e.g., birds) can include all species observed with breeding status recorded for each observed species (`all_species_and_breeding`), all species observed without breeding status recorded for all species observed, all species observed with or without breeding status recorded for all observed species (`all_species`), or only some of the species observed (`incomplete`).
- `complete_list_taxon_id` Filter by the taxon ID of the target group (e.g., birds) for a complete list.

`finbif_check_taxa` *Check FinBIF taxa*

Description

Check that taxa are in the FinBIF database.

Usage

```
finbif_check_taxa(taxa, cache = getOption("finbif_use_cache"))
```

Arguments

`taxa`	Character (or list of named character) vector(s). If a list each vector can have the name of a taxonomic rank (genus, species, etc.,). The elements of the vectors should be the taxa to check.
`cache`	Logical or Integer. If `TRUE` or a number greater than zero, then data-caching will be used. If not logical then cache will be invalidated after the number of hours indicated by the argument.

Value

An object of class `finbif_taxa`. A list with the same form as `taxa`.

Examples

```
## Not run:

# Check a scientific name
finbif_check_taxa("Cygnus cygnus")

# Check a common name
finbif_check_taxa("Whooper swan")

# Check a genus
finbif_check_taxa("Cygnus")

# Check a list of taxa
finbif_check_taxa(
  list(
    species = c("Cygnus cygnus", "Ursus arctos"),
    genus   = "Betula"
  )
)

## End(Not run)
```

finbif_clear_cache *Clear cache*

Description

Remove cached FinBIF API requests.

Usage

```
finbif_clear_cache()
```

Value

NULL invisibly

Examples

```
## Not run:

finbif_clear_cache()


## End(Not run)
```

finbif_collections *FinBIF collections*

Description

Get information on collections in the FinBIF database.

Usage

```
finbif_collections(
  filter,
  select,
  subcollections = TRUE,
  supercollections = FALSE,
  locale = getOption("finbif_locale"),
  nmin = 0,
  cache = getOption("finbif_use_cache_metadata")
)
```

Arguments

filter	Logical. Expression indicating elements or rows to keep: missing values are taken as false.
select	Expression. Indicates columns to select from the data frame.
subcollections	Logical. Return subcollection metadata of higher level collections.
supercollections	Logical. Return lowest level collection metadata.
locale	Character. Language of data returned. One of "en", "fi", or "sv".
nmin	Integer. Filter collections by number of records. Only return information on collections with greater than value specified. If NA then return information on all collections.
cache	Logical or Integer. If TRUE or a number greater than zero, then data-caching will be used. If not logical then cache will be invalidated after the number of hours indicated by the argument.

Value

A data.frame.

Examples

```
## Not run:

# Get collection metadata
collections <- finbif_collections()


## End(Not run)
```

---

finbif_informal_groups

FinBIF informal groups

Description

Display the informal taxonomic groups used in the FinBIF database.

Usage

```
finbif_informal_groups(
  group,
  limit = 5,
  quiet = FALSE,
  locale = getOption("finbif_locale"),
  cache = getOption("finbif_use_cache_metadata")
)
```

Arguments

group	Character. Optional, if supplied only display this top-level group and its sub-groups.
limit	Integer. The maximum number top-level informal groups (and their sub-groups) to display.
quiet	Logical. Return informal group names without displaying them.
locale	Character. One of the supported two-letter ISO 639-1 language codes. Current supported languages are English, Finnish and Swedish. For data where more than one language is available the language denoted by `locale` will be preferred while falling back to the other languages in the order indicated above.
cache	Logical or Integer. If `TRUE` or a number greater than zero, then data-caching will be used. If not logical then cache will be invalidated after the number of hours indicated by the argument.

Value

A character vector (invisibly).

Examples

```
## Not run:

# Display the informal taxonomic groups used by FinBIF
finbif_informal_groups()


## End(Not run)
```

---

finbif_last_mod *Get last modified date for FinBIF occurrence records*

Description

Get last modified date for filtered occurrence data from FinBIF.

Usage

```
finbif_last_mod(..., filter)
```

Arguments

...	Character vectors or list of character vectors. Taxa of records to download.
filter	List of named character vectors. Filters to apply to records.

