Package: f2g (via r-universe)

August 30, 2024

Title FinBIF to GBIF

Version 0.6.11.9000

Description Tools for publishing FinBIF data to GBIF.

Depends R (>= 3.5.0)

Imports config, digest, EML, emld, finbif, httr, jsonlite, rlang, tidyr, utils, xml2, wk

Suggests tinytest, webfakes, callr

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

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

RemoteUrl https://github.com/luomus/finbif2gbif

RemoteRef dev

RemoteSha 489c830b08364d9ec167a81efd50690bd4501596

Contents

`archive_occurrences` *Archive occurrences*

Description

Archive occurrence records in a Darwin Core archive.

Usage

```
archive_occurrences(
  archive,
  file_name,
  media_file_name,
  filter,
  select = sub("^.*:", "", config::get("fields")),
  facts = config::get("facts"),
  combine = config::get("combine"),
  n = config::get("nmax"),
  quiet = TRUE
)
```

Arguments

`archive`	Character. Path to the archive.
`file_name`	Character. The name of the file to write to the archive.
`media_file_name`	Character. The name of the media extension file to write to the archive.
`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.
facts	List of extra variables to be extracted from record, event and document "facts".
combine	List of fields to combine.
n	Integer. How many records to download/import.
quiet	Logical. Suppress the progress indicator for multipage downloads.

Value

The status value returned by the zip command, invisibly.

Examples

```
## Not run:

archive_occurrences(
  "dwca.zip", "occurrence.txt", list(collection = "HR.139"),
   c("occurrenceID", "basisOfRecord")
)


## End(Not run)
```

---

clean_occurrences *Clean occurrences*

Description

Clean occurrence files in an archive.

Usage

```
clean_occurrences(archive, filters)
```

Arguments

archive	Character. Path to the archive.
filters	List.

Value

The status value returned by the zip command, invisibly.

Examples

```
## Not run:

clean_occurrences("dwca.zip", list())


## End(Not run)
```

`count_occurrences` *Count occurrences*

Description

Count the number of occurrences.

Usage

```
count_occurrences(x, ...)
```

Arguments

`x`	Object to count occurrences for.
`...`	Arguments passed to methods.

Value

Integer.

Examples

```
 ## Not run:

count_occurrences(list(collection = "HR.3991"))


## End(Not run)
```

get_archive_path *Get archive file path*

Description

Get the file path of an archive for a collection.

Usage

```
get_archive_path(collection_id, dir = "archives/split")
```

Arguments

collection_id	Character. Collection id.
dir	Character. Path to the archive directory.

Value

Character. The file path of the archive.

Examples

```
## Not run:

get_archive_path("HR.3991")


## End(Not run)
```

get_collection_ids *Get collection IDs*

Description

Get collection IDs of FinBIF collections that are published to GBIF.

Usage

```
get_collection_ids(datasets, collection_ids = config::get("collections"))
```

Arguments

datasets	List. GBIF dataset metadata retrieved using gbif_datasets.
collection_ids	Character. Collection ids to include regardless of sharing status.

Value

A character vector.

Examples

```
## Not run:

get_collection_ids()


## End(Not run)
```

`get_endpoint` *Get endpoint*

Description

Get FinBIF collection data endpoint needed for GBIF registration.

Usage

```
get_endpoint(collection_id, url_base = Sys.getenv("ENDPOINTS"))
```

Arguments

`collection_id`	Character. ID string of FinBIF collection.
`url_base`	Character. The base URL for the collection's data endpoint. Defaults to system environment variable, "ENDPOINTS".

Value

A list.

Examples

```
## Not run:

get_endpoint("HR.3991")


## End(Not run)
```

get_file_name *Get occurrence file name.*

Description

Get the file name of occurrences in an archive

Usage

```
get_file_name(filter, select = config::get("fields"), prefix = "occurrence")
```

Arguments

filter	List.
select	Character.
prefix	Character.

Value

Character. The file name holding occurrence records.

Examples

```
## Not run:

get_file_name(list())


## End(Not run)
```

get_gbif_datasets *GBIF datasets*

Description

Get metadata for GBIF registered datasets of a given installation.

Usage

```
get_gbif_datasets(
  url = Sys.getenv("GBIF_API"),
  installation = Sys.getenv("GBIF_INSTALLATION")
)
```

Arguments

url	Character. URL of GBIF API. Defaults to system environment variable, "GBIF_API".
installation	Character. ID key of GBIF installation. Defaults to system environment variable, "GBIF_INSTALLATION".

Value

A list.

Examples

```
## Not run:

get_gbif_datasets()


## End(Not run)
```

---

get_metadata *Get metadata*

Description

Get FinBIF collection metadata needed for GBIF registration.

Usage

```
get_metadata(
  collection_id,
  metadata_fields = config::get("metadata"),
  org = Sys.getenv("GBIF_ORG"),
  installation = Sys.getenv("GBIF_INSTALLATION")
)
```

Arguments

collection_id	Character. ID string of FinBIF collection.
metadata_fields	List. Map of GBIF to FinBIF metadata fields to use.
org	Character. GBIF organization key. Defaults to system environment variable, "GBIF_ORG".
installation	Character. ID key of GBIF installation. Defaults to system environment variable, "GBIF_INSTALLATION".

Value

A list.

Examples

```
## Not run:

get_metadata("HR.3991")


## End(Not run)
```

`get_occurrences` *Get occurrences*

Description

Get occurrence records from FinBIF.

Usage

```
get_occurrences(filter, select, facts, combine, n, quiet = TRUE)
```

Arguments

`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.
`facts`	List of extra variables to be extracted from record, event and document "facts".
`combine`	List of fields to combine.
`n`	Integer. How many records to download/import.
`quiet`	Logical. Suppress the progress indicator for multipage downloads.

Value

A finbif_occ object.

Examples

```
## Not run:

get_occurrences(
  c(collection = "HR.3991"), c("occurrenceID", "basisOfRecord"), 100
)


## End(Not run)
```

get_registration *Check registration*

Description

Check if a FinBIF collection is registered with GBIF.

Usage

```
get_registration(datasets, collection_id, quiet = FALSE)
```

Arguments

datasets	List. GBIF dataset metadata retrieved using gbif_datasets.
collection_id	Character. ID string of FinBIF collection.
quiet	Logical. Suppress messages.

Value

Integer.

Examples

```
## Not run:

get_registration(gbif_datasets(), "HR.3991")


## End(Not run)
```

get_subsets *Get subsets*

Description

Get subset filters for a collection.

Usage

```
get_subsets(
  collection_id,
  filters = config::get("filters"),
  nmax = config::get("nmax")
)
```

Arguments

collection_id	Character. ID string of FinBIF collection.
filters	List.
nmax	Integer. Maximum allowed size of subset.

Value

A list.

Examples

```
## Not run:

get_subsets("HR.3991")


## End(Not run)
```

---

get_uuid *Get UUID*

Description

Get the UUID of a registered dataset.

Usage

```
get_uuid(registration)
```

Arguments

registration	Integer.

Value

Character.

Examples

```
## Not run:

registration <- get_registration(gbif_datasets(), "HR.3991")
get_uuid(registration)


## End(Not run)
```

initiate_gbif_ingestion *Initiate ingestion*

Description

Ingitiate GBIF ingestion of FinBIF data.

Usage

```
initiate_gbif_ingestion(
  uuid,
  url = Sys.getenv("GBIF_API"),
  user = Sys.getenv("GBIF_USER"),
  pass = Sys.getenv("GBIF_PASS")
)
```

Arguments

uuid	Integer. GBIF registration id.
url	Character. URL of GBIF API. Defaults to system environment variable, "GBIF_API".
user	Character. GBIF username. Defaults to system environment variable, "GBIF_USER".
pass	Character. GBIF password. Defaults to system environment variable, "GBIF_PASS".

Value

NULL.

Examples

```
## Not run:

collection <- get_collection_ids()[[1L]]
registration <- get_registration(get_gbif_datasets(), collection)
initiate_gbif_ingestion(registration)


## End(Not run)
```

last_mod *Get last modified date*

Description

Get the last modified data for FinBIF records

Usage

```
last_mod(x, ...)
```

Arguments

x	Object to get last modified time for.
...	Arguments passed to methods.

Value

A Date object.

Examples

```
## Not run:

last_mod(list(collection = "HR.3991"))


## End(Not run)
```

n_archived_subsets *Number of archived subsets*

Description

Count the number of occurrence data subsets that have been archived.

Usage

```
n_archived_subsets(archive)
```

Arguments

archive	Darwin Core archive file.

Value

Integer.

Examples

```
 ## Not run:

n_archived_subsets("archive.zip")


## End(Not run)
```

`publish_archive` *Publish archive*

Description

Publish a Darwin Core archive.

Usage

```
publish_archive(staged_archive, dir = "archives")
```

Arguments

`staged_archive`	Character. Path to the staged archive.
`dir`	Character. Path to the archive directory.