Value

A `Date` object

Examples

```
## Not run:

# Get last modified date for Whooper Swan occurrence records from Finland
finbif_last_mod("Cygnus cygnus", filter = c(country = "Finland"))


## End(Not run)
```

`finbif_metadata` *FinBIF metadata*

Description

Display metadata from the FinBIF database.

Usage

```
finbif_metadata(
  which,
  locale = getOption("finbif_locale"),
  cache = getOption("finbif_use_cache_metadata")
)
```

Arguments

`which`	Character. Which category of metadata to display. If unspecified, function returns the categories of metadata available.
`locale`	Character. One of the supported two-letter ISO 639-1 language codes. Current supported languages are English, Finnish and Swedish. For data where more than one language is available the language denoted by `locale` will be preferred while falling back to the other languages in the order indicated above.
`cache`	Logical or Integer. If `TRUE` or a number greater than zero, then data-caching will be used. If not logical then cache will be invalidated after the number of hours indicated by the argument.

Value

A data.frame.

Examples

```
## Not run:

finbif_metadata("red_list")


## End(Not run)
```

finbif_occurrence *Download FinBIF occurrence records*

Description

Download filtered occurrence data from FinBIF as a `data.frame`.

Usage

```
finbif_occurrence(
  ...,
  filter = NULL,
  select = NULL,
  order_by = NULL,
  aggregate = "none",
  sample = FALSE,
  n = 10,
  page = 1,
  count_only = FALSE,
  quiet = getOption("finbif_hide_progress"),
  cache = getOption("finbif_use_cache"),
  dwc = getOption("finbif_use_dwc"),
  date_time_method = NULL,
  check_taxa = TRUE,
  on_check_fail = c("warn", "error"),
  tzone = getOption("finbif_tz"),
  locale = getOption("finbif_locale"),
  seed = NULL,
  drop_na = FALSE,
  aggregate_counts = TRUE,
  exclude_na = FALSE,
  unlist = FALSE,
  facts = NULL,
  duplicates = FALSE,
  filter_col = NULL,
  restricted_api = NULL
)
```

Arguments

`...`	Character vectors or list of character vectors. Taxa of records to download.
`filter`	List of named character vectors. Filters to apply to records.
`select`	Character vector. Variables to return. If not specified, a default set of commonly used variables will be used. Use `"default_vars"` as a shortcut for this set. Variables can be deselected by prepending a `-` to the variable name. If only deselects are specified the default set of variables without the deselection will be returned.

`order_by` Character vector. Variables to order records by before they are returned. Most, though not all, variables can be used to order records before they are returned. Ordering is ascending by default. To return in descending order append a `-` to the front of the variable (e.g., `"-date_start"`). Default order is `"-date_start"` > `"-load_date"` > `"reported_name"` > `"record_id"`.

`aggregate` Character. If `"none"` (default), returns full records. If one or more of `"records"`, `"species"`, `"taxa"`, `"individuals"`, `"pairs"`, `"events"` or `"documents"`; aggregates combinations of the selected variables by counting records, species, taxa, individuals or events or documents. Aggregation by events or documents cannot be done in combination with any of the other aggregation types.

`sample` Logical. If `TRUE` randomly sample the records from the FinBIF database.

`n` Integer. How many records to download/import.

`page` Integer. Which page of records to start downloading from.

`count_only` Logical. Only return the number of records available.

`quiet` Logical. Suppress the progress indicator for multipage downloads. Defaults to value of option `finbif_hide_progress`.

`cache` Logical or Integer. If `TRUE` or a number greater than zero, then data-caching will be used. If not logical then the cache will be invalidated after the number of hours indicated by the argument. If a length one vector is used, its value will only apply to caching occurrence records. If the value is length two, then the second element will determine how metadata is cached.

`dwc` Logical. Use Darwin Core (or Darwin Core style) variable names.

`date_time_method` Character. Passed to `lutz::tz_lookup_coords()` when `date_time` and/or `duration` variables have been selected. Default is `"fast"` when less than 100,000 records are requested and `"none"` when more. Using method `"none"` assumes all records are in timezone "Europe/Helsinki", Use `date_time_method = "accurate"` (requires package `sf`) for greater accuracy at the cost of slower computation.

`check_taxa` Logical. Check first that taxa are in the FinBIF database. If true only records that match known taxa (have a valid taxon ID) are returned.