Value

Character. The file path of the staged archive.

Examples

```
## Not run:

publish_archive("stage/archive.zip")


## End(Not run)
```

send_gbif_dataset_endpoint *GBIF dataset endpoint*

Description

Send FinBIF dataset endpoint to GBIF.

Usage

```
send_gbif_dataset_endpoint(
  endpoint,
  uuid,
  url = Sys.getenv("GBIF_API"),
  user = Sys.getenv("GBIF_USER"),
  pass = Sys.getenv("GBIF_PASS")
)
```

Arguments

endpoint	Character. URL of dataset endpoint generated by `get_endpoint`.
uuid	Character. GBIF dataset identifier. Returned by `send_gbif_dataset_metadata`.
url	Character. URL of GBIF API. Defaults to system environment variable, "GBIF_API".
user	Character. GBIF username. Defaults to system environment variable, "GBIF_USER".
pass	Character. GBIF password. Defaults to system environment variable, "GBIF_PASS".

Value

If successful returns `NULL` invisibly.

Examples

```
## Not run:

m <- get_metadata("HR.3991")
ep <- get_endpoint("HR.3991")
uuid <- send_gbif_dataset_metadata(m)
send_gbif_dataset_endpoint(ep, uuid)


## End(Not run)
```

send_gbif_dataset_id *GBIF dataset identifier*

Description

Send FinBIF dataset identifier to GBIF.

Usage

```
send_gbif_dataset_id(
  id,
  uuid,
  url = Sys.getenv("GBIF_API"),
  user = Sys.getenv("GBIF_USER"),
  pass = Sys.getenv("GBIF_PASS")
)
```

Arguments

id	Character. FinBIF collection ID for dataset.
uuid	Character. GBIF dataset identifier. Returned by send_gbif_dataset_metadata.
url	Character. URL of GBIF API. Defaults to system environment variable, "GBIF_API".
user	Character. GBIF username. Defaults to system environment variable, "GBIF_USER".
pass	Character. GBIF password. Defaults to system environment variable, "GBIF_PASS".

Value

If successful returns NULL invisibly.

Examples

```
## Not run:

m <- get_metadata("HR.3991")
uuid <- send_gbif_dataset_metadata(m)
send_gbif_dataset_id("HR.3991", uuid)


## End(Not run)
```

send_gbif_dataset_metadata *Send metadata*

Description

Send FinBIF dataset metadata to GBIF.

Usage

```
send_gbif_dataset_metadata(
  metadata,
  url = Sys.getenv("GBIF_API"),
  user = Sys.getenv("GBIF_USER"),
  pass = Sys.getenv("GBIF_PASS")
)
```

Arguments

metadata	List. FinBIF dataset metadata generated by get_metadata.
url	Character. URL of GBIF API. Defaults to system environment variable, "GBIF_API".
user	Character. GBIF username. Defaults to system environment variable, "GBIF_USER".
pass	Character. GBIF password. Defaults to system environment variable, "GBIF_PASS".

Value

A list.

Examples

```
## Not run:

m <- get_metadata("HR.3991")
send_gbif_dataset_metadata(m)


## End(Not run)
```

skip_collection *Skip collection*

Description

Should the collection be skipped?

Usage

```
skip_collection(
  collection_id,
  enabled = config::get("enabled"),
  whitelist = "whitelist.txt"
)
```

Arguments

collection_id	Character. Collection id.
enabled	Logical.
whitelist	Character. Path to white-list file.

Value

Logical.

Examples

```
## Not run:

skip_collection("HR.139")


## End(Not run)
```

skip_gbif *Skip GBIF update*

Description

Should updating the collection for GBIF be skipped?

Usage

```
skip_gbif(collection_id, enabled = config::get("gbif"))
```

Arguments

collection_id	Character. Collection id.
enabled	Logical.

Value

Logical.

Examples

```
## Not run:

skip_gbif("HR.139")


## End(Not run)
```

---

stage_archive *Get archive file path*

Description

Get the file path of an archive for a collection.

Usage

```
stage_archive(archive, stage = "stage")
```

Arguments

archive	Character. Path to the archive.
stage	Character. Path to the staging directory.

Value

Character. The file path of the staged archive.

Examples

```
## Not run:

stage_archive("archive.zip")


## End(Not run)
```

unstage_archive *Unstage archive*

Description

Unstage an updated archive file.

Usage

```
unstage_archive(staged_archive, dir = "archives")
```

Arguments

staged_archive	Character. Path to the staged archive.
dir	Character. Path to the archive directory.