`on_check_fail` Character. What to do if a taxon is found not valid. One of `"warn"` (default) or `"error"`.

`tzone` Character. If `date_time` has been selected the timezone of the outputted date-time. Defaults to system timezone.

`locale` Character. One of the supported two-letter ISO 639-1 language codes. Current supported languages are English, Finnish and Swedish. For data where more than one language is available the language denoted by `locale` will be preferred while falling back to the other languages in the order indicated above.

`seed` Integer. Set a seed for randomly sampling records.

`drop_na` Logical. A vector indicating which columns to check for missing data. Values recycled to the number of columns. Defaults to all columns.

`aggregate_counts` Logical. Should count variables be returned when using aggregation.

exclude_na	Logical. Should records where all selected variables have non-NA values only be returned.
unlist	Logical. Should variables that contain non atomic data be concatenated into a string separated by ";"?
facts	Character vector. Extra variables to be extracted from record, event and document "facts".
duplicates	Logical. If TRUE, allow duplicate records/aggregated records when making multi-filter set requests. If FALSE (default) duplicate records are removed.
filter_col	Character. The name of a column, with values derived from the names of the filter sets used when using multiple filters, to include when using multiple filter sets. If NULL (default), no column is included.
restricted_api	Character. If using a restricted data API token in addition to a personal access token, a string indicating the name of an environment variable storing the restricted data API token.

Value

A data.frame. If count_only = TRUE an integer.

Examples

```
## Not run:

# Get recent occurrence data for taxon
finbif_occurrence("Cygnus cygnus")

# Specify the number of records
finbif_occurrence("Cygnus cygnus", n = 100)

# Get multiple taxa
finbif_occurrence("Cygnus cygnus", "Ursus arctos")

# Filter the records
finbif_occurrence(
  species = "Cygnus cygnus",
  filter = list(coordinate_uncertainty_max = 100)
)


## End(Not run)
```

---

finbif_occurrence_load

Load FinBIF occurrence records from a file

Description

Load occurrence data from a file as a data.frame.

Usage

```
finbif_occurrence_load(
  file,
  select = NULL,
  n = -1,
  count_only = FALSE,
  quiet = getOption("finbif_hide_progress"),
  cache = getOption("finbif_use_cache"),
  dwc = getOption("finbif_use_dwc"),
  date_time_method = NULL,
  tzone = getOption("finbif_tz"),
  write_file = tempfile(),
  dt = NA,
  keep_tsv = FALSE,
  facts = list(),
  type_convert_facts = TRUE,
  drop_na = FALSE,
  drop_facts_na = drop_na,
  locale = getOption("finbif_locale"),
  skip = 0
)
```

Arguments

`file`	Character or Integer. Either the path to a Zip archive or tabular data file that has been downloaded from "laji.fi", a URI linking to such a data file (e.g., https://tun.fi/HBF.49381) or an integer representing the URI (i.e., 49381).
`select`	Character vector. Variables to return. If not specified, a default set of commonly used variables will be used. Use `"default_vars"` as a shortcut for this set. Variables can be deselected by prepending a `-` to the variable name. If only deselects are specified the default set of variables without the deselection will be returned. Use `"all"` to select all available variables in the file.
`n`	Integer. How many records to import. Negative and other invalid values are ignored causing all records to be imported.
`count_only`	Logical. Only return the number of records available.
`quiet`	Logical. Suppress the progress indicator for multipage downloads. Defaults to value of option `finbif_hide_progress`.
`cache`	Logical or Integer. If `TRUE` or a number greater than zero, then data-caching will be used. If not logical then the cache will be invalidated after the number of hours indicated by the argument. If a length one vector is used, its value will only apply to caching occurrence records. If the value is length two, then the second element will determine how metadata is cached.
`dwc`	Logical. Use Darwin Core (or Darwin Core style) variable names.
`date_time_method`	Character. Passed to `lutz::tz_lookup_coords()` when `date_time` and/or `duration` variables have been selected. Default is `"fast"` when less than 100,000

records are requested and "none" when more. Using method "none" assumes all records are in timezone "Europe/Helsinki", Use date_time_method = "accurate" (requires package sf) for greater accuracy at the cost of slower computation.

tzone — Character. If date_time has been selected the timezone of the outputted date-time. Defaults to system timezone.

write_file — Character. Path to write downloaded zip file to if file refers to a URI. Will be ignored if getOption("finbif_cache_path") is not NULL and will use the cache path instead.

dt — Logical. If package, data.table, is available return a data.table object rather than a data.frame.

keep_tsv — Logical. Whether to keep the TSV file if file is a ZIP archive or represents a URI. Is ignored if file is already a TSV. If TRUE the tsv file will be kept in the same directory as the ZIP archive.

facts — List. A named list of "facts" to extract from supplementary "fact" files in a local or online FinBIF data archive. Names can include one or more of "record", "event" or "document". Elements of the list are character vectors of the "facts" to be extracted and then joined to the return value.

type_convert_facts — Logical. Should facts be converted from character to numeric or integer data where applicable?

drop_na — Logical. A vector indicating which columns to check for missing data. Values recycled to the number of columns. Defaults to all columns.

drop_facts_na — Logical. Should missing or "all NA" facts be dropped? Any value other than a length one logical vector with the value of TRUE will be interpreted as FALSE. Argument is ignored if drop_na is TRUE for all variables explicitly or via recycling. To only drop some missing/NA-data facts use drop_na argument.

locale — Character. One of the supported two-letter ISO 639-1 language codes. Current supported languages are English, Finnish and Swedish. For data where more than one language is available the language denoted by locale will be preferred while falling back to the other languages in the order indicated above.

skip — Integer. The number of lines of the data file to skip before beginning to read data (not including the header).

Value

A data.frame, or if count_only = TRUE an integer.

Examples

```
## Not run:

# Get occurrence data
finbif_occurrence_load(49381)


## End(Not run)
```

finbif_request_token *Get a FinBIF personal access token*

Description

Have a personal access token for use with the FinBIF API sent to a specified email address.

Usage

```
finbif_request_token(email, quiet = FALSE)
```

Arguments

email	Character. The email address to which to send the API access token.
quiet	Logical. Suppress messages.

Value

If an access token has already been set then NULL (invisibly) if not then, invisibly, a finbif_api object containing the response from the FinBIF server.

Examples

```
## Not run:

# Request a token for example@email.com
finbif_request_token("example@email.com")


## End(Not run)
```

finbif_taxa *Search the FinBIF taxa*

Description

Search the FinBIF database for taxon.

Usage

```
finbif_taxa(
  name,
  n = 1,
  type = c("exact", "partial", "likely"),
  cache = getOption("finbif_use_cache"),
  locale = getOption("finbif_locale")
```

```
)

common_name(name, locale = getOption("finbif_locale"))

scientific_name(name)

taxon_id(name)
```

Arguments

name	Character. The name or ID of a taxon. Or, for functions other than finbif_taxa a finbif_taxa object.
n	Integer. Maximum number of matches to return. For types "exact" and "likely" only one taxon will be returned.
type	Character. Type of match to make. Must be one of exact, partial or likely.
cache	Logical or Integer. If TRUE or a number greater than zero, then data-caching will be used. If not logical then cache will be invalidated after the number of hours indicated by the argument.
locale	Character. One of the supported two-letter ISO 639-1 language codes. Current supported languages are English, Finnish and Swedish. For data where more than one language is available the language denoted by locale will be preferred while falling back to the other languages in the order indicated above.

Value

For finbif_taxa a finbif_taxa object. Otherwise, a character vector.

Examples

```
## Not run:

# Search for a taxon
finbif_taxa("Ursus arctos")

# Use partial matching
finbif_taxa("Ursus", n = 10, "partial")

# Get Swedish name of Eurasian Eagle-owl
common_name("Bubo bubo", "sv")

# Get scientific name of "Otter"
scientific_name("Otter")

# Get taxon identifier of "Otter"
taxon_id("Otter")


## End(Not run)
```

finbif_update_cache *Update cache*

Description

Update all cached FinBIF API requests.

Usage

```
finbif_update_cache()
```

Value

NULL invisibly

Examples

```
## Not run:

finbif_update_cache()


## End(Not run)
```

to_dwc *Convert variable names*

Description

Convert variable names to Darwin Core or FinBIF R package native style.