Value

Character. The file path of the staged archive.

Examples

```
## Not run:

publish_archive("stage/archive.zip")


## End(Not run)
```

update_gbif_dataset_endpoint *Update GBIF endpoint*

Description

Update FinBIF dataset endpoint for GBIF.

Usage

```
update_gbif_dataset_endpoint(
  endpoint,
  uuid,
  url = Sys.getenv("GBIF_API"),
  user = Sys.getenv("GBIF_USER"),
  pass = Sys.getenv("GBIF_PASS")
)
```

Arguments

endpoint	Character. URL of dataset endpoint generated by get_endpoint.
uuid	Character. GBIF dataset identifier.
url	Character. URL of GBIF API. Defaults to system environment variable, "GBIF_API".
user	Character. GBIF username. Defaults to system environment variable, "GBIF_USER".
pass	Character. GBIF password. Defaults to system environment variable, "GBIF_PASS".

Value

If successful returns NULL invisibly.

Examples

```
## Not run:

m <- get_metadata("HR.3991")
ep <- get_endpoint("HR.3991")
uuid <- send_gbif_dataset_metadata(m)
update_gbif_dataset_endpoint(ep, uuid)


## End(Not run)
```

---

update_gbif_dataset_metadata *Update metadata*

Description

Update FinBIF dataset metadata at GBIF.

Usage

```
update_gbif_dataset_metadata(
  metadata,
  registration,
  url = Sys.getenv("GBIF_API"),
  user = Sys.getenv("GBIF_USER"),
  pass = Sys.getenv("GBIF_PASS")
)
```

Arguments

metadata	List. FinBIF dataset metadata generated by get_metadata.
registration	Integer. GBIF registration.
url	Character. URL of GBIF API. Defaults to system environment variable, "GBIF_API".
user	Character. GBIF username. Defaults to system environment variable, "GBIF_USER".
pass	Character. GBIF password. Defaults to system environment variable, "GBIF_PASS".

Value

NULL.

Examples

```
## Not run:

collection <- get_collection_ids()[[1L]]
registration <- get_registration(get_gbif_datasets(), collection)
update_gbif_dataset_metadata(get_metadata(collection), registration)


## End(Not run)
```

`write_eml` *Write EML*

Description

Write an EML metadata file.

Usage

```
write_eml(archive, collection_id, uuid, metadata, eml = config::get("eml"))
```

Arguments

`archive`	Character. Path to a DarwinCore archive.
`collection_id`	Character. Collection ID.
`uuid`	Character. GBIF ID.
`metadata`	List.
`eml`	List.

Value

The status value returned by the zip command, invisibly.

Examples

```
## Not run:

registration <- get_registration(gbif_datasets(), "HR.3991")
uuid <- get_uuid(registration)
write_eml("dwca.zip", "HR.447", uuid, list())


## End(Not run)
```

`write_meta` *Write metafile*

Description

Write a Darwin Core archive metadata file.

Usage

```
write_meta(
  archive,
  filters,
  fields = config::get("fields"),
  facts = config::get("facts"),
  id = 1
)
```

Arguments

`archive`	Character. Path to the archive.
`filters`	List.
`fields`	Character vector. The field names of the data files. Field names can optionally be prepended with a namespace (one of "dwc", "dwciri", "dc" or "dcterms") separated from the field by a ":". If no namespace is specified, "dwc" will be assumed.
`facts`	List of extra variables to be extracted from record, event and document "facts".
`id`	Integer. Indicates which field can be considered the record identifier. No ID field will be specified if `id` is not an integer between 1 and the number of fields specified.

Value

The status value returned by the zip command, invisibly.

Examples

```
## Not run:

write_meta(
  "dwca.zip", list(collection = "HR.447"), c("occurrenceID", "basisOfRecord")
)


## End(Not run)
```

`write_occurrences` *Write occurrences*

Description

Write occurrence records to a Darwin Core archive.

Usage

```
write_occurrences(
  data,
  archive,
  file_name = "occurrence.txt",
  media_file_name = "media.txt"
)
```

Arguments

`data`	A data.frame. Occurrence records.
`archive`	Character. Path to the archive.
`file_name`	Character. The name of the file to write to the archive.
`media_file_name`	Character. The name of the media extension file to write to the archive.

Value

The status value returned by the zip command, invisibly.

Examples

```
## Not run:

data <- get_occurrences(
  c(collection = "HR.3991"), c("occurrenceID", "basisOfRecord"), 100
)
write_occurrences(data, "dwca.zip")


## End(Not run)
```

Index