Usage

```
to_dwc(...)

to_native(...)

from_schema(..., to = c("native", "dwc"), file = c("none", "citable", "lite"))
```

Arguments

`...`	Character. Variable names to convert. For `to_dwc` and `to_native` the names must be in the opposite format. For `from_schema` the names must be from the FinBIF schema (e.g., names returned by https://api.laji.fi) or a FinBIF download file (citable or lite).
`to`	Character. Type of variable names to convert to.
`file`	Character. For variable names that are derived from a FinBIF download file which type of file.

Value

Character vector.

Examples

```
to_dwc("record_id", "date_time", "scientific_name")
```

`variables` *FinBIF record variables*

Description

FinBIF record variables that can be selected in a finbif occurrence search.

Identifiers

All identifiers are returned in the form of a URI. Identifiers include:

- `occurrenceID` Character. The ID of a record of organism's occurrence at a time and place.
- `organismID` Character. ID of an individual organism (e.g., a ringed bird that has been captured multiple times will have a single `organismID` and multiple `occurrenceID`s corresponding to each capture).
- `eventID` Character. Event ID. An event can contain one or more records (e.g., a survey of plants at a particular location and time.)
- `catalogNumber` Character. Document ID. A set of events that share common metadata.
- `formID` Character. Form ID. The form used to create the document, event, record data.
- `datasetID` Character. Dataset ID. All documents, events, and records belong to a collection/dataset (e.g., a museum collection, or the datasets collected by a specific institution). Collections themselves can be part of a larger (super)collection (e.g., all the collections at a specific museum). Only the lowest level dataset ID for a record is returned. Use `finbif_collections()` to explore the hierarchy of collections/datasets.
- `sourceID` Character. Source ID. The source of the collection's data.

Taxa

Variables related to taxonomy of records include:

- `taxonConceptID` Character. The taxon ID in the form of a URI.
- `originalTaxonConceptID` Character. The taxon ID before (if any) annotation.
- `annotatedTaxonConceptID` Character. The new taxon ID if the record has had it's taxonomy annotated.
- `verbatimTaxonConceptID` Character. The taxon ID as originally reported by the record creator.
- `scientificName` Character. Scientific name of taxon.

- originalScientificName Character. The scientific name before (if any) annotation.
- scientificNameDisplay Character. Scientific name of taxon formatted for display (e.g., taxa with genus only will be formatted as *Genus sp.*).
- originalScientificNameDisplay Character. Scientific display name before (if any) annotation.
- vernacularName Character. Common (vernacular) name of taxon.
- originalVernacularName Character. Common name before (if any) annotation.
- verbatimIdentification Character. The name of the taxon as originally reported by the record creator.
- scientificNameItalicised Logical. Is the scientific name normally italicised (i.e., is the taxonomic rank genus or below.)
- originalScientificNameItalicised Logical. Is the original scientific name normally italicised.
- scientificNameAuthorship Character. The authority for the taxon scientific name.
- originalScientificNameAuthorship Character. The authority for the taxon scientific name before (if any) annotation.
- verbatimScientificNameAuthorship Character. The authority of the taxon as originally reported by the record creator.
- taxonRank Character. The taxonomic rank of the taxon.
- originalTaxonRank Character. The taxonomic rank of the taxon before (if any) annotation.
- informalTaxonGroups List. The informal taxonomic groups that the taxon belongs to (e.g., birds) in the form of URIs.
- originalInformalTaxonGroups List. The informal taxonomic groups that the taxon belonged to before (if any) annotation.
- verbatimInformalTaxonGroups List. The informal taxonomic groups that the taxon belongs to as reported by the record creator.
- taxonChecklist Character. The checklist (as a URI) that that taxon is found in.
- originalTaxonChecklist Character. The checklist (as a URI) that that taxon was found in before (if any) annotation.
- taxonFinnishStatus Logical. Is the taxon considered Finnish. The definition of a Finnish taxon differs by taxonomic group?
- originalFinnishStatus Logical. Was the taxon considered Finnish before (if any) annotation?

Abundance, sex & life history

Variables related to abundance, sex and life history include:

- individualCount Integer. Number of individuals recorded.
- individualCountInterpreted Integer. Number of individuals recorded or inferred from the record. Note that many records with individualCount == 1L only indicate the record is of one individual and may not necessarily imply that this was the abundance at that specified place and time (e.g., a preserved museum specimen consisting of a single individual).

- `individualCount{Female|Male}` Integer. Number of female or male individuals recorded.
- `pairCount` Integer. Number of mating pairs recorded.
- `organismQuantity` Character. The abundance as reported by the record creator.
- `lifeStage` Character. Life stage of individual(s) recorded.
- `sex` Character. Sex of individual(s) recorded.

Location

Variables related to the location of records include:

- `decimal{Latitude|Longitude}` Numeric. Coordinates (in WGS84 coordinate system) of the central point of a bounding box encompassing the record's geographic coverage.
- `decimal{Latitude|Longitude}EUREF` Numeric. Coordinates (in EUREF coordinate system) of the central point of a bounding box encompassing the record's geographic coverage.
- `decimal{Latitude|Longitude}{Min|Max}{EUREF|YKJ|WGS84}`. Numeric. Vertices of a bounding box encompassing the record's geographic coverage. Coordinates are available in EUREF, YKJ, or WGS84.
- `coordinateUncertaintyInMeters` Integer. The horizontal distance (in meters) from the record's given coordinates describing the smallest circle containing the whole of the record's location.
- `georeferenceSources` Character. Source of coordinates.
- `footprintWKT` Character. Well-Known Text (WKT) representation of the geographic shape defining the location of the record in WGS84 coordinate systems.
- `footprintWKT{_EUREF|_YKJ}` Character. Well-Known Text (WKT) representation of the geographic shape defining the location of the record in either EUREF or YKJ coordinate systems.
- `country` Character. The country of the record's location.
- `stateProvince` Character. The administrative area directly below the level of country.
- `bioStateProvince` Character. For data from Finland FinBIF uses the concept of Biogeographical Province. See link for details.
- `county`. Character. Administrative level below region
- `higherGeography` Character. Geographic place name that is at higher level than country.
- `lineLengthMeters` Integer. The length of linear locations (e.g., line transect surveys).
- `areaMetersSquared` Integer. The size of record's location in meters squared.
- `isBreedingSite` Logical. Whether or not the occurrence is recorded at a known breeding location.
- `locationID` Character. A location ID in the form of a URI.
- `eventSection` Integer. A numeric identifier for a sub-location of a location (e.g., a specific part of a transect that undergoes repeated surveys.)

Time

Variables related to time of record include:

- `eventDateTime` POSIXct. The date and time of the recording event. This variable is computed after records are downloaded from FinBIF. Its timezone and accuracy can be controlled see `finbif_occurrence()` for details.
- `samplingEffort` Duration. The duration of the recording event. This variable is computed after records are downloaded from FinBIF.
- `eventDateStart` Character. The date the recording event began.
- `eventDateEnd` Character. The date the recording event ended.
- `hourStart` Integer. The hour (24 hour time) of the day the recording event began.
- `hourEnd` Integer. The hour (24 hour time) of the day the recording event ended.
- `minuteStart` Integer. The minute of the hour the recording event began.
- `minuteEnd` Integer. The minute of the hour the recording event started.
- `startDayOfYear` Integer. The ordinal day of the year the recording event began.
- `endDayOfYear` Integer. The ordinal day of the year the recording event ended
- `seasonStart` Integer. The day of the year the recording event began. A four digit number indicating the day of the year in MMDD (%m%d) format.
- `seasonEnd` Integer. The day of the year the recording event ended. A four digit number indicating the day of the year in MMDD (%m%d) format.
- `century` Integer. The century during which the recording event occurred (`NA` if the event spans multiple centuries).
- `decade` Integer. The decade during which the recording event occurred (`NA` if the event spans multiple decades).
- `year` Integer. The year during which the recording event occurred (`NA` if the event spans multiple years).
- `month` Integer. The month of the year during which the recording event occurred (`NA` if the event spans multiple months).
- `day` Integer. The day of the month during which the recording event occurred (`NA` if the event spans multiple days).
- `eventDateTimeDisplay` Character. Date and time of the recording event formatted for display.
- `documentCreatedDate` Character. The date the original data was created.
- `available` Character. The date the record was first loaded into the FinBIF database.
- `modified` Character. The most recent date the original data was modified.
- `documentLoadedDate` Character. The most recent date the record was loaded into the FinBIF database.

Data restrictions

Variables related to restricted records include:

- `informationRestricted` Logical. Has the record been restricted in some way (e.g., geospatially aggregated).
- `dataGeneralizations` Character. What level of restriction has been applied to the record.
- `informationWithheld` Character. Description of information about a record that is not public available but may accessed on request.

Data quality

Variables related to the quality of records include:

- `hasIssues` Logical. Are there any data quality issues associated with the record, its event or document.
- `identificationQualifer` Reliability of the record's taxonomic identification as reported by the original data author.
- `{document|time|location|event|occurrence}Issue` Character. Issues with record associated with its document, time, location, event, or the record itself.
- `{document|time|location|event|occurrence}IssueRemarks` Character. Details about the issue.
- `{document|time|location|event|occurrence}IssueSource` Character. Source determining the issue.
- `requiresVerification` Logical. Has the record been flagged for expert verification?
- `requiresIdentification` Logical. Has the record been flagged for expert identification?
- `occurrenceReliability` Character. Indication of the record's reliability.
- `identificationVerificationStatus` Character. Indication of the record's quality.

Misc

Other variables:

- `datasetName` Character. Dataset name. All documents, events, and records belong to a collection/dataset (e.g., a museum collection, or the datasets collected by a specific institution). Collections themselves can be part of a larger (super)collection (e.g., all the collections at a specific museum). Only the lowest level collection name for a record is returned. Use `finbif_collections()` to explore the hierarchy of collections/datasets.
- `recordedByIDs` List. List of observer identifiers for the record.
- `identifiedBy` Character. Person who determined the taxonomic identification of the record.
- `basisOfRecord` Character. The type of or method used to obtain the record.
- `basisOfSuperRecord` Character. Higher level type of or method used to obtain the record.
- `typeStatus` Logical. Whether or not the record is of a type specimen.
- `occurrenceIsWild` Logical. Whether or not the record is of a "wild" organism.
- `license` Character. The license of the data associated with the record.
- `{document|event|occurrence}Remarks` Character. Notes associated with the document, event or record itself.
- `{document|occurrence}Keywords` List. List of keywords associated with the document or record.
- `occurrenceAnnotationCount` Integer. How many annotations are associated with the record.
- `materialSampleCount` Integer. How many material samples (DNA extractions, etc., ...) are associated with the record.
- `{document|event|occurrence}AssociatedMediaCount` Integer. How many media items (images, audio, video, etc., ...) are associated with the record's document, event or the record itself.

Index

caching, 4
common_name *(*finbif_taxa*)*, 20

Date, *7*

fb_check_taxa *(*finbif_check_taxa*)*, 10
fb_clear_cache *(*finbif_clear_cache*)*, 11
fb_collections *(*finbif_collections*)*, 11
fb_informal_groups
        *(*finbif_informal_groups*)*, 12
fb_last_mod *(*finbif_last_mod*)*, 13
fb_metadata *(*finbif_metadata*)*, 14
fb_occurrence *(*finbif_occurrence*)*, 15
fb_occurrence_load
        *(*finbif_occurrence_load*)*, 17
fb_renew_token *(*finbif_request_token*)*,
        20
fb_request_token
        *(*finbif_request_token*)*, 20
fb_taxa *(*finbif_taxa*)*, 20
fb_update_cache *(*finbif_update_cache*)*,
        22
filters, 5
finbif *(*finbif-package*)*, 2
finbif-package, 2
finbif_check_taxa, 10
finbif_check_taxa(), *5*
finbif_clear_cache, 11
finbif_collections, 11
finbif_informal_groups, 12
finbif_informal_groups(), *5*
finbif_last_mod, 13
finbif_metadata, 14
finbif_metadata(), *5*, *6*
finbif_occurrence, 15
finbif_occurrence_load, 17
finbif_request_token, 20
finbif_taxa, 20
finbif_taxa(), *5*
finbif_update_cache, 22

from_schema *(*to_dwc*)*, 22

scientific_name *(*finbif_taxa*)*, 20

taxon_id *(*finbif_taxa*)*, 20
to_dwc, 22
to_native *(*to_dwc*)*, 22

variables, 23